Using vSphere Integrated Containers as a DevOps Administrator, Developer, or Viewer

VMware vSphere Integrated Containers 1.5.x

vmware

Table of Contents

Using vSphere Integrated Containers as a DevOps Administrator, Developer, or Viewer

Using vSphere Integrated Containers as a DevOps Administrator, Developer, or Viewer provides information about how to use VMware vSphere® Integrated Containers™ Management Portal as a user with the DevOps administrator, Developer, or Viewer role.

Product version: 1.5

This documentation applies to all 1.5.x releases.

Intended Audience

This information is intended for vSphere Integrated Containers users who have the DevOps Administrator, Developer, or Viewer roles in vSphere Integrated Containers Management Portal. Knowledge of container technology and Docker is assumed.

VMware, Inc. 3401 Hillview Ave. Palo Alto, CA 94304

www.vmware.com

Running Development Projects

Running Development Projects in vSphere Integrated Containers Management Portal

As a developer, you can perform the following tasks in vSphere Integrated Containers Management Portal:

- Provision containers. For more information, see Provisioning Container VMs in the Management Portal.
- Add networks and volumes to virtual container hosts. For more information, see Create New Networks for Provisioning Containers.
- View the repositories and virtual container hosts for your project.
- Create applications.

As a DevOps administrator, in addition to developer tasks, you can perform the following tasks in vSphere Integrated Containers Management Portal:

- Add developers and viewers projects and assign other DevOps administrators. For more information, see Add Viewers, Developers, or DevOps Administrators to Projects.
- Change project configurations, such as making the project registry public, changing deployment security settings, and enabling vulnerability scanning. For more information, see Configure Project Settings.

Provisioning Container VMs in the Management Portal

You can provision containers or container VMs from the management portal depending on the target host. If your target host is a VCH, you provision container VMs. If your target host is a Docker host, you provision standard containers. You can quick-provision by using default settings or you can customize your deployment by using the available settings. You can either provision or save as a template your configured container.

IMPORTANT: vSphere Integrated Containers Management Portal allows you to provision containers from the registries that are included in the lists of global registries that the Management Portal Administrator configures, or project registries that the DevOps administrator configures. However, if the vSphere administrator deployed a VCH with whitelist mode enabled, and if the whitelist on the VCH is more restrictive than the global and project registry lists, you can only provision containers from the registries that the VCH permits in its whitelist, even if the VCH is included in a project that permits other registries. For more information, see VCH Whitelists and Registry Lists in vSphere Integrated Containers Management Portal in *vSphere Integrated Containers for vSphere Administrators*.

You can provision containers, templates, or images.

- To provision a single container, go to **Home** > **Containers** and click **+ Container**.
- To provision an image with additional settings, go to **Home** > **Templates** and import a new template from file that you can later provision.

When you create containers from the Containers page in the management portal, you can configure the following settings:

- Basic configuration
  - Image to be used
  - Name of the container
  - Custom commands
  - Links
- Network configuration
  - Port bindings and ports publishing
  - Hostname
  - Network mode
- Storage configuration
  - Select volumes
  - Configure a working directory
- Policy configuration
  - Define clusters
  - Resource allocation
  - Anti-affinity rules
- Custom environment variables
- Health checks
  - HTTP
  - TCP connection
  - Command
- Logging

When you configure a container, on the **Environment** tab, you can add industry standard variables. For information about using Docker environment variables, see Environment variables in Compose in the Docker documentation.

Related topics

- Configuring Links
- Configuring Health Checks
- Configuring Cluster Size and Scale

Configuring Links

You configure links to templates or images. You can use links to enable communication between multiple services in your application. Links in vSphere Integrated Containers are similar to Docker links, but connect containers across hosts. Alink consists of two parts: a service name and an alias. The service name is the name of the service or template being called. The alias is the hostname that you use to communicate with that service.

For example, if you have an application that contains a Web and database service and you define a link in the Web service to the database service by using an alias of *my-db*, the Web service application opens a TCP connection to *my-db:PORT_OF_DB*. The *PORT_OF_DB* is the port that the database listens to, regardless of the public port that is assigned to the host by the container settings. If MySQL is checking for updates on its default *3306* port, and the published port for the container host is *32799*, the Web application accesses the database at *my-db:3306*.

You can use networks instead of links. Links are a legacy Docker feature with significant limitations when linking container clusters, including:

- Docker does not support multiple links with the same alias.
- You cannot update the links of a container runtime. When scaling up or down a linked cluster, the dependent container's links will not be updated.

Configuring Health Checks

You can configure a health check method to update the status of a container based on custom criteria. vSphere Integrated Containers uses an own implementation of health checks and not the standard Docker implementation.

You can use HTTP or TCP protocols when executing a command on the container. You can also specify a health check method. The available health configuration modes are described below.

Mode	Description
None	Default. No health checks are configured.
HTTP	If you select HTTP, you must provide an API to access and an HTTP method and version to use. The API is relative and you do not need to enter the address of the container. You can also specify a timeout period for the operation and set health thresholds. For example, a healthy threshold of 2 means that two consecutive successful calls must occur for the container to be considered healthy and in the RUNNING status. An unhealthy threshold of 2 means that two unsuccessful calls must occur for the container to be considered unhealthy and in the ERROR status. For all the states in between the healthy and unhealthy thresholds, the container status is DEGRADED.
TCP connection	If you select TCP connection, you must only enter a port for the container. The health check attempts to establish a TCP connection with the container on the provided port. You can also specify a timeout value for the operation and set healthy or unhealthy thresholds as with HTTP.
Command	If you select Command, you must enter a command to be run on the container. The success of the health check is determined by the exit status of the command.

You can also enable a health check as part of the provisioning process for a container. By default, health checks are not performed during provisioning. Deselect the **Ignore health check on provision** check box to require at least one successful health check before a container can be considered successfully provisioned.

When a container returns an ERROR status, you can configure an automated redeploy for that container by selecting the **Autoredeploy** check box.

Configuring Cluster Size and Scale

You can create container clusters by using Containers placement settings to specify cluster size.

When you configure a cluster, a specified number of containers is provisioned. Requests are load balanced among all containers in the cluster. You can modify the cluster size on a provisioned container or application to increase or decrease the size of the cluster by one. When you modify the cluster size at runtime, all affinity filters and placement rules are considered.

Create New Networks for Provisioning Containers

You can create, modify, and attach network configurations to containers and container templates.

Procedure

1. In the management portal, navigate to **Home** > **Networks** and click **+ Network**.
2. On the Create Network page, select the **Advanced** check box to access all available settings.
3. Configure the new network settings and click **Create**.

Setting	Description
Name	Enter a name for the network.
IPAM config	Enter subnet, IP range, and gateway values that are unique to this network configuration. They must not overlap with any other networks on the same container host.
Custom Properties	(Optional) Specify custom properties for the new network configuration. `containers.ipam.driver` - for use with containers only. Specifies the IPAM driver to be used when adding a network component. The supported values depend on the drivers that are installed in the container host environment in which they are used. For example, a supported value might be infoblox or calico depending on the IPAM plug-ins that are installed on the container host. This property name and value are case-sensitive. The property value is not validated when you add it. If the specified driver does not exist on the container host at provisioning time, an error message is returned and provisioning fails. `containers.network.driver` - for use with containers only. Specifies the network driver to be used when adding a network component. The supported values depend on the drivers that are installed in the container host environment in which they are used. By default, Docker-supplied network drivers include bridge, overlay, and macvlan, while VCH-supplied network drivers include the bridge driver. Third-party network drivers such as weave and calico might also be available, depending on what networking plug-ins are installed on the container host. This property name and value are case-sensitive. The property value is not validated when you add it. If the specified driver does not exist on the container host at provisioning time, an error message is returned and provisioning fails.
Hosts	Select the hosts to use the new network.

Result

New network is created and you can provision containers on it.