VeloCloud Virtual Edge

Deployment Guide

Introduction

Overview

The Virtual Edge is available as a virtual machine that can be installed on standard hypervisors. See the sections below for more information on how to install the Virtual Edge on KVM and VMware ESXi hypervisors.

- Option 1: Virtual Edge installation on KVM
- Option 2: Virtual Edge installation on VMware ESXi

Prerequisites & Instance Requirements

Virtual Edge Specification

- 2 x Intel vCPUs with AES-NI instruction set
- 4Gb of memory
- Virtual disk (approximately 8 Gb of disk space)
- 3 to 8 vNICs (default is 2 x L2 interfaces and 6 x L3 interfaces)

Firewall/NAT Requirements

If the VeloCloud virtual Edge is deployed behind the Firewall and/or NAT device, the following applies:

- The Firewall must allow outbound traffic from the VeloCloud Virtual Edge to TCP/443 (for communication with the VeloCloud Orchestrator).

- The Firewall needs to allow traffic outbound to Internet on ports UDP/2426 (VCMP).

Other Considerations

- The VeloCloud Edge is a latency sensitive application. Please consult the VMware documentation to tune the Virtual Machine for latency sensitive application.

- For best performance, set the CPU scheduling affinity to dedicate CPU cores to the Virtual Edge and turn on Intel Virtualization Technology (Intel VT) on the hypervisor.
- For best performance, VMware and KVM should be set with SR-IOV support. KVM instructions are provided in the document below. For VMware, please see: https://pubs.vmware.com/vsphere-51/index.jsp#com.vmware.vsphere.networking.doc/GUID-C5043E19-F84D-4E2E-9162-16D9967C2DB8.html

Cloud-init Overview

Cloud-init is a Linux package responsible for handling early initialization of instances. If available in the distributions, it allows for configuration of many common parameters of the instance directly after installation. This creates a fully functional instance that is configured based on a series of inputs. This mode of installation requires two files, meta-data and user-data.

Cloud-init's behavior can be configured via user-data. User-data can be given by the user at the time of launching the instance. This is typically done by attaching a secondary disk in ISO format that the cloud-init will look for at first boot time. This disk contains all early configuration data that will be applied at that time.

The VeloCloud Virtual Edge supports cloud-init and all essential configurations packaged in an ISO image.

Create Cloud-init Meta-data and User Data Files

Follow the steps below to create the Cloud-init meta-data file and the user-data file:

Step 1. Create the Cloud-init Meta-data File and User-data Files.

The final installation configuration options are set with a pair of cloud-init configuration files. The first installation configuration file contains the metadata. Create this file with a text editor and name it meta-data. This file provides information that identifies the instance of the VeloCloud Virtual Edge being installed. The instance-id can be any identifying name, and the local-hostname should be a host name that follows your site standards, for example:

```
instance-id: vedge1
local-hostname: vedge1
network-interfaces:
    GE3:
        type: static
        ipaddr: 10.1.0.2
        netmask: 255.255.255.0
        gateway: 10.1.0.1
```

Step 1a: Create the meta-data file which contains the instance name and WAN configuration. Only WAN interfaces that require static IP addressing need to be specified here. By default, all VCE WAN interfaces are configured for DHCP. Multiple interfaces can be specified.

Step 1b: Create the user-data file. This file contains three main modules: VCO, Activation Code, and Ignore Certificates Errors.

- vco – this is the IP Address/URL of the VCO.
- activation_code - this is the activation code for the virtual Edge. The activation code is generated while creating an Edge instance on the VCO.
- vco_ignore_cert_errors - this is the option to verify or ignore any certificate validity errors.

The activation code is generated while creating an Edge instance on the VCO.

```
#cloud-config

velocloud:

  vce:

    vco: 10.32.0.3

    activation_code: F54F-GG4S-XGFI

    vco_ignore_cert_errors: true
```

Step 2. Create the ISO file

Once you have completed your files, they need to be packaged into an ISO image. This ISO image is used as a virtual configuration CD with the virtual machine. This ISO image (called seed.iso in the example below), is created with the following command on Linux system:

```
genisoimage -output seed.iso -volid cidata -joliet -rock user-data meta-data
```

Once the ISO image is generated, transfer the image to a datastore on the host machine.

Option One: Virtual Edge installation on KVM

The installation on KVM option describes the installation and activation of the Virtual Edge on KVM using a Cloud-init config file. The Cloud-init config will have interface configurations and the activation key of the Edge. The Virtual Edge has been tested on host OS Ubuntu 14.04.LTS with KVM version 2.0

Considerations

- KVM provides multiple ways to provide networking to virtual machines. The following have been used by VeloCloud
  - SR-IOV
  - Linux Bridge
  - OpenVSwitch Bridge

Validating or Enabling SR-IOV

This section is only necessary if you need to enable SR-IOV on the HOST. To properly validate that SR-IOV is ready to be used, you can verify this by typing:

```
lspci | grep -i ethernet
```

Verify that you have Virtual Functions:

```
01:10.0 Ethernet controller: Intel Corporation 82599 Ethernet
Controller Virtual Function (rev 01)
```

If you don't have Virtual Functions, but you have a NIC that supports Virtual Functions, you will need to enable it.

Generally, enabling SR-IOV consists of the following in three steps:

1. Enable SR-IOV in BIOS.

This will be dependent on your BIOS. Login to the BIOS console and look for SR-IOV Support/DMA. You can verify support on prompt by checking that Intel has the correct CPU flag.

```
cat /proc/cpuinfo | grep vmx
```

2. Add the Options on Boot.

```
GRUB_CMDLINE_LINUX="intel_iommu=on"
```

(in /etc/default/grub)

a) After this Run command, "update-grub" and "update-initramfs -u"
b) Reboot and make sure iommu is enabled.

```
velocloud@KVMperf3:~$ dmesg | grep -i IOMMU

[ 0.000000] Command line: BOOT_IMAGE=/vmlinuz-3.13.0-107-generic
root=/dev/mapper/qa--multiboot--002--vg-root ro intel_iommu=on splash
quiet vt.handoff=7
```

```
[ 0.000000] Kernel command line: BOOT_IMAGE=/vmlinuz-3.13.0-107-
generic root=/dev/mapper/qa--multiboot--002--vg-root ro intel_iommu=on
splash quiet vt.handoff=7

[ 0.000000] Intel-IOMMU: enabled

[ 0.083191] dmar: IOMMU 0: reg_base_addr fbffc000 ver 1:0 cap
d2078c106f0466 ecap f020de

[ 0.083197] dmar: IOMMU 1: reg_base_addr c7ffc000 ver 1:0 cap
d2078c106f0466 ecap f020de

velocloud@KVMperf3:~$
```

3. Add the ixgbe Driver in Linux.

https://downloadcenter.intel.com/download/14687/Intel-Network-Adapter-Driver-for-PCIe-Intel-10-Gigabit-Ethernet-Network-Connections-Under-Linux-

a) On the left section of the Intel Website (**Other Versions** section), click the 5.2.1 link.
b) Download ixgbe from intel. Follow compile options
c) Configure ixgbe config (tar and sudo make install)

```
velocloud@KVMperf1:~$ cat /etc/modprobe.d/ixgbe.conf
```

c) If the file doesn't exist, create it.

```
options ixgbe max_vfs=32,32

options ixgbe allow_unsupported_sfp=1

options ixgbe MDD=0,0

blacklist ixgbevf
```

d) Remember to execute the command "`update-initramfs -u`" and reboot.
e) Use modinfo to see if it is property installed.

```
velocloud@KVMperf1:~$ modinfo ixgbe and ip link

filename: /lib/modules/4.4.0-62-
generic/updates/drivers/net/ethernet/intel/ixgbe/ixgbe.ko

version: 5.0.4

license: GPL

description: Intel(R) 10GbE PCI Express Linux Network Driver

author: Intel Corporation, <linux.nics@intel.com>

srcversion: BA7E024DFE57A92C4F1DC93
```

After rebooting the VM you should see the interfaces.

Detailed Installation Steps

The following steps explain how to run VeloCloud Virtual Edge on KVM using the libvirt. This deployment was tested in Ubuntu 14.04LTS

Step 1 Use gunzip to extract the qcow2 file to the image location, e.g. /var/lib/libvirt/images

Step 2 Create the Network pools that you are going to use for the device. Provided below sample on pool using SR-IOV and pool using OpenVswitch

SR-IOV Sample

```
<network>

  <name>sriovpool</name> <!--This is the name of the file you created-
->

  <forward mode='hostdev' managed='yes'>

    <pf dev='eth1'/> <!--Use the netdev name of your SR-IOV devices PF
here-->

  </forward>

</network>
```

OpenVSwitch Sample

```
<network>
 <name>passthrough</name>
   <model type='virtio'/>
 <forward mode="bridge"/>
 <bridge name="passthrough"/>
 <virtualport type='openvswitch'>
         </virtualport>
 <vlan trunk='yes'>
    <tag id='33' nativeMode='untagged'/>
     <tag id='200'/>
    <tag id='201'/>
    <tag id='202'/>
 </vlan>
</network>
```

Bridge

```
<network>
 <name>passthrough</name>
   <model type='virtio'/>
 <forward mode="bridge"/>

</network>
```

Note: SR-IOV and OpenvSwitch allow VLAN support. Linux Bridge does not allow for this

```
<domain type='kvm'>
<name>vedge1</name>
<memory unit='KiB'>4194304</memory>
<currentMemory unit='KiB'>4194304</currentMemory>
<vcpu placement='static'>2</vcpu>
<resource>
   <partition>/machine</partition>
</resource>
<os>
   <type arch='x86_64' machine='pc-i440fx-trusty'>hvm</type>
   <boot dev='hd'/>
</os>
<features>
   <acpi/>
   <apic/>
   <pae/>
            </features>
            <!--
              Set the CPU mode to host model to leverage
              all the available features on the host CPU
            -->
            <cpu mode='host-model'>
<model fallback='allow'/>
```

```
            </cpu>
            <clock offset='utc'/>
            <on_poweroff>destroy</on_poweroff>
            <on_reboot>restart</on_reboot>
            <on_crash>restart</on_crash>
            <devices>
<emulator>/usr/bin/kvm-spice</emulator>
<!--
                Below is the location of the qcow2 disk image
-->
<disk type='file' device='disk'>
               <driver name='qemu' type='qcow2'/>
          <source file='/var/lib/libvirt/images/edge-
          VC_KVM_GUEST-x86_64-2.3.0-18- R23-20161114-GA-
          updatable-ext4.qcow2'/>
               <target dev='sda' bus='sata'/>
               <address type='drive' controller='0' bus='0'
               target='0' unit='0'/>
</disk>
<!--
               If using cloud-init to boot up virtual
               edge, attach the 2nd disk as CD-ROM
-->
<disk type='file' device='cdrom'>
               <driver name='qemu' type='raw'/>
               <source file='/home/vcadmin/cloud-
               init/vedge1/seed.iso'/>
               <target dev='sdb' bus='sata'/>
               <readonly/>
               <address type='drive' controller='1' bus='0'
               target='0' unit='0'/>
</disk>

<controller type='usb' index='0'>
               <address type='pci' domain='0x0000' bus='0x00'
               slot='0x01' function='0x2'/>
</controller>
<controller type='pci' index='0' model='pci-root'/>
<controller type='sata' index='0'>
               <address type='pci' domain='0x0000' bus='0x00'
               slot='0x05' function='0x0'/>
</controller>
<controller type='ide' index='0'>
               <address type='pci' domain='0x0000' bus='0x00'
               slot='0x01' function='0x1'/>
</controller>
```

```
<!--
               The first two interfaces are for the default L2
               interfaces, NOTE VLAN support just for SR-IOV and
               OpenvSwitch
-->
               <interface type='network'>
                <model type='virtio'/>
                <source network='LAN1'/>
                <vlan><tag id='#hole2_vlan#'/></vlan>
                <alias name=LAN1/>
                <address type='pci' domain='0x0000' bus='0x00'
               slot='0x12' function='0x0'/>
               </interface>
               <interface type='network'>
                <model type='virtio'/>
                <source network=LAN2/>
                <vlan><tag id='#LAN2_VLAN#'/></vlan>
                <alias name='hostdev1'/>
                <address type='pci' domain='0x0000' bus='0x00'
               slot='0x13' function='0x0'/>
               </interface>

<!--
               The next two interfaces are for the default L3
               interfaces. Note that additional 6 routed
               interfaces are supported for a combination of 8
               interfaces total
-->

               <interface type='network'>
                <model type='virtio'/>
                <source network=WAN1/>
                <vlan><tag id='#hole2_vlan#'/></vlan>
                <alias name=LAN1/>
                <address type='pci' domain='0x0000' bus='0x00'
               slot='0x12' function='0x0'/>
               </interface>
               <interface type='network'>
                <model type='virtio'/>
                <source network=LAN2/>
                <vlan><tag id='#LAN2_VLAN#'/></vlan>
                <alias name='hostdev1'/>
                <address type='pci' domain='0x0000' bus='0x00'
               slot='0x13' function='0x0'/>
               </interface>
<serial type='pty'>
               <target port='0'/>
</serial>
```

```
<console type='pty'>
               <target type='serial' port='0'/>
</console>
<input type='mouse' bus='ps2'/>
<input type='keyboard' bus='ps2'/>
<graphics type='vnc' port='-1' autoport='yes' listen='127.0.0.1'>
               <listen type='address' address='127.0.0.1'/>
</graphics>
<sound model='ich6'>

      <address type='pci' domain='0x0000' bus='0x00' slot='0x04'
      function='0x0'/>
    </sound>
    <video>
      <model type='cirrus' vram='9216' heads='1'/>
      <address type='pci' domain='0x0000' bus='0x00' slot='0x02'
      function='0x0'/>
    </video>
    <memballoon model='virtio'>
      <address type='pci' domain='0x0000' bus='0x00' slot='0x06'
      function='0x0'/>
    </memballoon>
</devices>
 </domain>
```

Step 3 Save the above domain XML file, e.g. vedge1.xml and run ***virsh define vedge1.xml*** to create the VM.

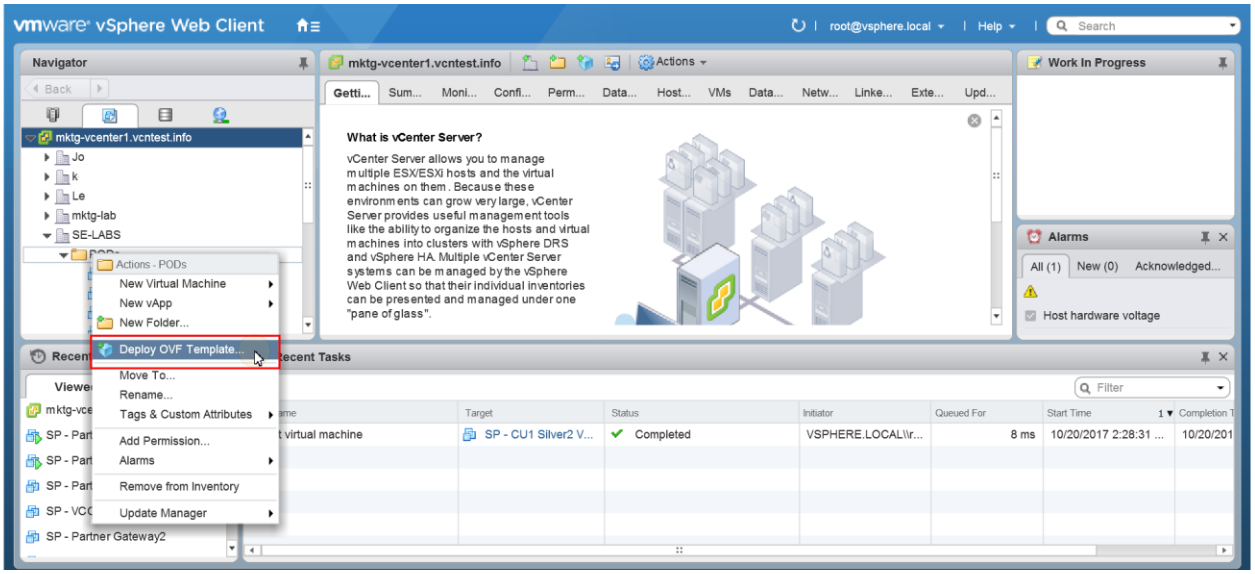
Step 4 Start the VM by ***virsh start vedge1***. Note that ***vedge1*** is the name of the VM defined in the <name> element of the domain XML file. Replace ***vedge1*** with the name you specify in the <name> element.

The Cloud-init already includes the activation key, which was generated while creating a new Virtual Edge on the VCO. The Virtual Edge is configured with the config settings from the Cloud-init file. This will configure the interfaces as the Virtual Edge is powered up. Once the Virtual Edge is online, it will activate with the VCO using the activation key. The VCO IP address and the activation key have been defined in the Cloud-init file.

Option Two: Virtual Edge installation on VMware ESXi

Detailed Installation Steps

Step 1 Use the vSphere client to deploy an OVF template, and then select the VCE OVA file.

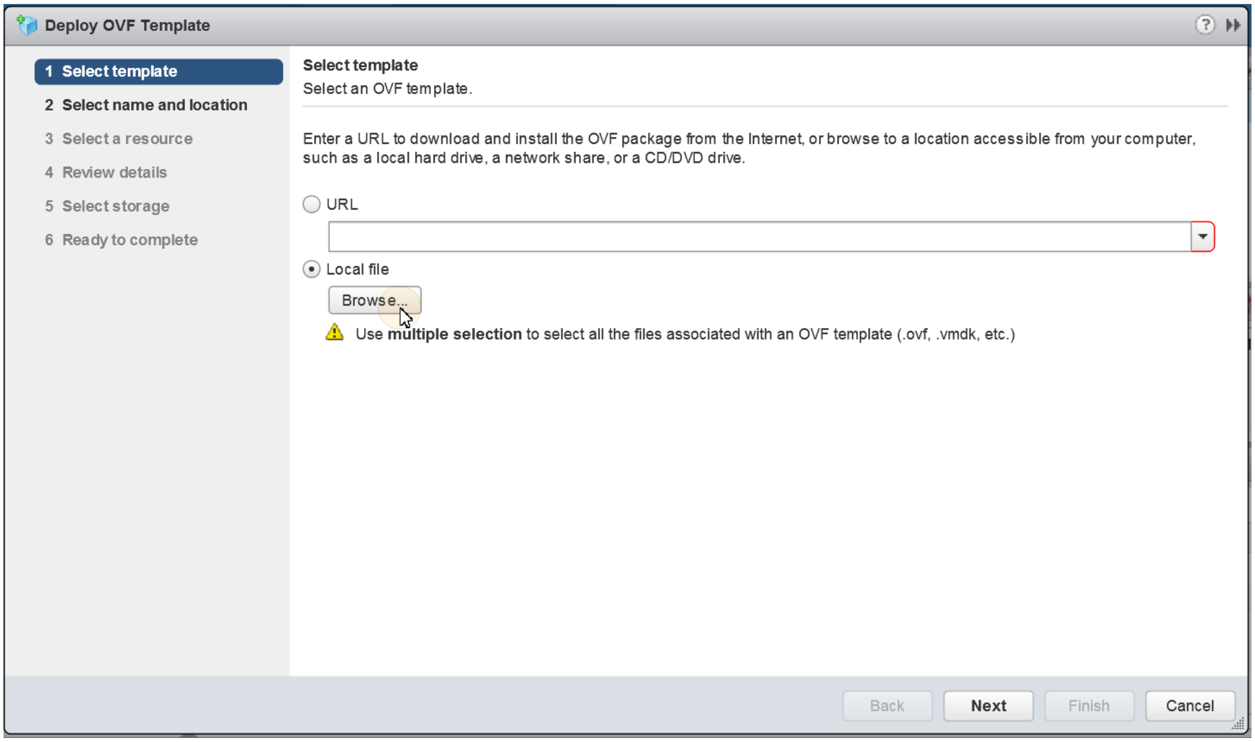


What is vCenter Server?

vCenter Server allows you to manage multiple ESX/ESXi hosts and the virtual machines on them. Because these environments can grow very large, vCenter Server provides useful management tools like the ability to organize the hosts and virtual machines into clusters with vSphere DRS and vSphere HA. Multiple vCenter Server systems can be managed by the vSphere Web Client so that their individual inventories can be presented and managed under one "pane of glass".

me	Target	Status	Initiator	Queued For	Start Time	Completion T
t virtual machine	SP - CU1 Silver2 V...	✓ Completed	VSPHERE.LOCAL\\r...	8 ms	10/20/2017 2:28:31 ...	10/20/201

Step 2 Select an OVF template from a URL or Local file.



Select template

Select an OVF template.

Enter a URL to download and install the OVF package from the Internet, or browse to a location accessible from your computer, such as a local hard drive, a network share, or a CD/DVD drive.

○ URL

● Local file

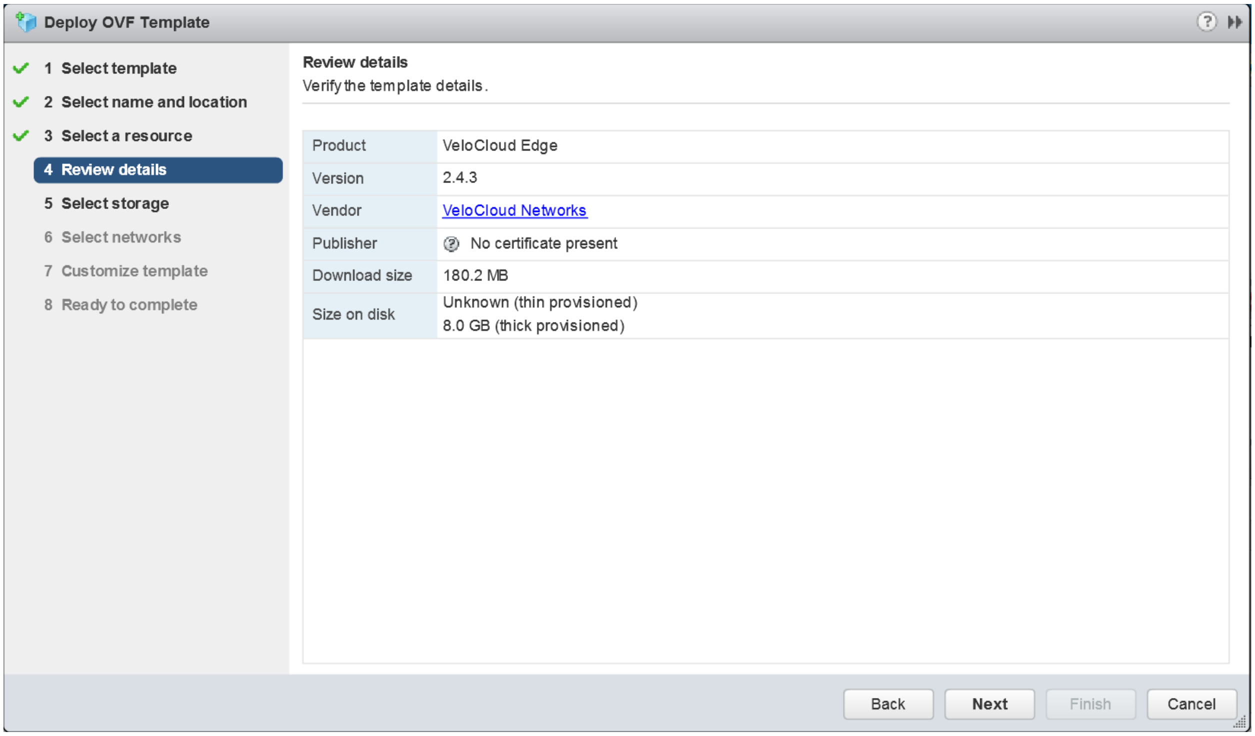
Browse...

⚠ Use **multiple selection** to select all the files associated with an OVF template (.ovf, .vmdk, etc.)

Step 3 Select a name and location of the virtual machine.

Step 4 Select a resource.

Step 5 Verify the template details.

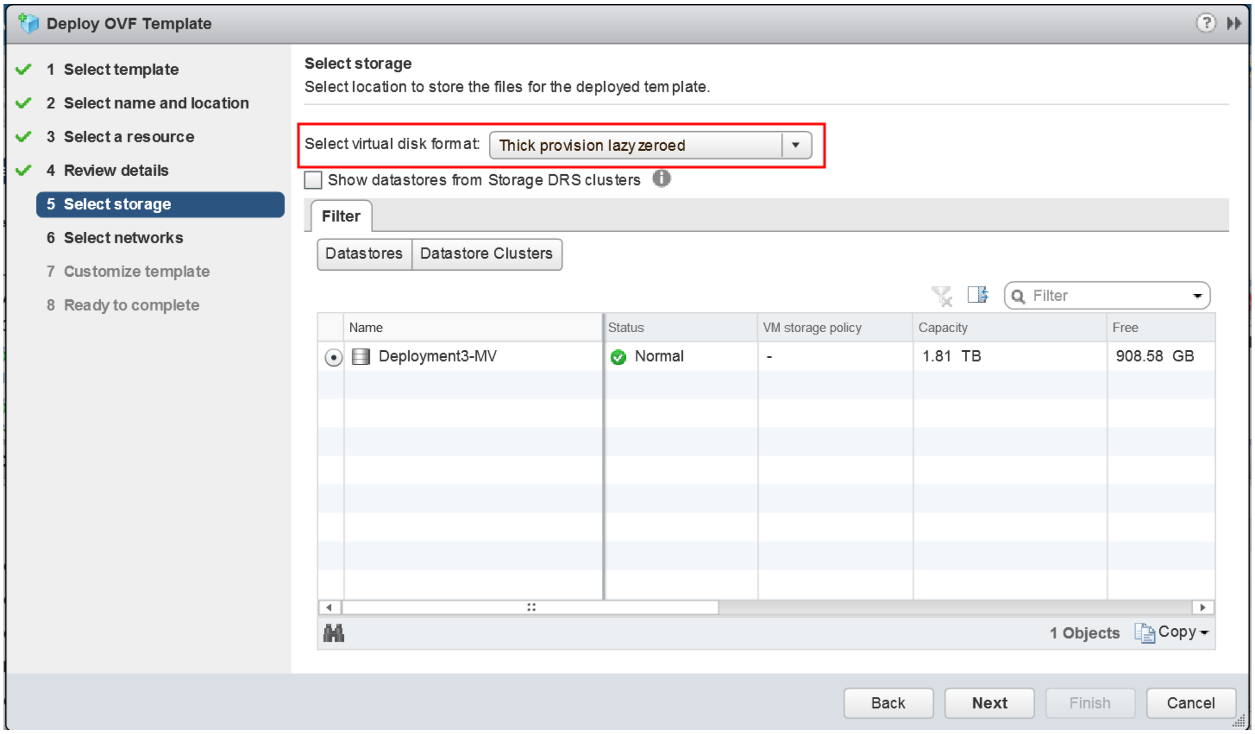


Review details

Verify the template details.

Product	VeloCloud Edge
Version	2.4.3
Vendor	VeloCloud Networks
Publisher	No certificate present
Download size	180.2 MB
Size on disk	Unknown (thin provisioned) 8.0 GB (thick provisioned)

Step 6 Select the storage location to store the files for the deployment template.



Select storage

Select location to store the files for the deployed tem plate.

Select virtual disk format: Thick provision lazy zeroed

□ Show datastores from Storage DRS clusters

Filter

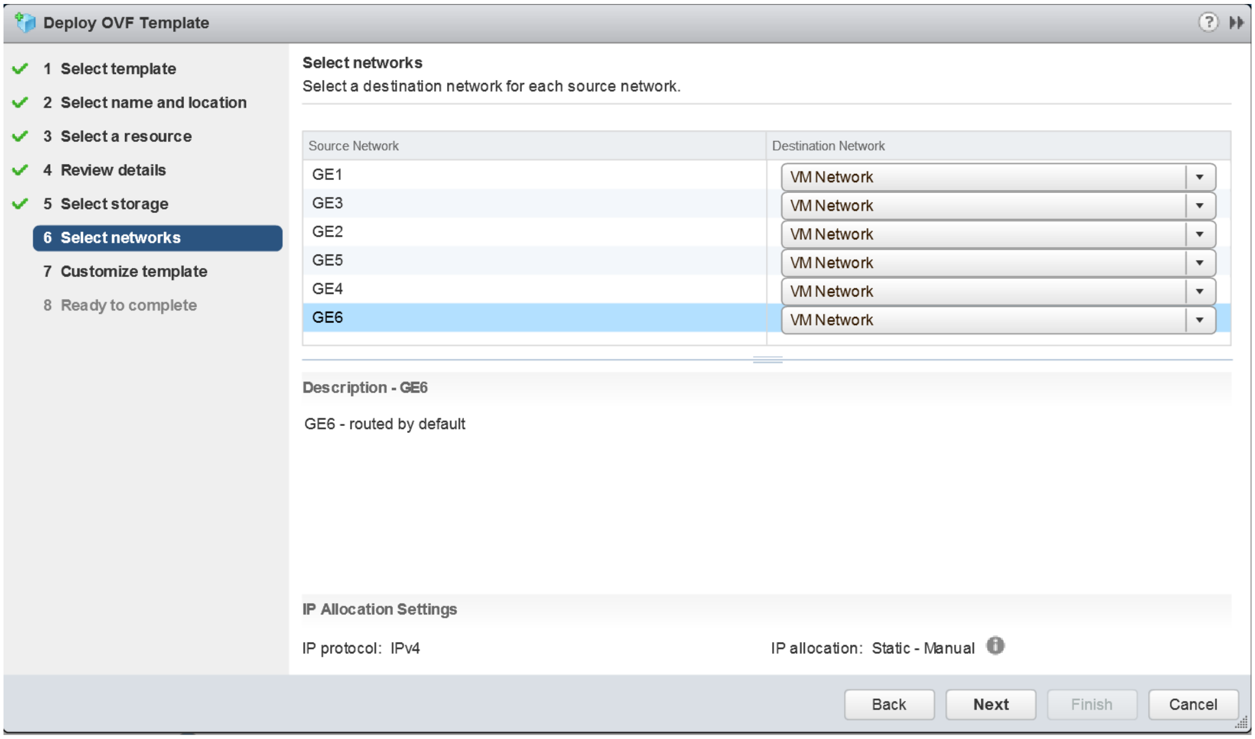
Datastores | Datastore Clusters

	Name	Status	VM storage policy	Capacity	Free
●	Deployment3-MV	✓ Normal	-	1.81 TB	908.58 GB

1 Objects Copy

NOTE: Please skip step 7 below, if using a cloud-init file to provision the Virtual Edge on ESXi.

Step 7 Configure the networks for each of the interfaces.



Select networks

Select a destination network for each source network.

Source Network	Destination Network
GE1	VM Network
GE3	VM Network
GE2	VM Network
GE5	VM Network
GE4	VM Network
GE6	VM Network

Description - GE6

GE6 - routed by default

IP Allocation Settings

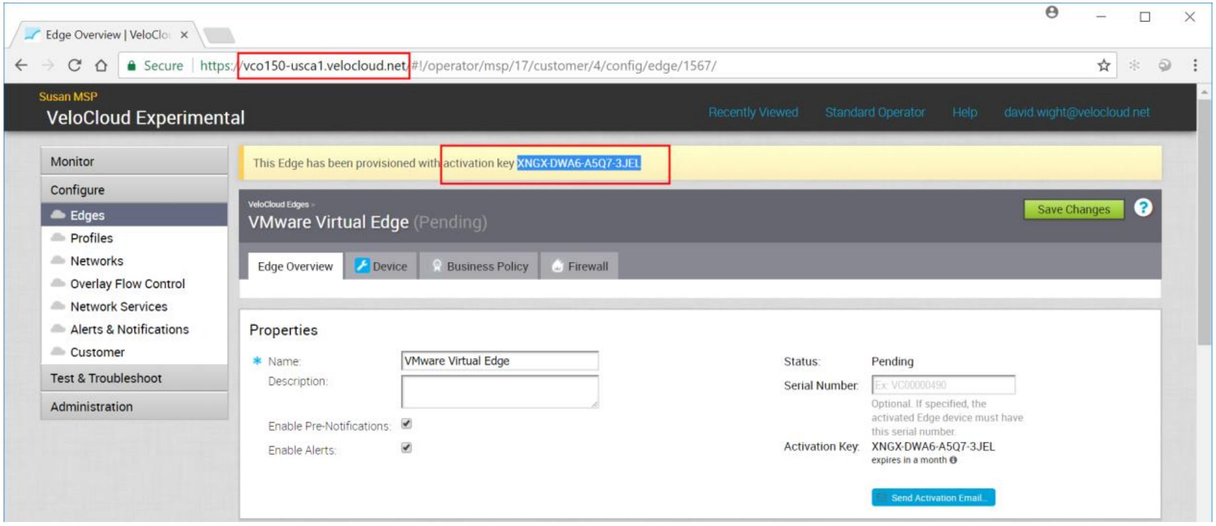
IP protocol: IPv4

IP allocation: Static - Manual

Step 8 Customize the template by specifying the deployment properties. See the image below of the VCO which highlights for the following substeps.

a) From the VCO UI, retrieve the VCO URL/IP Address.

b) Create a new Virtual Edge on the VCO for the Enterprise. Once the Edge is created, copy the Activation Key.



https://vco150-usca1.velocloud.net/#!/operator/msp/17/customer/4/config/edge/1567/

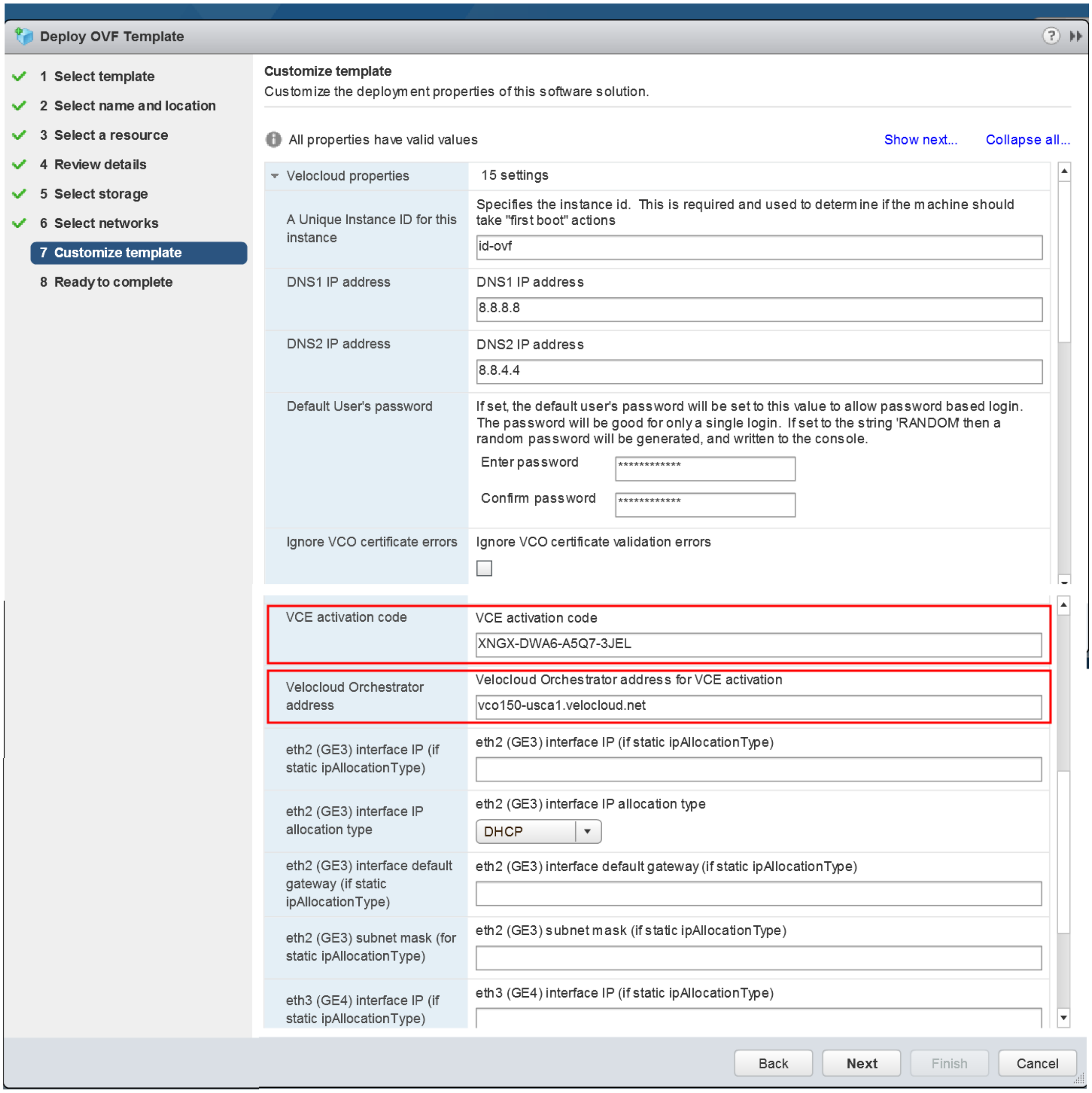
This Edge has been provisioned with activation key XNGX-DWA6-A5Q7-3JEL

VMware Virtual Edge (Pending)

Properties

* Name:	VMware Virtual Edge	Status:	Pending
Description:		Serial Number:	Ex: VC00000490 Optional. If specified, the activated Edge device must have this serial number.
Enable Pre-Notifications:	☑		
Enable Alerts:	☑	Activation Key:	XNGX-DWA6-A5Q7-3JEL expires in a month

c) On the customize template page shown below, type in the Activation Code retrieved in step b) and the VCO URL/IP Address retrieved in step a) above, into the corresponding fields.



Deploy OVF Template

- ✓ 1 Select template
- ✓ 2 Select name and location
- ✓ 3 Select a resource
- ✓ 4 Review details
- ✓ 5 Select storage
- ✓ 6 Select networks
- 7 Customize template
- 8 Ready to complete

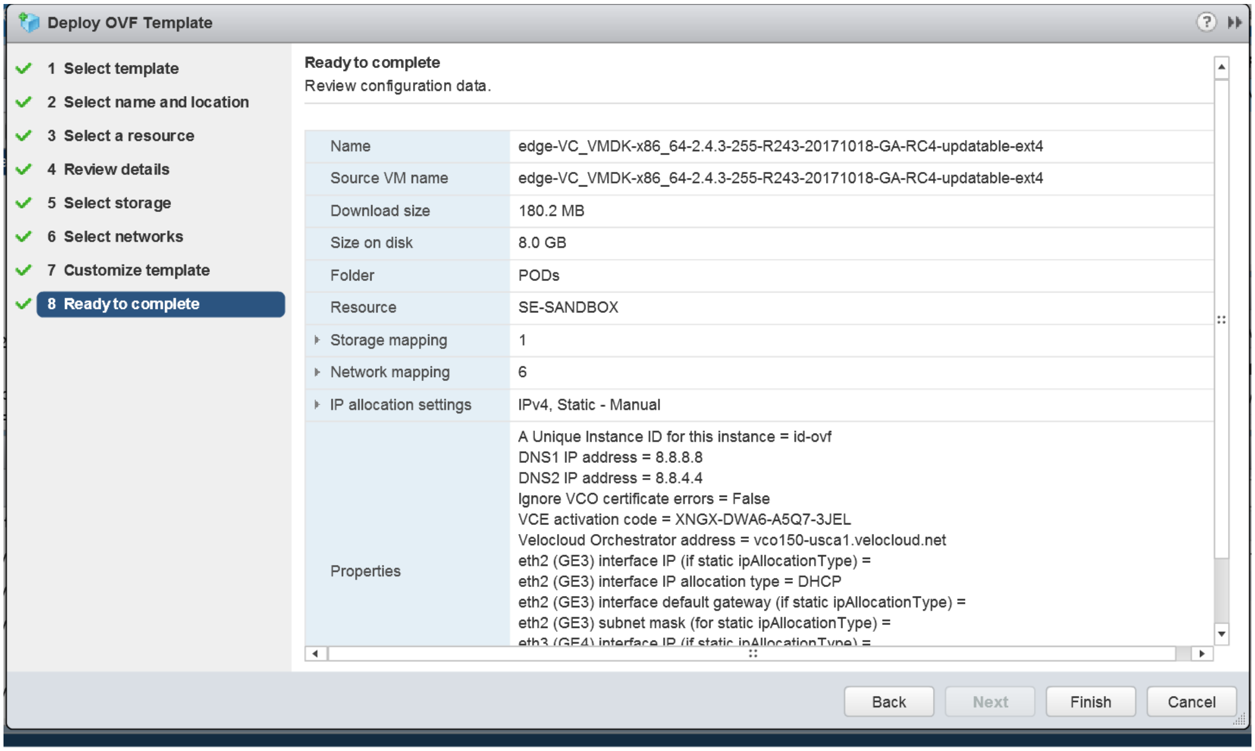
Customize template

Customize the deployment properties of this software solution.

All properties have valid values

Velocloud properties	15 settings
A Unique Instance ID for this instance	Specifies the instance id. This is required and used to determine if the machine should take "first boot" actions id-ovf
DNS1 IP address	DNS1 IP address 8.8.8.8
DNS2 IP address	DNS2 IP address 8.8.4.4
Default User's password	If set, the default user's password will be set to this value to allow password based login. The password will be good for only a single login. If set to the string 'RANDOM' then a random password will be generated, and written to the console. Enter password ************ Confirm password ************
Ignore VCO certificate errors	Ignore VCO certificate validation errors □
VCE activation code	VCE activation code XNGX-DWA6-A5Q7-3JEL
Velocloud Orchestrator address	Velocloud Orchestrator address for VCE activation vco150-usca1.velocloud.net
eth2 (GE3) interface IP (if static ipAllocationType)	eth2 (GE3) interface IP (if static ipAllocationType)
eth2 (GE3) interface IP allocation type	eth2 (GE3) interface IP allocation type DHCP
eth2 (GE3) interface default gateway (if static ipAllocationType)	eth2 (GE3) interface default gateway (if static ipAllocationType)
eth2 (GE3) subnet mask (for static ipAllocationType)	eth2 (GE3) subnet mask (if static ipAllocationType)
eth3 (GE4) interface IP (if static ipAllocationType)	eth3 (GE4) interface IP (if static ipAllocationType)

Step 9 Review the configuration data.



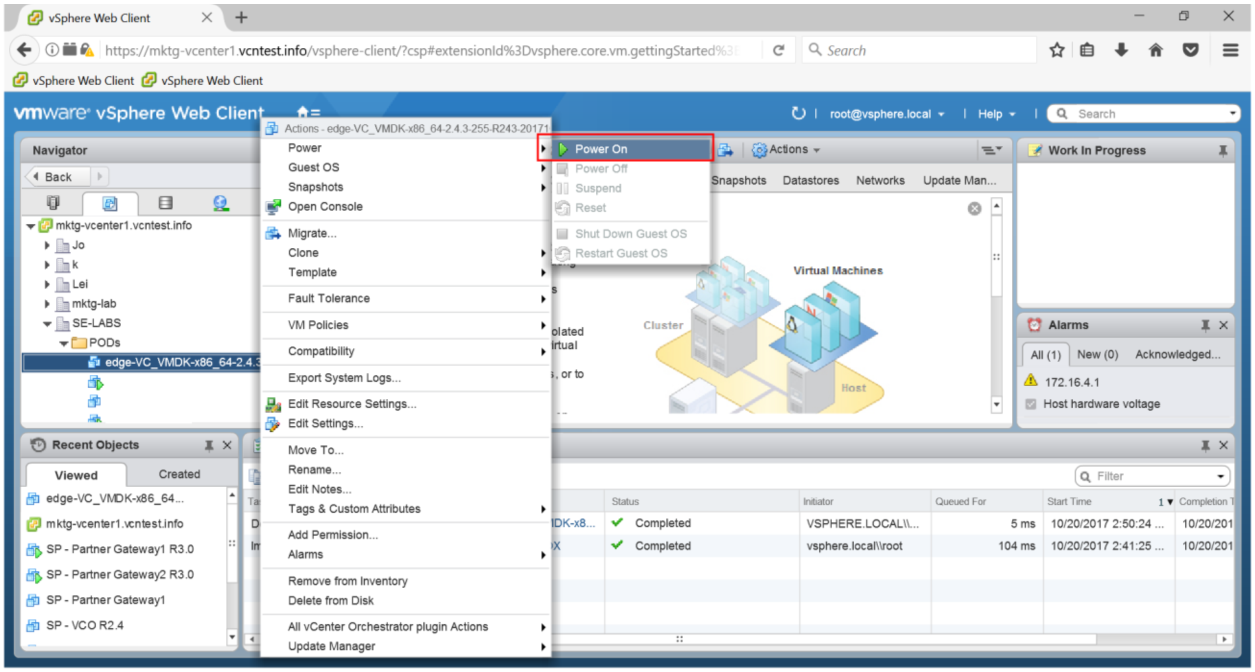
Deploy OVF Template

1 Select template
2 Select name and location
3 Select a resource
4 Review details
5 Select storage
6 Select networks
7 Customize template
8 Ready to complete

Ready to complete
Review configuration data.

Name	edge-VC_VMDK-x86_64-2.4.3-255-R243-20171018-GA-RC4-updatable-ext4
Source VM name	edge-VC_VMDK-x86_64-2.4.3-255-R243-20171018-GA-RC4-updatable-ext4
Download size	180.2 MB
Size on disk	8.0 GB
Folder	PODs
Resource	SE-SANDBOX
Storage mapping	1
Network mapping	6
IP allocation settings	IPv4, Static - Manual
Properties	A Unique Instance ID for this instance = id-ovf DNS1 IP address = 8.8.8.8 DNS2 IP address = 8.8.4.4 Ignore VCO certificate errors = False VCE activation code = XNGX-DWA6-A5Q7-3JEL Velocloud Orchestrator address = vco150-usca1.velocloud.net eth2 (GE3) interface IP (if static ipAllocationType) = eth2 (GE3) interface IP allocation type = DHCP eth2 (GE3) interface default gateway (if static ipAllocationType) = eth2 (GE3) subnet mask (for static ipAllocationType) = eth3 (GE4) interface IP (if static ipAllocationType) =

Step 10 Power on the Virtual Edge.



Status	Initiator	Queued For	Start Time	Completion T
✓ Completed	VSPHERE.LOCAL\\...	5 ms	10/20/2017 2:50:24 ...	10/20/201
✓ Completed	vsphere.local\\root	104 ms	10/20/2017 2:41:25 ...	10/20/201

Step 11 Once the Edge powers up, it will establish connectivity to the VCO.