VMware vSphere Integrated Containers for vSphere Administrators

vSphere Integrated Containers

vmware

Table of Contents

vSphere Integrated Containers for vSphere Administrators

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

Product version: 0.5

NOTE This book is a work in progress.

Intended Audience

This information is intended for vSphere® Administrators who must manage a vSphere Integrated Containers implementation in their vSphere environment. The information is written for experienced vSphere administrators who are familiar with virtual machine technology and datacenter operations. Knowledge of container technology and Docker is assumed.

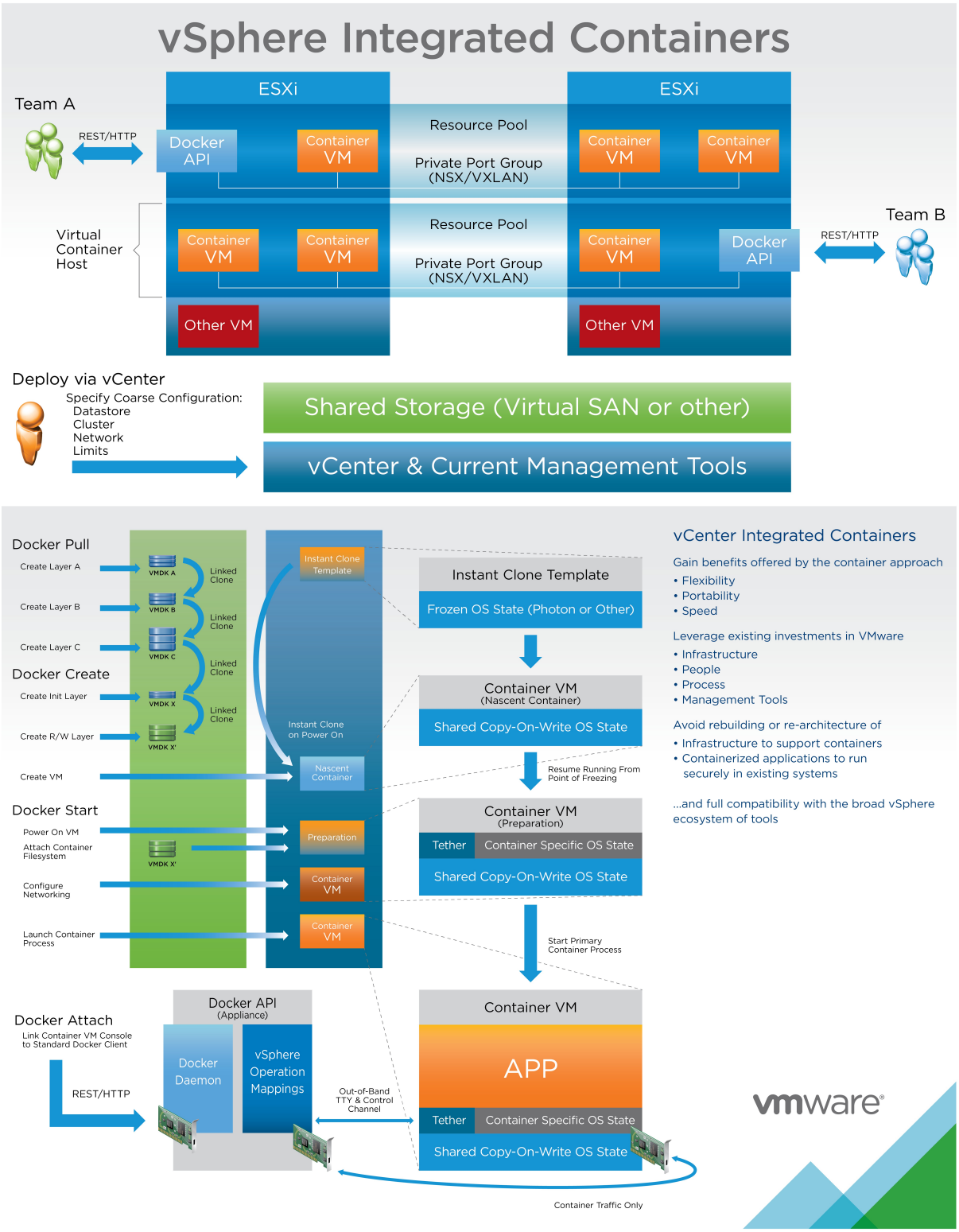
VMware, Inc. 3401 Hillview Ave. Palo Alto, CA94304

www.vmware.com

vSphere Integrated Containers Architecture

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

- vCenter management tools: monitor and manage virtual machines as well as container virtual machines.
- vCenter Server: manage a single ESXi host or cluster of ESXi hosts with DRS enabled. Specify and deploy datastores and network paths, define clusters, resource pools, and port groups.
- Trusted networks: Deploy and use vSphere Integrated Containers and connections between Docker clients and virtual container hosts.
- Virtual SAN datastores: specify a datastore and top level directory with the name of the virtual container host.
- Docker API appliance virtual machine: The installer deploys this appliance virtual machine, which is also referred to as the virtual container host. You set the docker client to this appliance.
- Docker container virtual machines: Using vSphere Instant Clone and Photon OS technology, you can create and provision multiple container virtual machines directly from a template. The Docker daemon runs outside the container virtual machine. The container is a x86 hardware virtualized virtual machine with a process ID, container interfaces and mounts.



Virtual Container Host

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

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

vSphere resource management handles container placement within the virtual container host, so that a virtual Docker host can serve as an entire vSphere cluster or a fraction of the same cluster. The only resource consumed by a container host in the cluster is the resource consumed by running containers.

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

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

You can also nest virtual container hosts, giving your team access to a large virtual container host, or sub-allocate smaller virtual container hosts for individuals.

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

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

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

vSphere Web Client Plugin

You can monitor and manage containers using the vSphere Integrated Containers plugin in the vSphere Web Client.

The plugin allows you to create virtual container hosts, perform administrative tasks on containers such as resource allocation, port mapping, and manage communications between administrators, developers, and application owners during troubleshooting.

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

Docker Client

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

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

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

vSphere Integrated Containers Interoperability with Other VMware Software

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

vRealize Suite

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

vSphere High Availability, Fault Tolerance, and vSphere vMotion

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

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

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

VMware Virtual SAN

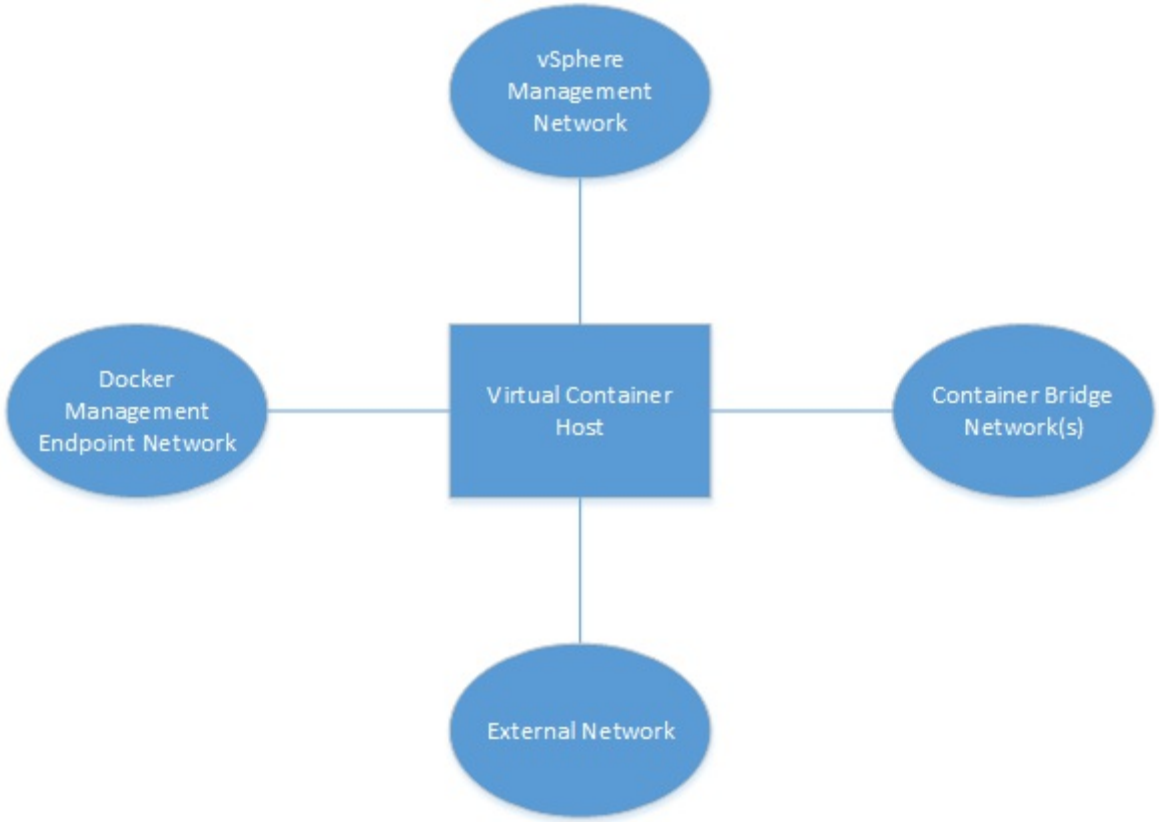
The virtual container host maintains filesystem layers inherent in container images by mapping to discrete VMDK files, all of which are housed in vSphere datastores on VSAN, NFS, or local disks.

vSphere Instant Clone

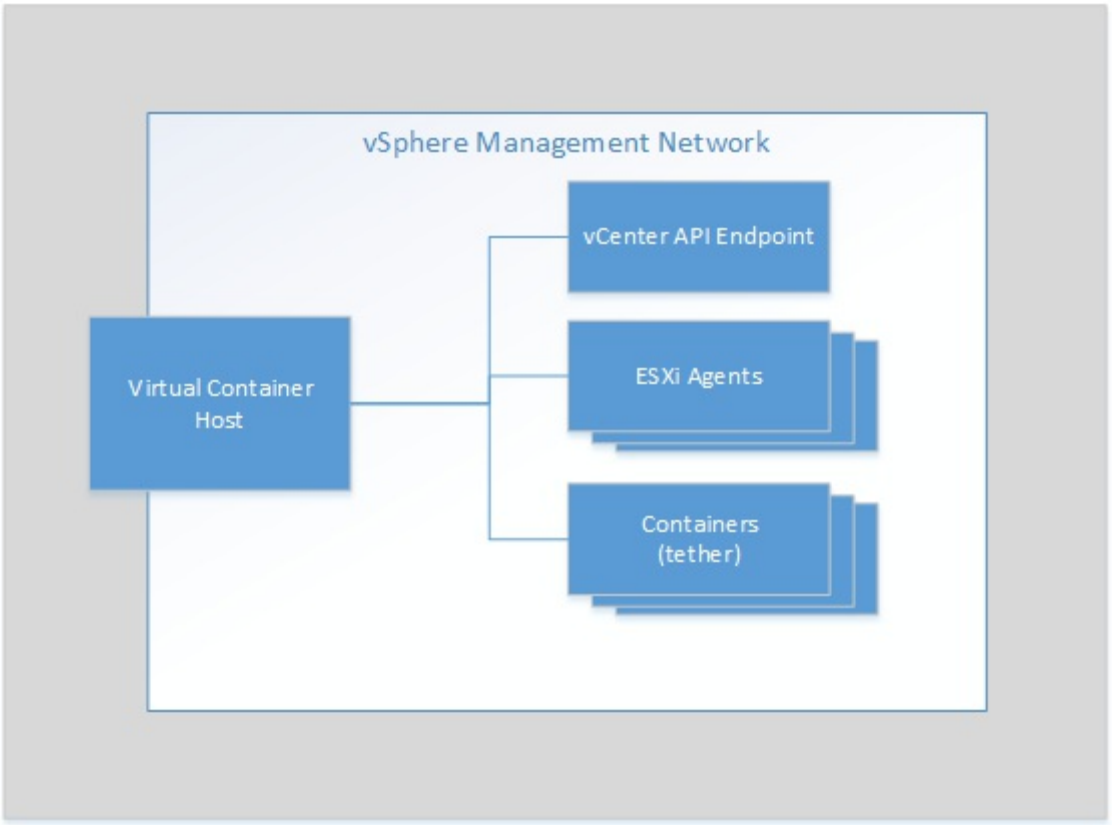
vSphere Integrated Containers allows you to create and run multiple containers rapidly with minimal overhead using vSphere 6 Instant Clone technology, which provisions child VMs forked directly from a parent VM template running a Linux kernel. vSphere Integrated Containers creates the kernel and a few supporting resources to run containers using Photon OS technology.

vSphere Integrated Containers Network Overview

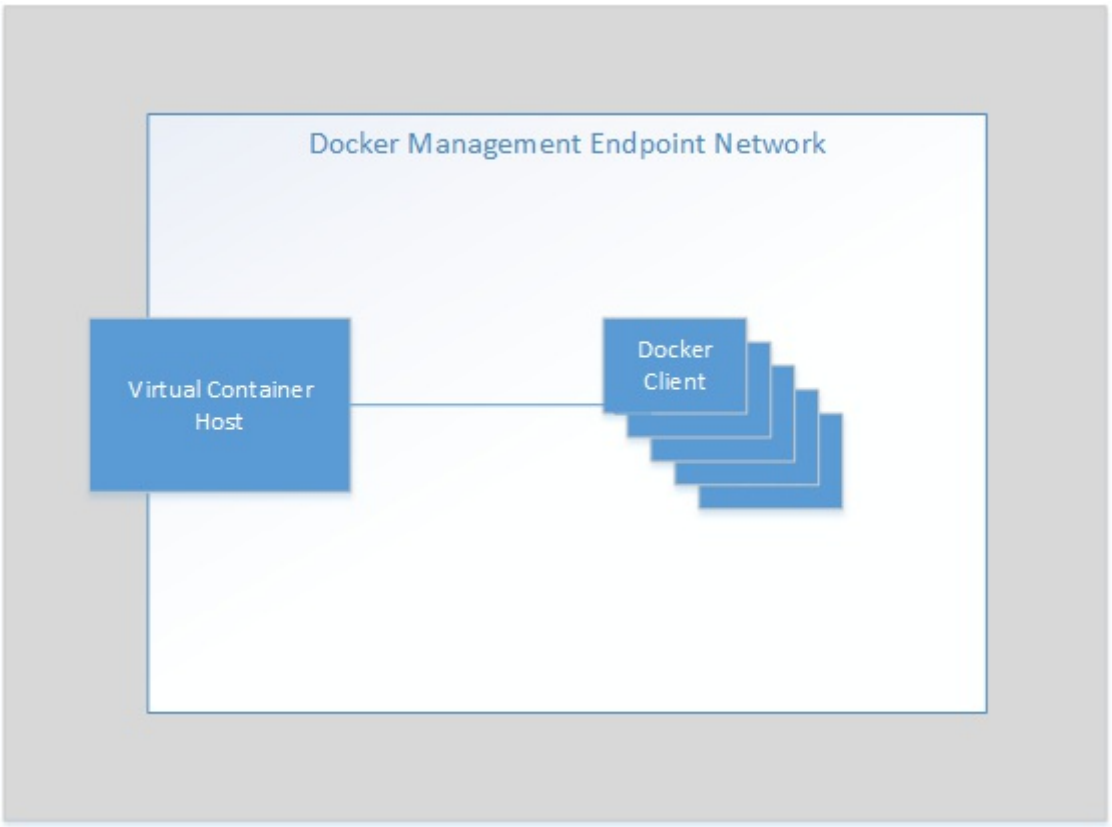
vSphere Container Host connects to four network types.



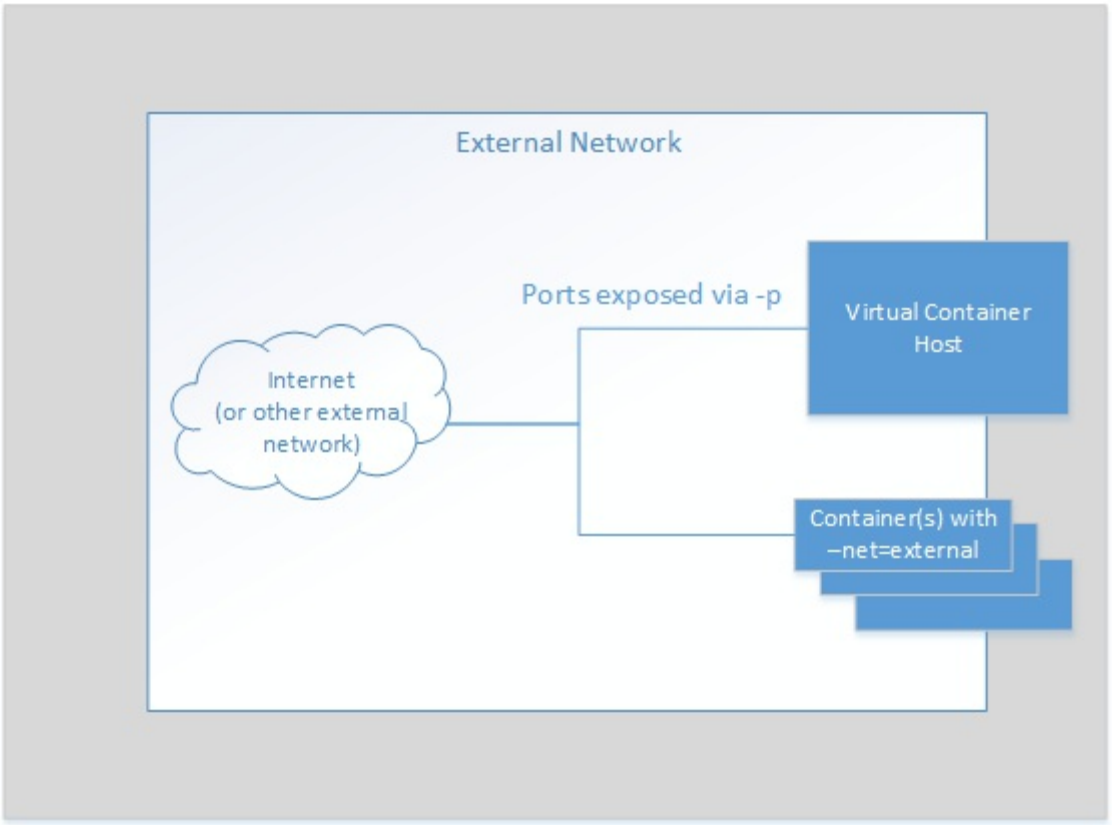
- vSphere Management Network: to communicate with vCenter and ESXi hosts. This network also serves as a tether within the containers to communicate with the vSphere Container Host.



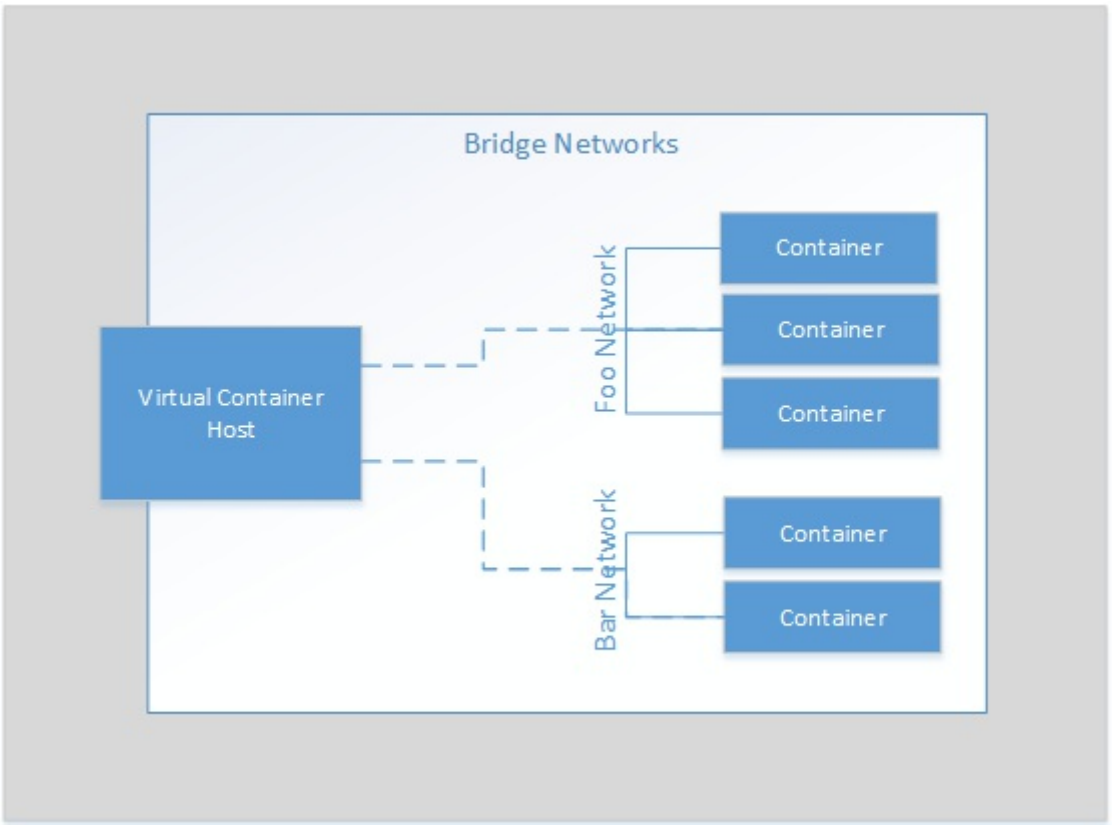
- Docker Management Endpoint Network: to connect to Docker clients and isolate the Docker endpoints from the more public external network.



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



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



The Port Layer

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

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

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

Tether Process

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

Virtual Container Host Administration

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

Obtain `vic-machine` Version Information

You can obtain information about the version of `vic-machine` by using the `vic-machine version` command.

Prerequisites

You have downloaded and unpacked the vSphere Integrated Containers binaries.

Procedure

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

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

   ```
   $ vic-machine-darwin-linux-windows version
   ```

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

Obtain Information About a Virtual Container Host

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

Prerequisites

You have a virtual container host.

Procedure

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

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

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

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

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

Virtual Container Host Inspect Options

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

`target`

Short name: `-t`

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

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

  ```
  --target esxi_host_address
  ```

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

  ```
  --target vcenter_server_address
  ```

- You can optionally include the user name and password of the ESXi host or vCenter Server 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_or_vcenter_server_username:password@esxi_or_vcenter_server_address
  ```

- If you deployed the virtual container host on a vCenter Server instance that includes more than one datacenter, include the datacenter name in the target URL. If you include an invalid datacenter name, `vic-machine inspect` 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 inspect` prompts you to enter the password.

`user`

Short name: `-u`

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

```
--user esxi_or_vcenter_server_username
```

`password`

Short name: `-p`

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

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'
```

compute-resource

Short name: `-r`

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

- vCenter Server includes multiple instances of standalone hosts or clusters, or a mixture of standalone hosts and clusters.
- The ESXi host includes multiple resource pools.
- You deployed the virtual container host in a specific resource pool in your environment.

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

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

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

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

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

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

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

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

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

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

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

name

Short name: `-n`

The name of the virtual container host appliance to inspect. This option is mandatory if the virtual container host to inspect has a name other than the default name, `virtual-container-host`. Specify `--name` with exactly the same value that you used when you ran `vic-machine create`.

```
--name vch_appliance_name
```

debug

Short name: `-v`

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

```
--debug 1
```

List Virtual Container Hosts

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

Prerequisites

You have deployed at least one virtual container host.

Procedure

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

   To obtain a list of virtual container hosts, you must provide the address of the target ESXi host or vCenter Server instance. You must specify the username and optionally the password, either in the `target` option or separately in the `user` and `password` options.

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

   ```
   $ vic-machine-darwin-linux-windows ls
   --target esxi_host username:password@esxi_host_address
   ```

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

Virtual container hosts running on a vCenter Server instance:

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

Virtual Container Host List Options

The command line utility for vSphere Integrated Containers, `vic-machine`, provides an `ls` command that allows you to see a list of virtual container hosts that are running on an ESXi host or vCenter Server instance. The options that `vic-machine ls` requires depend on the location in your vSphere environment in which you deployed the virtual container host.

`target`

Short name: `-t`

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

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

  ```
  --target esxi_host_address
  ```

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

  ```
  --target vcenter_server_address
  ```

- You can optionally include the user name and password of the ESXi host or vCenter Server 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_or_vcenter_server_username:password@esxi_or_vcenter_server_address
  ```

- If you deployed the virtual container hosts 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 inspect` 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 ls` prompts you to enter the password.

`user`

Short name: `-u`

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

```
--user esxi_or_vcenter_server_username
```

`password`

Short name: `-p`

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

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'
```

compute-resource

Short name: `-r`

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

- vCenter Server includes multiple instances of standalone hosts or clusters, or a mixture of standalone hosts and clusters.
- The ESXi host includes multiple resource pools.
- You deployed the virtual container hosts in a specific resource pool in your environment.

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

- To list the virtual container hosts in a specific resource pool on an ESXi host, specify the name of the resource pool:

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

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

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

- To list the virtual container hosts on a vCenter Server with more than one cluster, specify the name of the target cluster:

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

- To list the virtual container hosts in a specific resource pool on a standalone host that is managed by vCenter Server, specify the IPv4 address or FQDN of the target host and name of the resource pool:

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

- To list the virtual container hosts in a specific resource pool in a cluster, specify the names of the target cluster and the resource pool:

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

debug

Short name: `-v`

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

```
--debug 1
```

Remove a Virtual Container Host

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

Prerequisites

You have deployed a virtual container host that you no longer require.

Procedure

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

   The following example includes the options required to remove a named instance of a virtual container host from a vCenter Server environment.

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

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

   CAUTION Running `vic-machine delete` with the `--force` option removes all running container VMs that the virtual container host manages.

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

Virtual Container Host Delete Options

The command line utility for vSphere Integrated Containers, `vic-machine`, provides a `delete` command that allows you to cleanly remove virtual container hosts. The options that `vic-machine delete` requires depend on the location in your vSphere environment in which you deployed the virtual container host.

`target`

Short name: `-t`

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

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

  ```
  --target esxi_host_address
  ```

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

  ```
  --target vcenter_server_address
  ```

- You can optionally include the user name and password of the ESXi host or vCenter Server 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_or_vcenter_server_username:password@esxi_or_vcenter_server_address
  ```

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

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

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

`user`

Short name: `-u`

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

```
--user esxi_or_vcenter_server_username
```

`password`

Short name: `-p`

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

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'
```

compute-resource

Short name: `-r`

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

- vCenter Server includes multiple instances of standalone hosts or clusters, or a mixture of standalone hosts and clusters.
- The ESXi host includes multiple resource pools.
- You deployed the virtual container host in a specific resource pool in your environment.

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

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

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

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

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

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

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

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

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

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

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

name

Short name: `-n`

The name of the virtual container host appliance to delete. This option is mandatory if the virtual container host to delete has a name other than the default name, `virtual-container-host`. Specify `--name` with exactly the same value that you used when you ran `vic-machine create`.

```
--name vch_appliance_name
```

force

Short name: `-f`

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

- If you do not specify `force` and the virtual container host contains running container VMs, the deletion fails with a warning.
- If you do not specify `force` and the virtual container host has volume stores, the deletion of the virtual container host succeeds without deleting the volume stores. The list of volume stores appears in the `vic-machine delete` success message for reference and optional manual removal.

```
--force
```

timeout

Short name: none

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

```
--timeout 5m0s
```

debug

Short name: `-v`

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

```
--debug 1
```