

VeloCloud Partner Guide

VMware SD-WAN 3.3

You can find the most up-to-date technical documentation on the VMware website at:

<https://docs.vmware.com/>

VMware, Inc.
3401 Hillview Ave.
Palo Alto, CA 94304
www.vmware.com

Copyright © 2021 VMware, Inc. All rights reserved. [Copyright and trademark information.](#)

Contents

- 1** VMware SD-WAN by VeloCloud Release 3.3 5
- 2** Introduction 7
- 3** Log in to VCO Using SSO for Partner User 9
- 4** Monitor Customers 10
- 5** Manage Customers 11
 - Create a Customer 11
- 6** Monitor Events 15
- 7** Manage Admins 16
- 8** View Overview Settings 19
- 9** Configure Partner Settings 20
 - Overview of Single Sign On 20
 - Configure Single Sign On for Partner User 21
- 10** Manage Gateway Pools 24
 - Create a Gateway Pool 25
- 11** Manage Gateways 26
- 12** Configure Single Sign On for Identity Partners 27
 - Configure an IDP for Single Sign On 27
 - Configure Okta for Single Sign On 27
 - Configure OneLogin for Single Sign On 30
 - Configure PingIdentity for Single Sign On 34
 - Configure Azure Active Directory for Single Sign On 37
 - Configure VMware CSP for Single Sign On 43
- 13** Configure Edge Licensing 46
 - Edge Licenses and License Types 46
 - Generate an Edge Licensing Report 47

14	Install the VeloCloud Partner Gateway	48
	Installation Overview	48
	Hypervisor Minimum Hardware Requirements	49
	VeloCloud Gateway Installation Procedures	50
	Pre-Installation Considerations	51
	Install VeloCloud Gateway	58
	Post-Installation Tasks	72
	Upgrade VeloCloud Gateway	75
	Custom Configurations	75
	NTP Configuration	75
	Userdata	76
	OAM Interface and Static Routes	77
	OAM - SR-IOV with vmxnet3 or SR-IOV with VIRTIO	79
	Special Consideration When Using 802.1ad Encapsulation	81
	SNMP Integration	82
	Custom Firewall Rules	83

VMware SD-WAN by VeloCloud Release 3.3

1

The *VMware SD-WAN by VeloCloud Partner Guide release 3.3* includes new and updated content for versions 3.3.0, 3.3.1, and 3.3.2 as described below.

What's Changed in Version 3.3.2?

There are no new updates to the VMware SD-WAN by VeloCloud Partner Guide for the 3.3.2 release.

What's Changed in Version 3.3.1?

Status	Section
New	Overview of Single Sign On
New	Configure Single Sign On for Partner User
New	Chapter 3 Log in to VCO Using SSO for Partner User
New	Configure an IDP for Single Sign On
Updated	Create a Customer

What's Changed in Version 3.3.0?

Status	Section
New	<i>Edge Licensing</i>
New	<i>Push Activation</i>
New	<i>User Agreement</i>
New	<i>Self-service Password Reset</i>

For a complete list of new and updated sections to the documentation for Administrators, see *VMware SD-WAN by VeloCloud Release 3.3* in the *VMware SD-WAN by VeloCloud Administration Guide*.

Previous VMware SD-WAN by VeloCloud Versions

To get product documentation for previous VMware SD-WAN by VeloCloud versions, contact your VMware SD-WAN by VeloCloud representative.

Introduction

2

This guide describes the features accessible by the VeloCloud IT Partner roles. The VeloCloud IT Partner roles provide the functionality needed to create, monitor, and manage customers that use VeloCloud.

Before You Begin

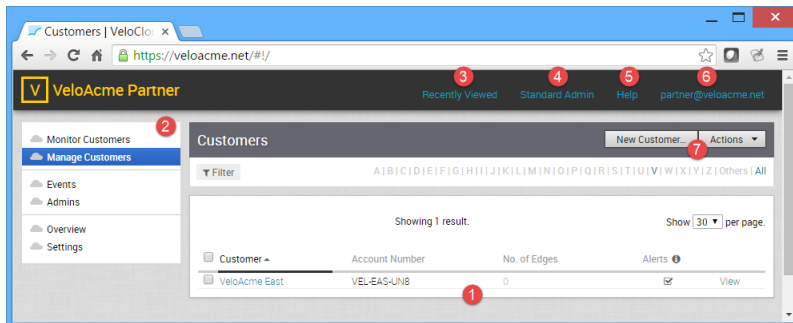
It is assumed that you are thoroughly familiar with the concepts described in the *VeloCloud Overview*. It is also strongly recommended that you read and understand the steps in the *IT Admin Quick Start Guide* and *IT Admin Guide for VeloCloud Orchestrator* to become familiar with the core function of the VeloCloud Orchestrator that is used by an Enterprise IT Administrator for a customer.

About this Guide

In this guide, a hypothetical company, VeloAcme, is used to describe the configuration for customers. This guide also provides steps to monitor, test, and troubleshoot the VeloCloud system.

Initial Partner Page

The following figure shows an example of the initial Partner web page:



Numbers in the figure correspond to the numbers in the following steps:

- 1 The list of the Customers that have been created and that are being managed by the Partner.
- 2 The navigation bar for partner tasks.

- 3 A quick link to the last Customer that was accessed from the partner web page.
- 4 Quick links to Partner functions and the list of recently accessed Customers. If you click on a customer link, you will be taken to the functionality provided for the IT Administrator. For more information, see the *IT Admin Guide for VeloCloud Orchestrator*.
- 5 Quick link to this help.
- 6 The partner that is currently logged in. Clicking the link also provides partner account information and a link to sign out.
- 7 Buttons to create and manage customers.

This guide describes partner functionality in detail.

Log in to VCO Using SSO for Partner User

3

Describes how to log in to VeloCloud Orchestrator (VCO) using Single Sign On (SSO) as a Partner user.

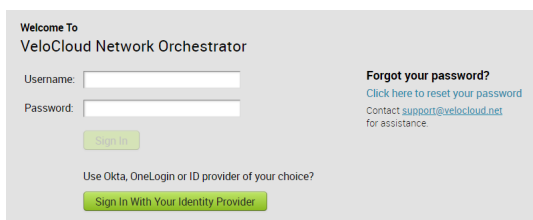
Prerequisites

- Ensure you have configured SSO authentication in VCO. For more information, see [Configure Single Sign On for Partner User](#).
- Ensure you have set up roles, users, and OIDC application for SSO in your preferred IDPs. For more information, see [Configure an IDP for Single Sign On](#).

Procedure

- 1 In a web browser, launch a VCO application as Enterprise or Partner user.

The **VeloCloud Network Orchestrator** screen appears.



The screenshot shows the login interface for the VeloCloud Network Orchestrator. It features a 'Welcome To VeloCloud Network Orchestrator' header. Below this, there are two input fields: 'Username:' and 'Password:'. To the right of these fields is a 'Forgot your password?' link with the text 'Click here to reset your password' and 'Contact support@velocloud.net for assistance.' Below the input fields is a green 'Sign In' button. At the bottom, there is a text prompt 'Use Okta, OneLogin or ID provider of your choice?' followed by a green 'Sign In With Your Identity Provider' button.

- 2 Click **Sign In With Your Identity Provider**.
- 3 In the **Enter your Organization Domain** text box, enter the domain name used for the SSO configuration and click **Sign In**.

The IDP configured for SSO will authenticate the user and redirect the user to the configured VCO URL.

Note Once the users log in to the VCO using SSO, they will not be allowed to login again as native users.

Monitor Customers

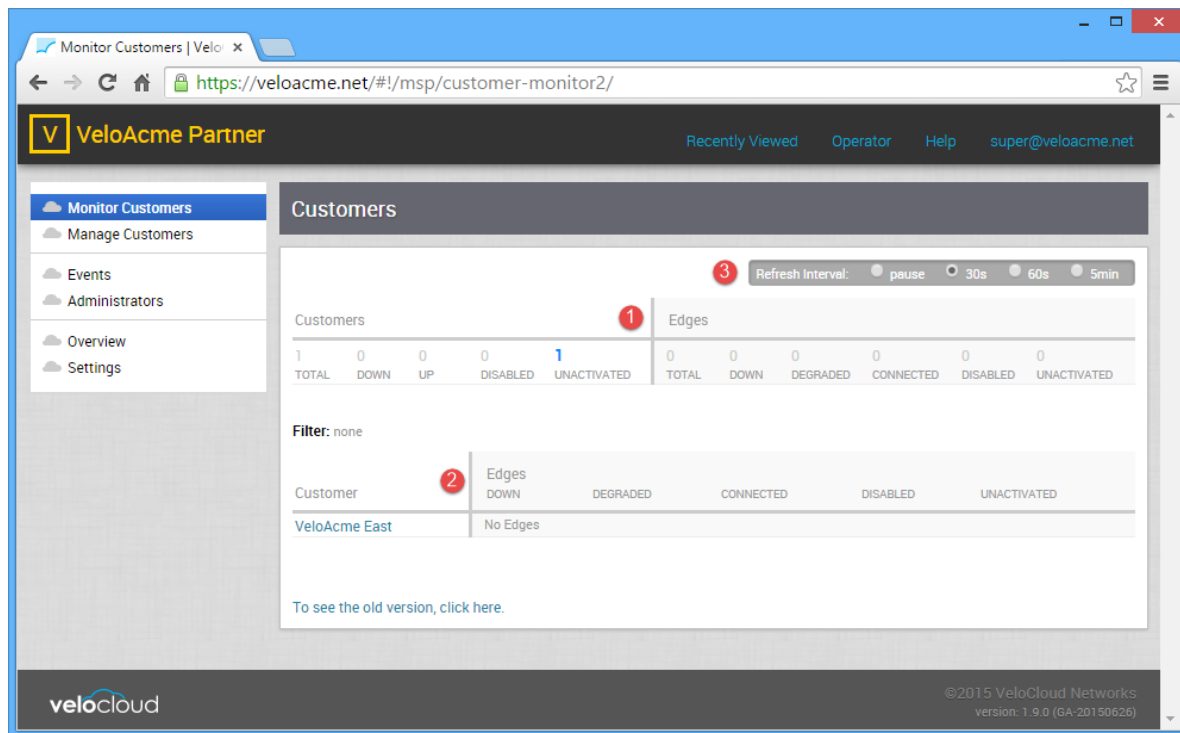
4

A partner can monitor customer status from the **Monitor Customers** link.

To monitor customers:

- In the navigation bar, click **Monitor Customers**.

The **Monitor Customers** page appears.



This screen shows the Edges and Links for all customers managed by this Partner. Selections can be made to control the interval for updating the information.

The major features of the **Monitor Customer** page include:

- 1 An aggregated summary of the status of all customers and their Edges.
- 2 A summary of the status of each customers and their Edges.
- 3 Interval selections that can be made to select a specific monitoring interval.

Manage Customers

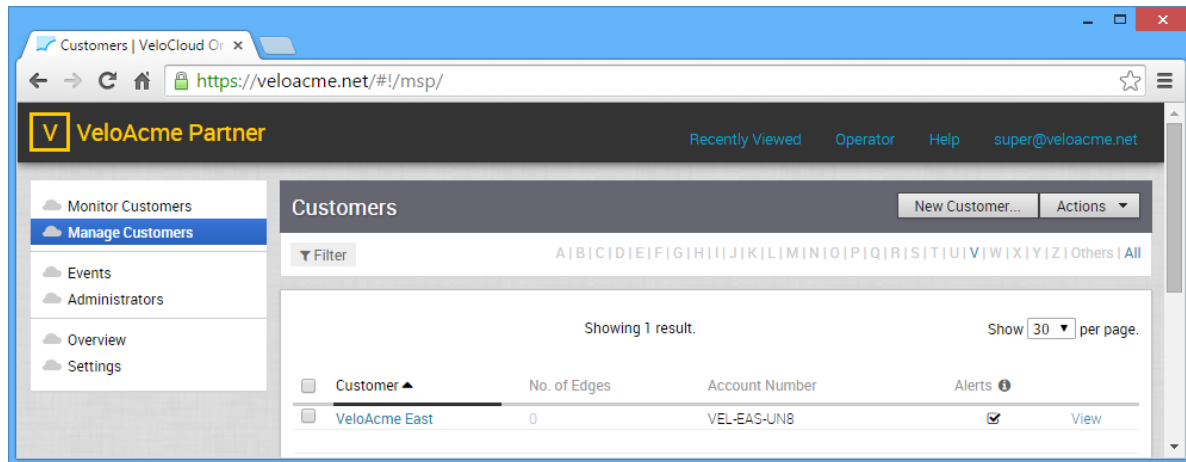
5

A partner can create, modify, and delete a customer account from the **Manage Customers** link. A link is also provided to originate a support email to the customer support staff.

To manage customers:

- In the navigation bar, click **Manage Customers**.

The **Manage Customers** page appears.



This chapter includes the following topics:

- [Create a Customer](#)

Create a Customer

Partner Superusers and Partner Standard Admins can create new customers by clicking the **New Customer** button in the **Customers** screen.

Note Operator Superusers can disable the ability to create a new customer by setting the following system property to true: `session.options.disableCreateEnterpriseProxy`. (One of the most common reasons to use this system property is if the VCO is reaching its usage capacity). When this system property is set to true, Partner Superusers and Partner Standard Admins will not be able to create a new customer from the VCO API or VCO UI. (Setting this system property to true, will not prevent Partner Superusers from creating Partner Admins).

To create a new customer:

- 1 From the VCO navigation panel, click **Manage Customers**.
- 2 On the **Customers** screen, click the **New Customer** button (top, right area of the screen) to create a new customer. The **New Customer** dialog box appears.
- 3 In **New Customer** dialog box, specify the following information:
 - a Type in the **Company Name** and **Account** number in the appropriate fields.
 - b If applicable, select the **Support Access** checkbox for the Partner to grant support access to the Partner.

Note When enabled, Partner Support is granted access to view, configure, and troubleshoot this Customer's Edges. As a security consideration, Partner Support will not be granted access to view user-identifiable information.

- c If applicable, select the **VeloCloud Support Access** checkbox to grant support to VeloCloud Support.

Note When selected, VeloCloud Support is granted access to view, configure and troubleshoot this Customer's Edges. As a security consideration, VeloCloud Support will not be granted access to view user-identifiable information.

- d If applicable, select the **VeloCloud User Management Access** checkbox to grant user management access to VeloCloud Support.

Note When selected, VeloCloud Support is able to assist in user management, including creating users, resetting passwords, and so on. VeloCloud Support is granted access to view all user-identifiable information.

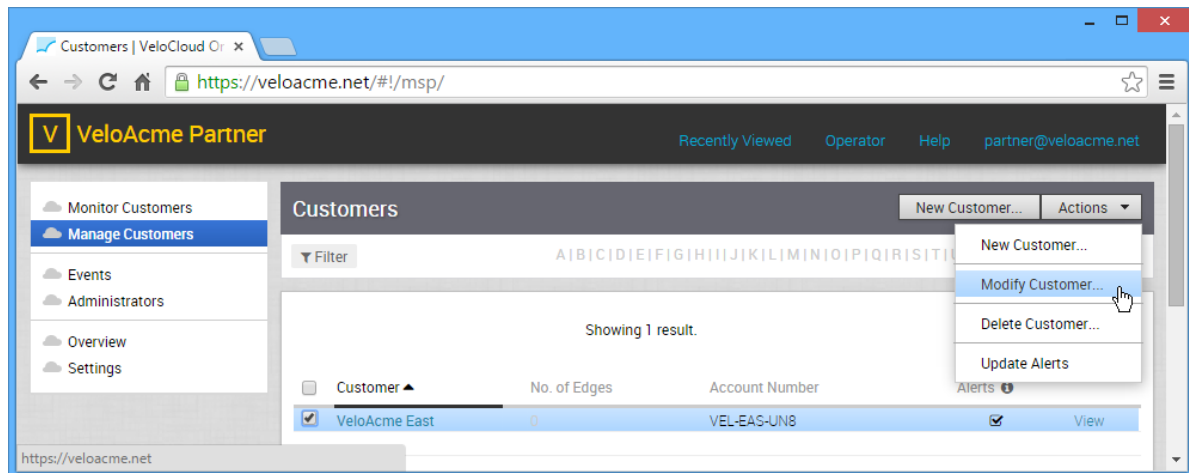
- 4 In the **Initial Admin Account** area, type in the **Username** and **Password** for the new customer in the appropriate text fields.
- 5 Type in other customer information (**First Name**, **Last Name**, **Phone**, **Mobile Phone**, and **Contact Email**).

- 6 In the **Customer Configuration** area, choose a profile from the **Operator Profile** drop-down menu.
- 7 From the **Gateway Pool** drop-down menu, choose a Gateway Pool.
- 8 From the **Default Edge Authentication** drop-down menu, choose **Certificate Disabled**, **Certificate Optional**, or **Certificate Required**.
- 9 In the **Edge Licensing** area, click the **Add** button. VeloCloud recommends that you give your customers access to all license types that match their edition and region. For more information about Edge Licenses, see [Chapter 13 Configure Edge Licensing](#).
- 10 From the **Edge Licenses** dialog box, use the appropriate arrows to select available licenses, and then click **OK**.
- 11 Click the **Create** button to create the customer.

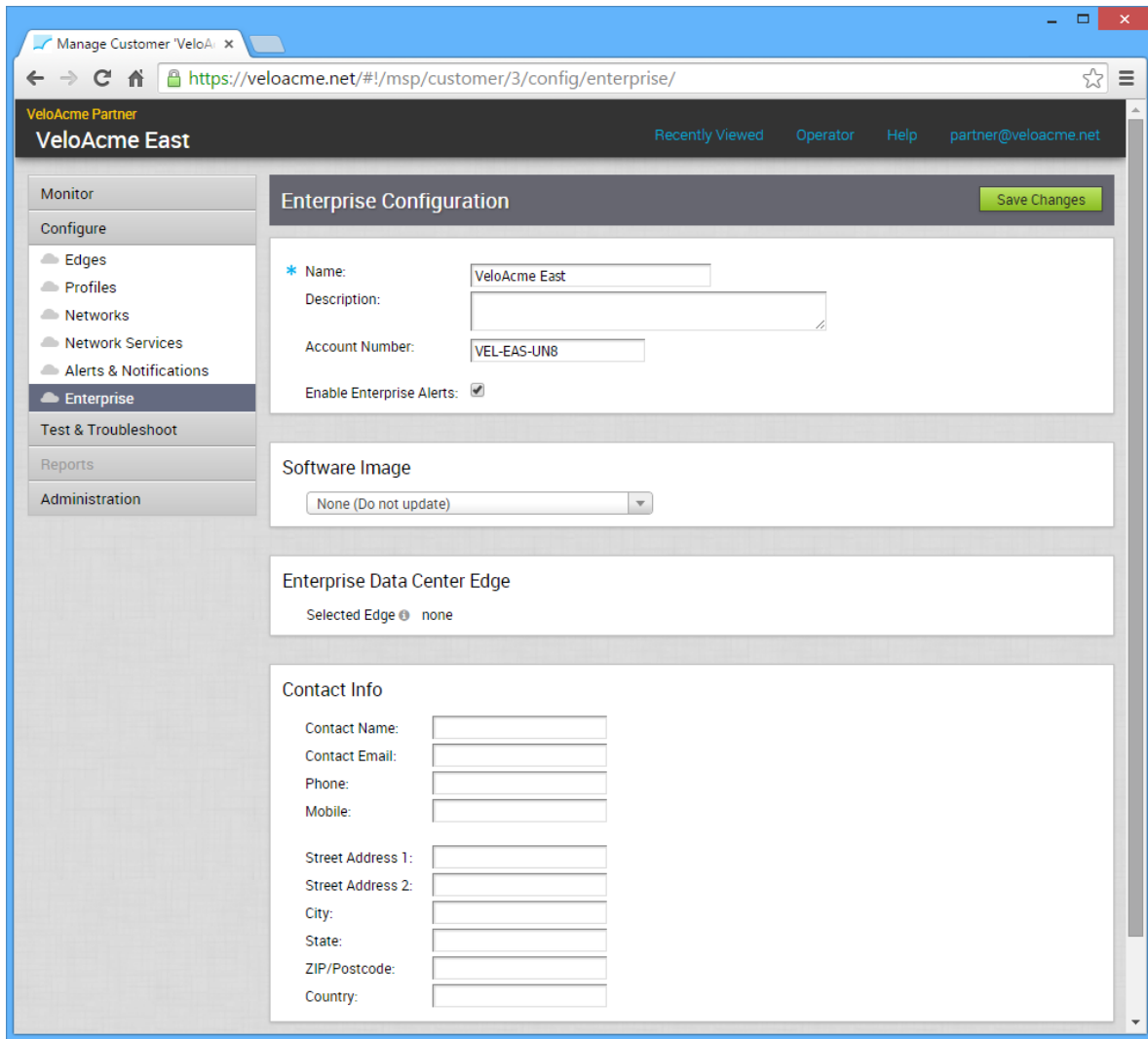
Note If the VeloCloud Support Access option is selected, an VeloCloud Operator with support privileges can configure and troubleshoot the customer's Edges. However, VeloCloud Support will not be able to view user-identifiable information.

Note The Initial Admin Account is given the superuser role. Once the customer is created, additional administrators can be created with other roles.

Once a customer has been created, selections under the **Actions** button can be chosen to delete or modify the customer configuration, or to send a support email.



When the **Modify** action is chosen, the following web page is displayed. The page can be used to update the customer's **Software Image**, **Enterprise Data Center Edge**, and **Contact Info**.



A partner can also access this page for a specific customer from the **Configure -> Enterprise** link.

Monitor Events

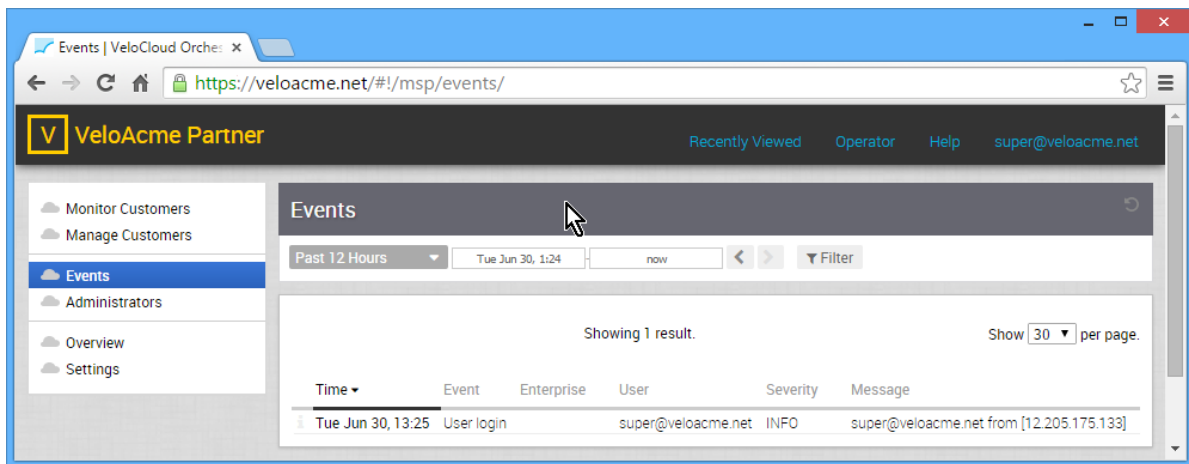
6

A partner can monitor operator events generated by the VeloCloud from the **Events** link.

To view events:

- In the navigation bar, click **Events**.

The **Events** page appears.



These events can help you determine the status of the VeloCloud system. For some events, you can click a link in the event to display more information.

Manage Admins

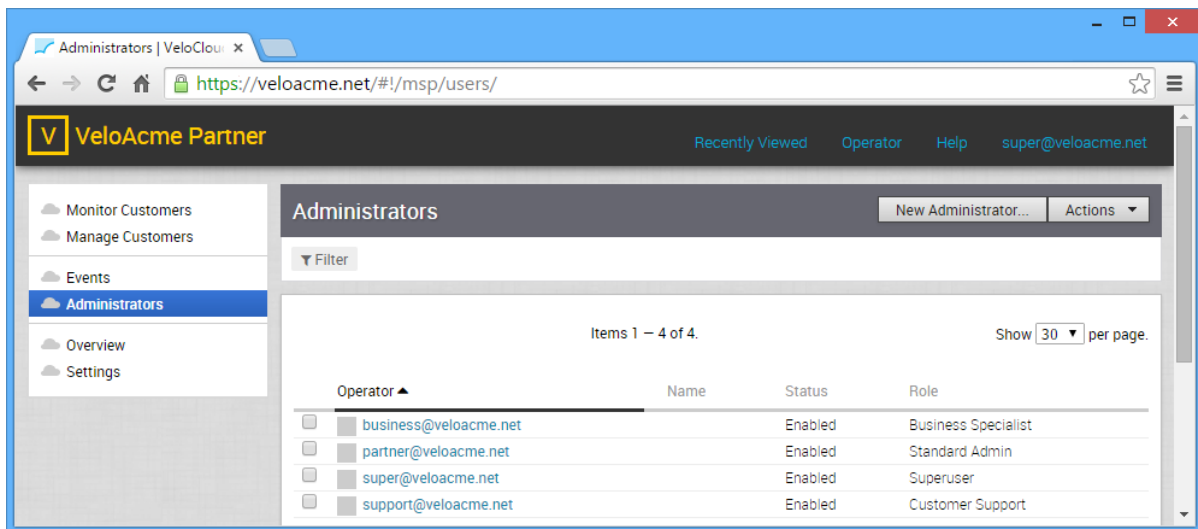
7

A partner can manage administrators from the **Admins** link.

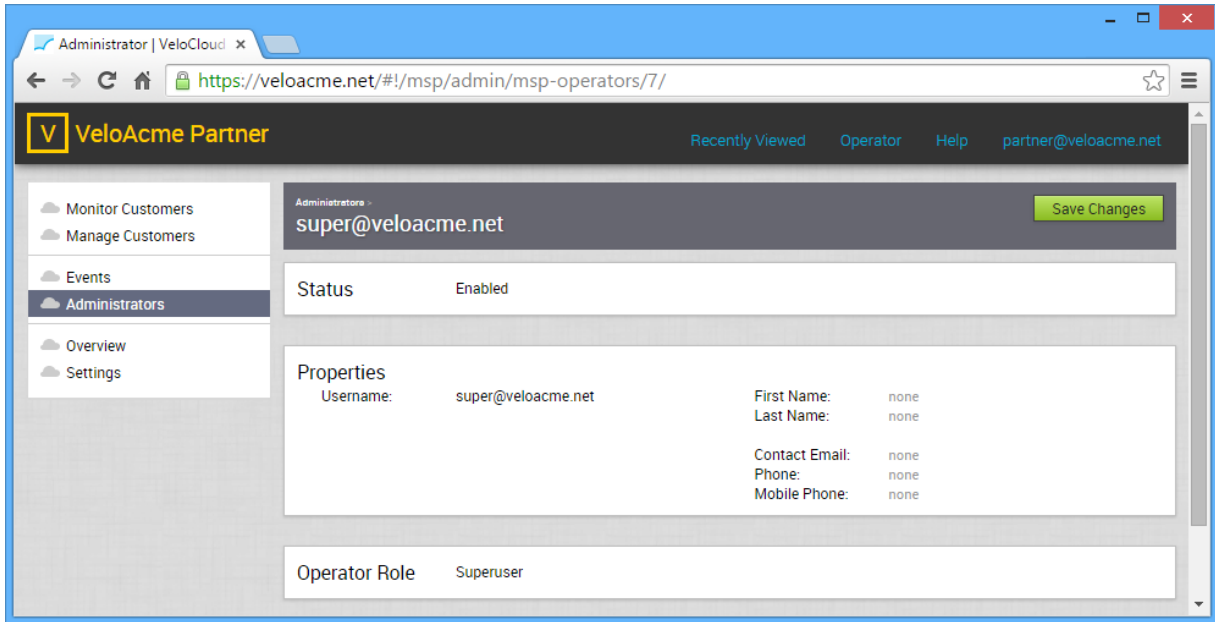
To manage administrators:

- In the navigation bar, click **Admins**.

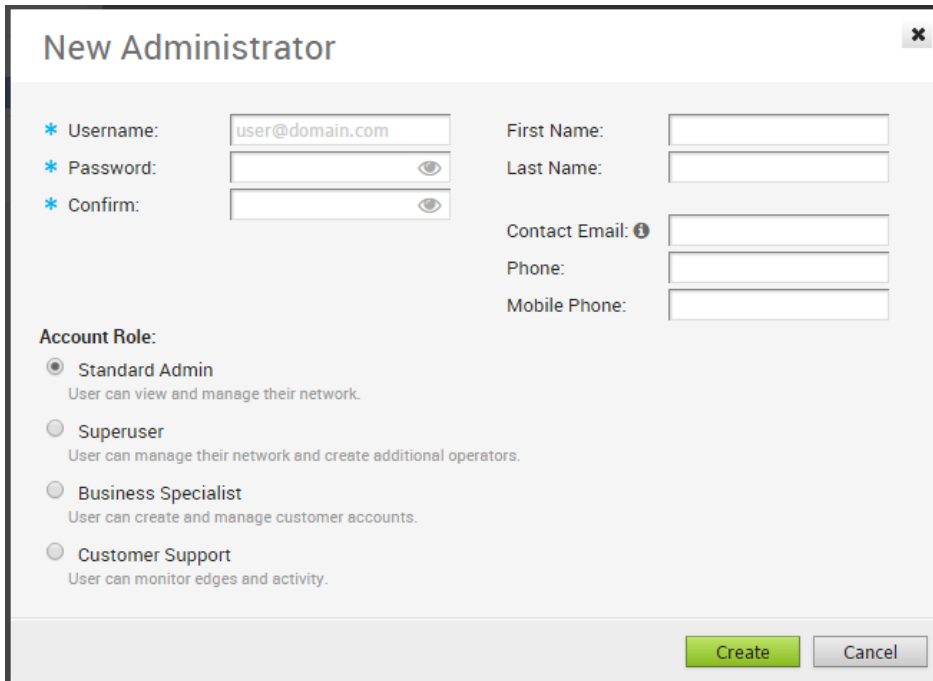
The **Administrators** page appears.



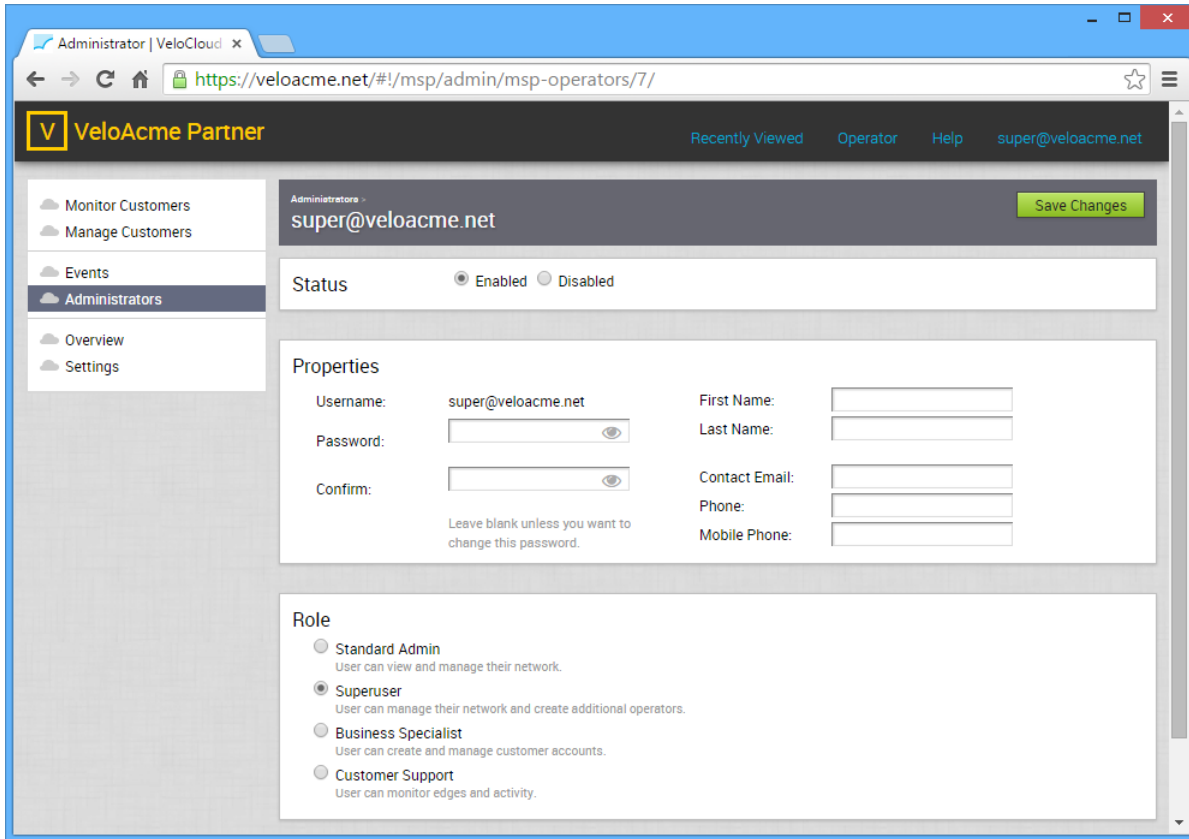
If you select one of the administrators, you can view the details for that administrator.



If you are logged in as a partner with the Superuser role, you can create additional partners and specify their role.



As an administrator with the Superuser role, you can choose an existing operator, and then update the administrator's account details.



View Overview Settings

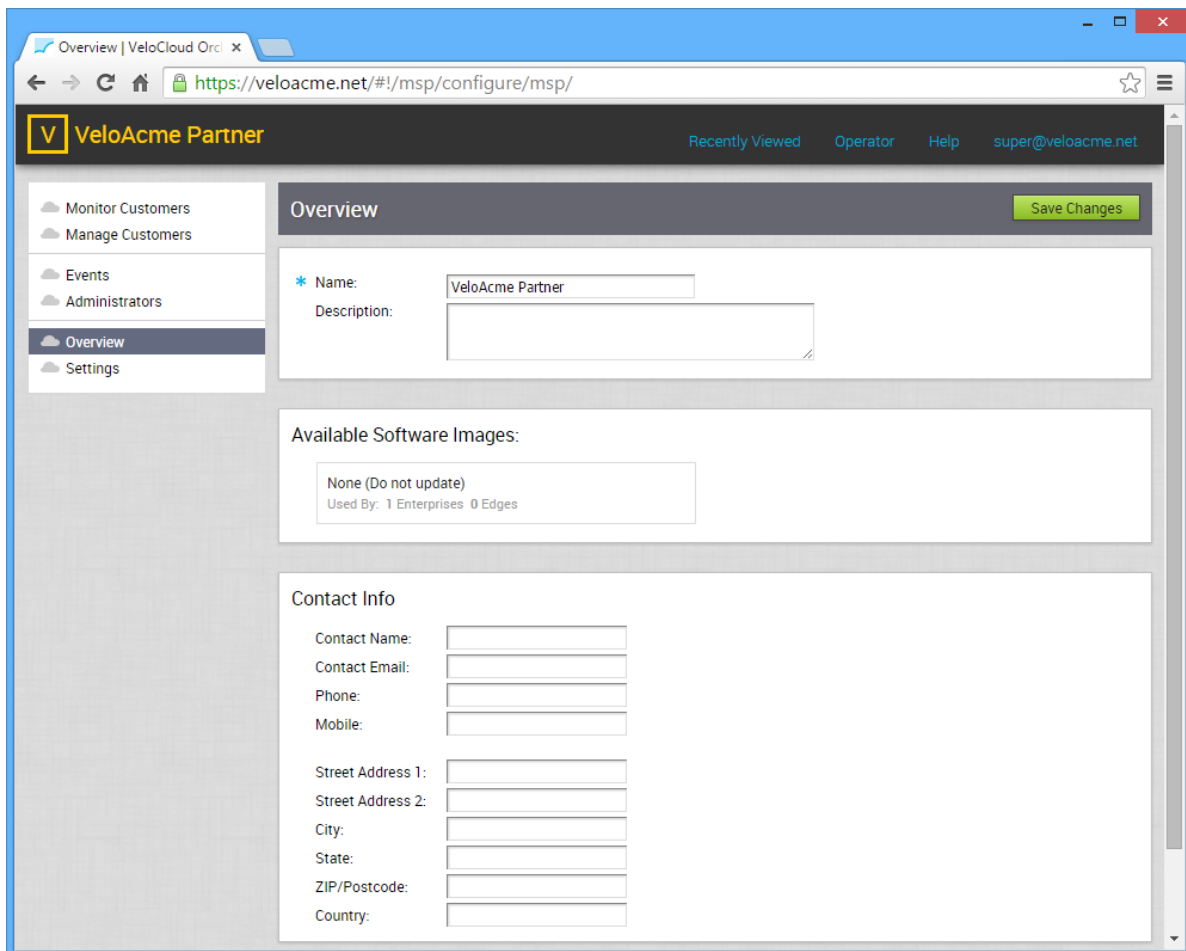
8

A partner can view overview information from the **Overview** link.

To manage software images and modify contact information:

- In the navigation bar, click **Overview**.

The **Overview** page appears.



Configure Partner Settings

9

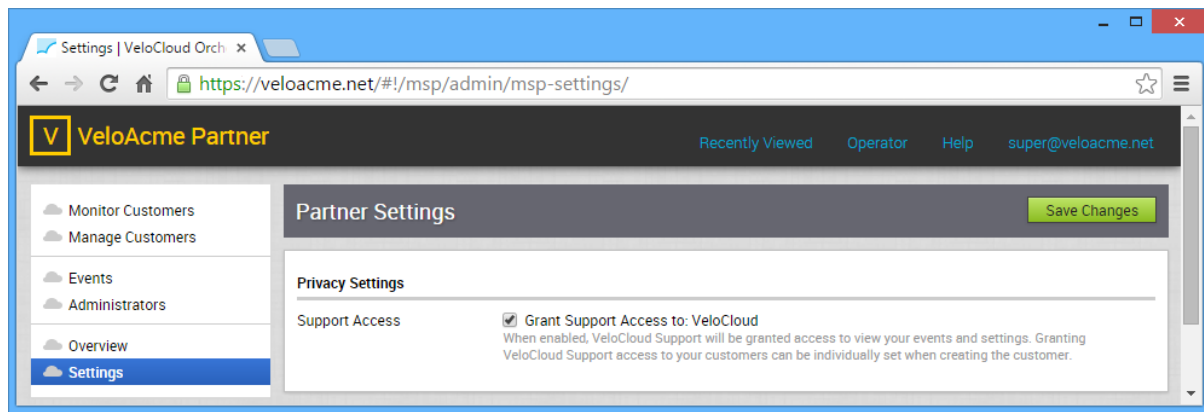
A partner can view and modify partner information (privacy and other settings) and authentication from the **Manage Settings** link.

To manage partner settings:

- In the navigation bar, click **Settings**.

The **Partner Settings** page appears.

Note Only administrators with the Superuser role can modify settings.



This chapter includes the following topics:

- [Overview of Single Sign On](#)
- [Configure Single Sign On for Partner User](#)

Overview of Single Sign On

For the 3.3.1 release, the VeloCloud Orchestrator (VCO) supports a new type of user authentication called Single Sign On (SSO) for all Orchestrator user types: Operator, Partner, and Enterprise.

Single Sign On (SSO) is a session and user authentication service that allows VCO users to log in to the VCO with one set of login credentials to access multiple applications. Integrating the SSO service with VCO improves the security of user authentication for VCO users and enables VCO to authenticate users from other OpenID Connect (OIDC)-based Identity Providers (IDPs). The following IDPs are currently supported:

- Okta
- OneLogin
- PingIdentity
- AzureAD
- VMwareCSP

Configure Single Sign On for Partner User

To setup Single Sign On (SSO) authentication for Partner user, perform the steps on this procedure.

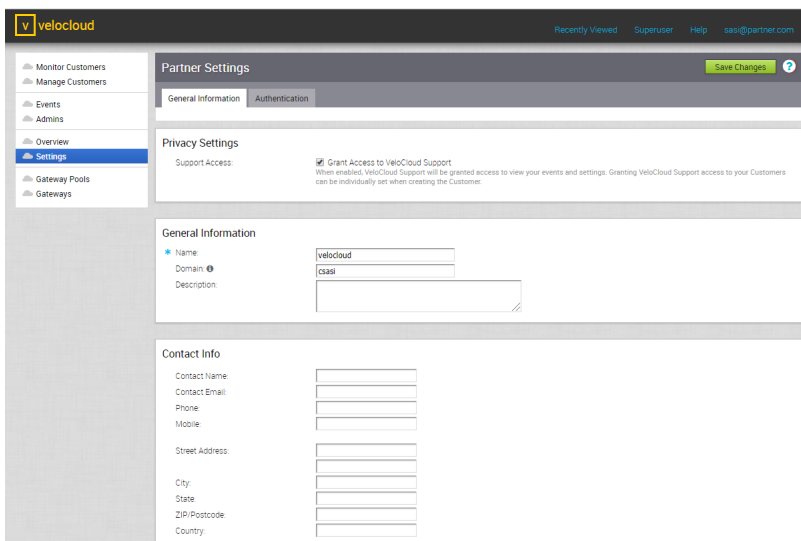
Prerequisites

- Ensure you have the Partner super user permission.
- Before setting up the SSO authentication in VeloCloud Orchestrator (VCO), ensure you have set up roles, users, and OpenID connect (OIDC) application for VCO in your preferred identity provider’s website. For more information, see [Configure an IDP for Single Sign On](#).

Procedure

- 1 Log in to the VCO application as Partner super user, with your login credentials.
- 2 Click **Settings**.

The **Partner Settings** screen appears.



- Click the **General Information** tab and in the **Domain** text box, enter the domain name for your partner, if it is not already set.

Note To enable SSO authentication for the VCO, you must set up the domain name for your partner.

- Click the **Authentication** tab and from the **Authentication Mode** drop-down menu, select **SSO**.

The screenshot shows the 'Partner Settings' page in the VeloCloud interface. The 'Authentication' tab is active. The 'Partner Authentication' section includes the following fields and options:

- Authentication Mode:** SSO (selected)
- Identity Provider template:** (dropdown menu)
- OIDC well-known config URL:** (text box)
- Issuer:** (text box)
- Authorization Endpoint:** (text box)
- Token Endpoint:** (text box)
- User Information Endpoint:** (text box)
- Client Id:** (text box)
- Client Secret:** (text box with toggle for visibility)
- Scopes:** openid,profile,email,offline_access
- Role Attribute:** groups
- Role Map:**
 - MSP Superuser: superuser
 - MSP Standard Admin: standard
 - MSP Support: support
 - MSP Business: business

At the bottom, there is a note: "Remember to set https://10.108.181.116/login/soologin/openidCallback as an allowed redirect URL with your IDP application/client"

- From the **Identity Provider template** drop-down menu, select your preferred Identity Provider (IDP) that you have configured for Single Sign On.

Note When you select VMwareCSP as your preferred IDP, ensure to provide your Organization ID in the following format: `/csp/gateway/am/api/orgs/<full organization ID>`.

When you sign in to [VMware CSP console](#), you can view the organization ID you are logged into by clicking on your username. A shortened version of the ID is displayed under the organization name. Click the ID to display the full organization ID.

You can also manually configure your own IDPs by selecting **Others** from the **Identity Provider template** drop-down menu.

- In the **OIDC well-known config URL** text box, enter the OpenID Connect (OIDC) configuration URL for your IDP. For example, the URL format for Okta will be: `https://{oauth-provider-url}/.well-known/openid-configuration`.
- The VCO application auto-populates endpoint details such as Issuer, Authorization Endpoint, Token Endpoint, and User Information Endpoint for your IDP.
- In the **Client Id** text box, enter the client identifier provided by your IDP.
- In the **Client Secret** text box, enter the client secret code provided by your IDP, that is used by the client to exchange an authorization code for a token.

10 To determine user's role in VCO, select one of the options:

- **Use Default Role** – Uses the role set up in the VCO, by default. The supported roles are: MSP Superuser, MSP Standard Admin, MSP Support, and MSP Business.
- **Use Identity Provider Roles** – Uses the roles set up in the IDP.

11 On selecting the **Use Identity Provider Roles** option, in the **Role Attribute** text box, enter the name of the attribute set in the IDP to return roles.

12 In the **Role Map** area, map the IDP-provided roles to each of the VCO roles, separated by using commas.

Roles in VMware CSP will follow this format: *external/<service definition uuid>/<service role name mentioned during service template creation>*.

13 Update the allowed redirect URLs in OIDC provider website with VCO URL (*https://<vco>/login/ssologin/openidCallback*).

14 Click **Save Changes** to save the SSO configuration.

15 Click **Test Configuration** to validate the entered OpenID Connect (OIDC) configuration.

The user is navigated to the IDP website and allowed to enter the credentials. On IDP verification and successful redirect to VCO test call back, a successful validation message will be displayed.

Results

The SSO authentication setup is complete in VCO.

What to do next

[Chapter 3 Log in to VCO Using SSO for Partner User](#)

Manage Gateway Pools

10

Partners can create and manage Gateway Pools and Gateways if their Operator has enabled this functionality. Once enabled, partners can access this feature from the **Gateway Pools** and **Gateway** links, respectively.

If an Operator has granted a Partner access to create and manage Gateway Pools, the partner will see a check mark in the **Managed Pool** column associated with a Gateway Pool.

The screenshot shows the 'VeloAcme Partner' interface. The sidebar on the left contains navigation links: Monitor Partner Customers, Manage Partner Customers, Partner Events, Partner Admins, Partner Overview, Partner Settings, Gateway Pools (highlighted), and Gateways. The main content area is titled 'Gateway Pools' and features a world map with a green dot in Southeast Asia. Below the map is a search bar and a table with the following data:

Gateway Pool	Gateways	Customers	Partner Gateway	Managed Pool
<input type="checkbox"/> Perf-pool	0	0	None	x
<input type="checkbox"/> gwpool	1	1	None	<input checked="" type="checkbox"/>

Partners cannot modify operator-owned Gateway Pools. These Gateway Pools will have a “x” associated with them under the **Managed Pool** column, and the settings in the **Properties** and **Gateways In Pool** areas are read-only.

The screenshot shows the VeloCloud Partner interface. The top navigation bar includes the logo 'VeloAcme Partner', 'Recently Viewed', 'Superuser', 'Help', and 'user@partner1.com'. The left sidebar contains a menu with options: Monitor Customers, Manage Customers, Events, Admins, Overview, Settings, Gateway Pools (selected), and Gateways. The main content area is titled 'Gateway Pools - _gwpool'. It features a 'Properties' section with the following details: Name: gwpool, Description: none, and Partner Gateway Hand Off: None. Below this is a table titled 'Gateways In Pool' with columns for Gateway, Location, IP Address, Service State, and Status. The table contains one entry: Gateway 'Velo.gateway', Location '00, SG', IP Address '52.76.95.186', Service State 'In Service', and Status 'Offline'. At the bottom, there is a 'Customers' section with a table that is currently empty, displaying 'No Items'.

This chapter includes the following topics:

- [Create a Gateway Pool](#)

Create a Gateway Pool

You can create a new Gateway Pool if your Operator has granted you access to this feature. Contact your operator if you want to gain access. Gateways owned and created by your Operator are read-only.

To create a new Gateway Pool:

- 1 Click the **New Gateway Pool** button to create a new Gateway Pool.
- 2 In the **New Gateway Pool** dialog box:
 - a Enter a unique **Name** and a **Description** of the Gateway Pool.
 - b Choose an option from the **Partner Gateway Hand Off** drop-down menu.
- 3 Click the **Create** button to create your Gateway Pool.

You can modify any Gateway Pools that you own. However, Operator-owned Gateway Pools are read-only.

Note Partner-created Gateways will be visible only to that specific Partner and can only be used within the Partner Pools.

Manage Gateways

11

A partner can manage gateways from the **Gateways** link.

To manage Gateways:

- In the navigation bar, click **Gateways**.

The **Gateways** page appears.

Configure Single Sign On for Identity Partners

12

The Identity Partner (IDP) Configuration for Single Sign On (SSO) is newly added for the 3.3.1 release.

This chapter includes the following topics:

- [Configure an IDP for Single Sign On](#)

Configure an IDP for Single Sign On

To enable Single Sign On (SSO) for VeloCloud Orchestrator (VCO), you must configure an Identity Partner (IDP) with details of VCO. Currently, the following IDPs are supported: Okta, OneLogin, PingIdentity, AzureAD, and VMware CSP.

For step-by-step instructions to configure an OpenID Connect (OIDC) application for VCO in various IDPs, see:

- [Configure Okta for Single Sign On](#)
- [Configure OneLogin for Single Sign On](#)
- [Configure PingIdentity for Single Sign On](#)
- [Configure Azure Active Directory for Single Sign On](#)
- [Configure VMware CSP for Single Sign On](#)

Configure Okta for Single Sign On

To support OpenID Connect (OIDC)-based Single Sign On (SSO) from Okta, you must first set up an application in Okta. To set up an OIDC-based application in Okta for SSO, perform the steps on this procedure.

Prerequisites

Ensure you have an Okta account to sign in.

Procedure

- 1 Log in to your **Okta** account as an Admin user.

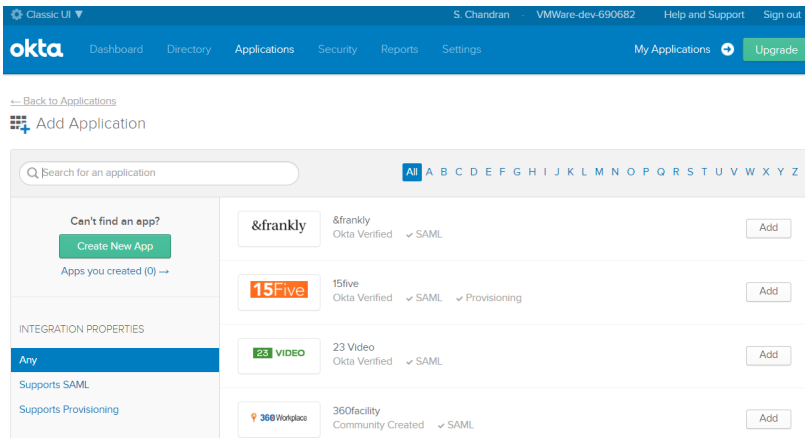
The **Okta** home screen appears.

Note If you are in the Developer Console view, then you must switch to the Classic UI view by selecting **Classic UI** from the **Developer Console** drop-down list.

- 2 To create a new application:

- a In the upper navigation bar, click **Applications > Add Application**.

The **Add Application** screen appears.



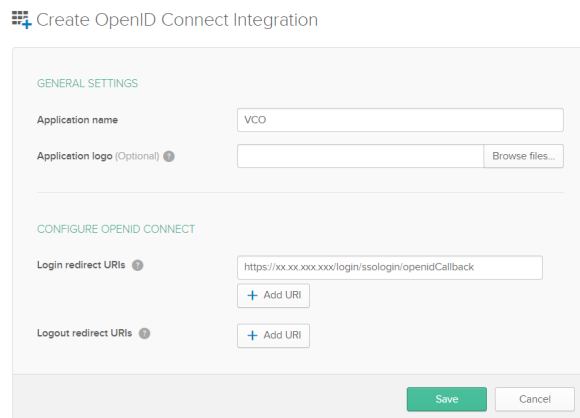
- b Click **Create New App**.

The **Create a New Application Integration** dialog box appears.

- c From the **Platform** drop-drop menu, select **Web**.

- d Select **OpenID Connect** as the Sign on method and click **Create**.

The **Create OpenID Connect Integration** screen appears.

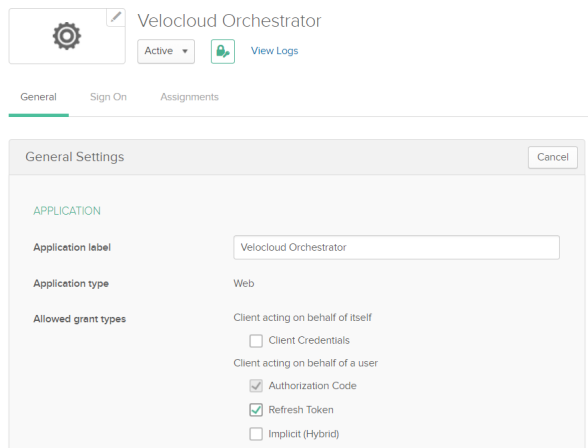


- e Under the **General Settings** area, in the **Application name** text box, enter the name for your application (for example, VCO).
- f Under the **CONFIGURE OPENID CONNECT** area, in the **Login redirect URIs** text box, enter the redirect URL that your VCO application uses as the callback endpoint.

In the VCO application, at the bottom of the **Configure Authentication** screen, you can find the redirect URL link. Ideally, the VCO redirect URL will be in this format: `https://<VCO URL>/login/ssologin/openidCallback`.

- g Click **Save**.
- h On the **General** tab, click **Edit** and select **Refresh Token** for Allowed grant types, and click **Save**.

Note down the Client Credentials (Client ID and Client Secret) to be used during the SSO configuration in VCO.



- i Click the **Sign On** tab and under the **OpenID Connect ID Token** area, click **Edit**.
- j In the **Groups claim filter** area, set the filter for the user groups and click **Save**.

The application is setup in IDP. You can assign groups and users to your VCO application.

3 To assign groups and users to your VCO application:

- a Go to **Application > Applications** and click on your VCO application link.
- b On the **Assignments** tab, from the **Assign** drop-down menu, select **Assign to Groups** or **Assign to People**.

The **Assign <Application Name> to Groups** or **Assign <Application Name> to People** dialog box appears.

- c Click **Assign** next to available user groups or users you want to assign the VCO application and click **Done**.

Results

You have completed setting up an OIDC-based application in Okta for SSO.

What to do next

Configure Single Sign On in VCO.

Create a New User Group in Okta

To create a new user group, perform the steps on this procedure.

Procedure

- 1 Click **Directory** > **Groups**.
- 2 Click **Add Group**.
The **Add Group** dialog box appears.
- 3 Enter the group name and description for the group and click **Save**.

Create a New User in Okta

To add a new user, perform the steps on this procedure.

Procedure

- 1 Click **Directory** > **People**.
- 2 Click **Add Person**.
The **Add Person** dialog box appears.
- 3 Enter all the mandatory details such as first name, last name, and email ID of the user.
- 4 If you want to set the password, select **Set by user** from the **Password** drop-down menu and enable **Send user activation email now**.
- 5 Click **Save**.
An activation link email will be sent your email ID. Click the link in the email to activate your Okta user account.

Configure OneLogin for Single Sign On

To set up an OpenID Connect (OIDC)-based application in OneLogin for Single Sign On (SSO), perform the steps on this procedure.

Prerequisites

Ensure you have an OneLogin account to sign in.

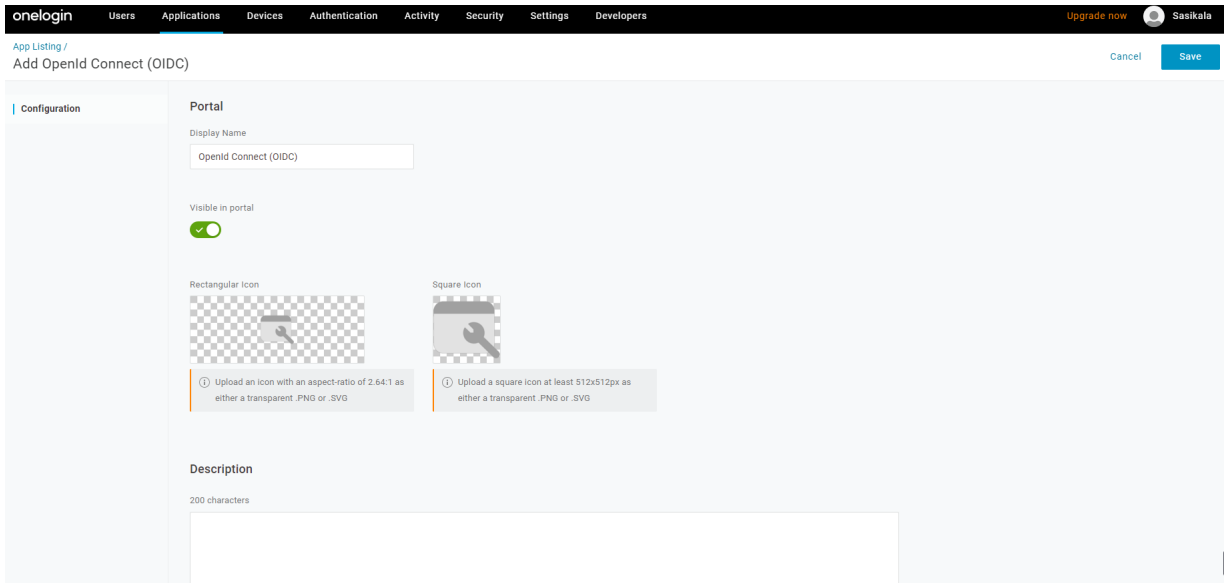
Procedure

- 1 Log in to your [OneLogin](#) account as an Admin user.
The **OneLogin** home screen appears.

2 To create a new application:

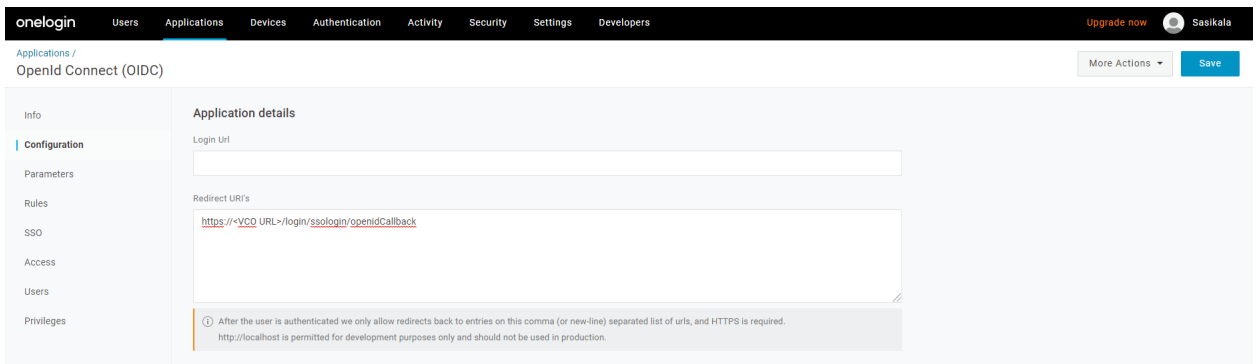
- a In the upper navigation bar, click **Apps > Add Apps**.
- b In the **Find Applications** text box, search for “OpenId Connect” or “oidc” and then select the **OpenId Connect (OIDC)** app.

The **Add OpenId Connect (OIDC)** screen appears.



- c In the **Display Name** text box, enter the name for your application (for example, VCO) and click **Save**.
- d On the **Configuration** tab, enter the redirect URI that VCO uses as the callback endpoint and click **Save**.

In the VCO application, at the bottom of the **Authentication** screen, you can find the redirect URL link. Ideally, the VCO redirect URL will be in this format: `https://<VCO URL>/login/ssologin/openidCallback`.



- e On the **Parameters** tab, under **OpenId Connect (OIDC)**, double click **Groups**.

The **Edit Field Groups** popup appears.

Edit Field Groups

Name
Groups

Value
Select Groups Add

Added Items

Default if no value selected
User Roles
--No transform-- (Single value output)

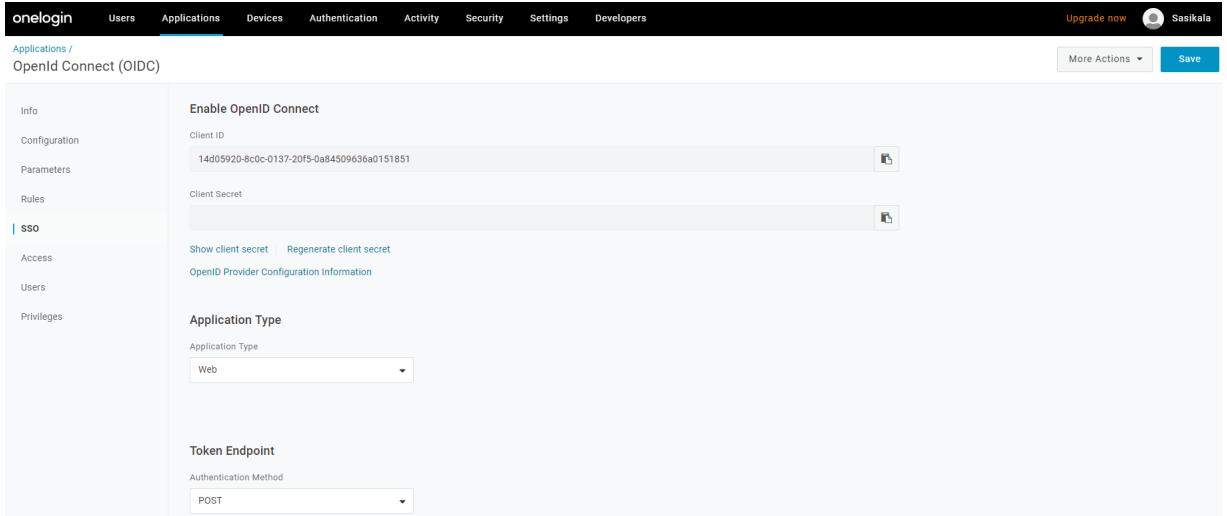
ⓘ This value will be used if no value has been selected in the table above

Cancel Save

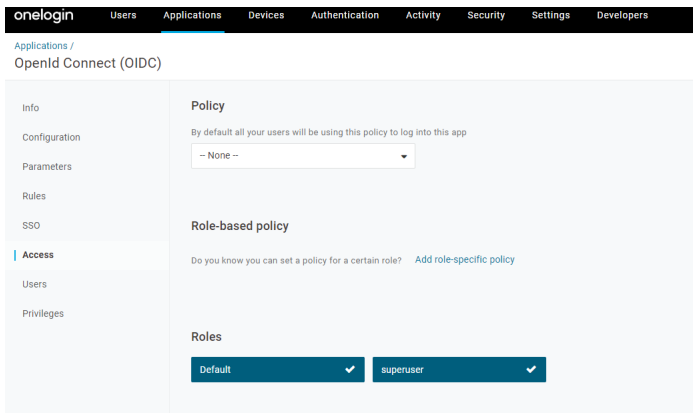
- f Configure User Roles with value “--No transform--(Single value output)” to be sent in groups attribute and click **Save**.
- g On the **SSO** tab, from the **Application Type** drop-down menu, select **Web**.

- h From the **Authentication Method** drop-down menu, select **POST** as the Token Endpoint and click **Save**.

Also, note down the Client Credentials (Client ID and Client Secret) to be used during the SSO configuration in VCO.



- i On the **Access** tab, choose the roles that will be allowed to login and click **Save**.



- 3 To add roles and users to your VCO application:
 - a Click **Users > Users** and select a user.
 - b On the **Application** tab, from the **Roles** drop-down menu, on the left, select a role to be mapped to the user.
 - c Click **Save Users**.

Results

You have completed setting up an OIDC-based application in OneLogin for SSO.

What to do next

Configure Single Sign On in VCO.

Create a New Role in OneLogin

To create a new role, perform the steps on this procedure.

Procedure

1 Click **Users > Roles**.

2 Click **New Role**.

3 Enter a name for the role.

When you first set up a role, the **Applications** tab displays all the apps in your company catalog.

4 Click an application to select it and click **Save** to add the selected apps to the role.

Create a New User in OneLogin

To create a new user, perform the steps on this procedure.

Procedure

1 Click **Users > Users > New User**.

The **New User** screen appears

2 Enter all the mandatory details such as first name, last name, and email ID of the user and click **Save User**.

Configure PingIdentity for Single Sign On

To set up an OpenID Connect (OIDC)-based application in PingIdentity for Single Sign On (SSO), perform the steps on this procedure.

Prerequisites

Ensure you have a PingOne account to sign in.

Note Currently, VeloCloud Orchestrator (VCO) supports PingOne as the Identity Partner (IDP); however, any PingIdentity product supporting OIDC can be easily configured.

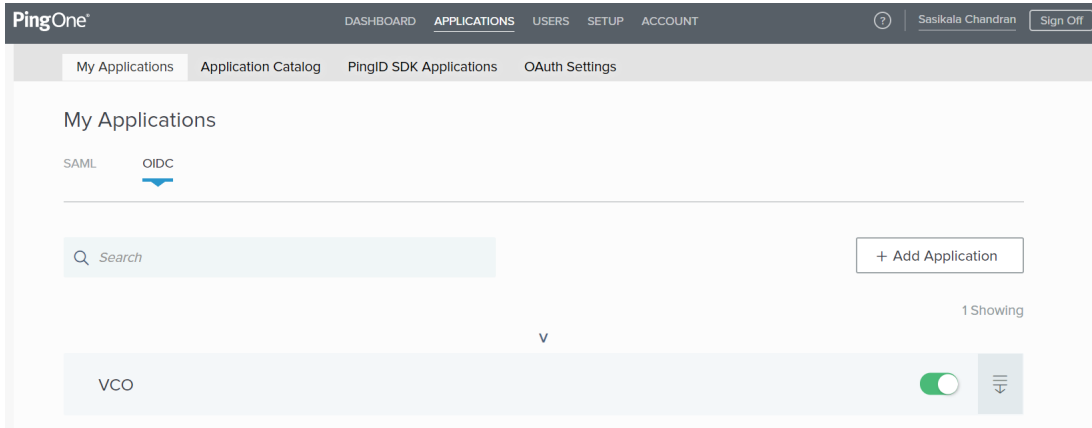
Procedure

1 Log in to your [PingOne](#) account as an Admin user.

The **PingOne** home screen appears.

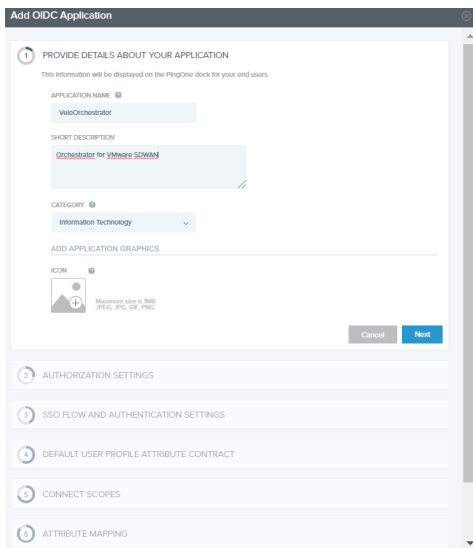
2 To create a new application:

- a In the upper navigation bar, click **Applications**.



- b On the **My Applications** tab, select **OIDC** and then click **Add Application**.

The **Add OIDC Application** pop-up window appears.



- c Provide basic details such as name, short description, and category for the application and click **Next**.

- d Under **AUTHORIZATION SETTINGS**, select **Authorization Code** as the allowed grant types and click **Next**.

Also, note down the Discovery URL and Client Credentials (Client ID and Client Secret) to be used during the SSO configuration in VCO.

- e Under **SSO FLOW AND AUTHENTICATION SETTINGS**, provide valid values for Start SSO URL and Redirect URL and click **Next**.

In the VCO application, at the bottom of the **Configure Authentication** screen, you can find the redirect URL link. Ideally, the VCO redirect URL will be in this format: `https://<VCO URL>/login/ssologin/openidCallback`. The Start SSO URL will be in this format: `https://<vco>/<domain name>/login/doEnterpriseSsoLogin`.

- f Under **DEFAULT USER PROFILE ATTRIBUTE CONTRACT**, click **Add Attribute** to add additional user profile attributes.
- g In the **Attribute Name** text box, enter `group_membership` and then select the **Required** checkbox, and select **Next**.

Note The `group_membership` attribute is required to retrieve roles from PingOne.

- h Under **CONNECT SCOPES**, select the scopes that can be requested for your VCO application during authentication and click **Next**.
- i Under **Attribute Mapping**, map your identity repository attributes to the claims available to your VCO application.

Note The minimum required mappings for the integration to work are email, given_name, family_name, phone_number, sub, and group_membership (mapped to memberOf).

- j Under **Group Access**, select all user groups that should have access to your VCO application and click **Done**.

The application will be added to your account and will be available in the **My Application** screen.

Results

You have completed setting up an OIDC-based application in PingOne for SSO.

What to do next

Configure Single Sign On in VCO.

Create a New User Group in PingIdentity

To create a new user group, perform the steps on this procedure.

Procedure

- 1 Click **Users > User Directory**.
- 2 On the **Groups** tab, click **Add Group**
The **New Group** screen appears.
- 3 In the **Name** text box, enter a name for the group and click **Save**.

Create a New User in PingIdentity

To add a new user, perform the steps on this procedure.

Procedure

- 1 Click **Users > User Directory**.
- 2 On the **Users** tab, click the **Add Users** drop-down menu and select **Create New User**.
The **User** screen appears.
- 3 Enter all the mandatory details such as username, password, and email ID of the user.
- 4 Under **Group Memberships**, click **Add**.
The **Add Group Membership** pop-up window appears.
- 5 Search and add the user to a group and click **Save**.

Configure Azure Active Directory for Single Sign On

To set up an OpenID Connect (OIDC)-based application in Microsoft Azure Active Directory (AzureAD) for Single Sign On (SSO), perform the steps on this procedure.

Prerequisites

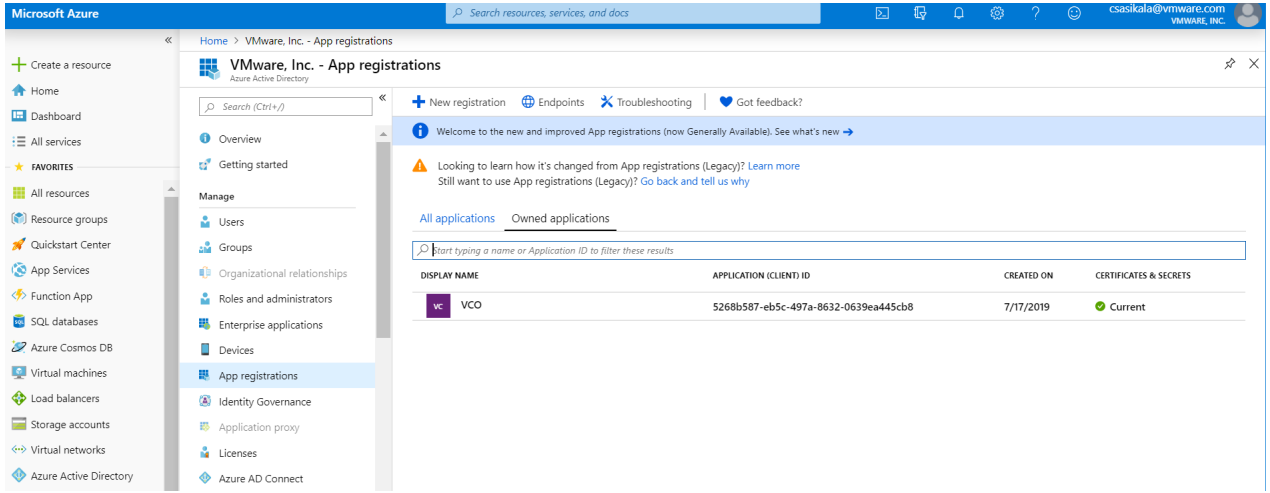
Ensure you have an AzureAD account to sign in.

Procedure

- 1 Log in to your [Microsoft Azure](#) account as an Admin user.
The **Microsoft Azure** home screen appears.

2 To create a new application:

- a Search and select the **Azure Active Directory** service.



- b Go to **App registration > New registration**.

The **Register an application** screen appears.

Register an application

* Name
The user-facing display name for this application (this can be changed later).

Supported account types
Who can use this application or access this API?
 Accounts in this organizational directory only (VeloCloud Networks, incit@velo)
 Accounts in any organizational directory
 Accounts in any organizational directory and personal Microsoft accounts (e.g. Skype, Xbox, Outlook.com)
[Help me choose...](#)

Redirect URI (optional)
We'll return the authentication response to this URI after successfully authenticating the user. Providing this now is optional and it can be changed later, but a value is required for most authentication scenarios.

By proceeding, you agree to the [Microsoft Platform Policies](#)

Register

- c In the **Name** field, enter the name for your VeloCloud Orchestrator (VCO) application.
- d In the **Redirect URL** field, enter the redirect URL that your VCO application uses as the callback endpoint.

In the VCO application, at the bottom of the **Configure Authentication** screen, you can find the redirect URL link. Ideally, the VCO redirect URL will be in this format: `https://<VCO URL>/login/ssologin/openidCallback`.

- e Click **Register**.

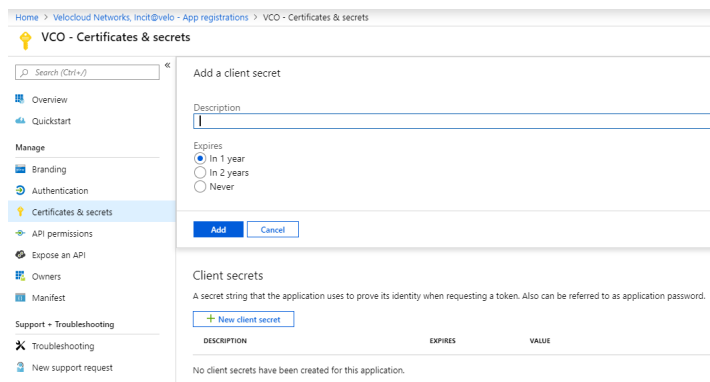
Your VCO application will be registered and displayed in the **All applications** and **Owned applications** tabs. Make sure to note down the Client ID/Application ID to be used during the SSO configuration in VCO.

- f Click **Endpoints** and copy the well-known OIDC configuration URL to be used during the SSO configuration in VCO.

- g To create a client secret for your VCO application, on the **Owned applications** tab, click on your VCO application.

- h Go to **Certificates & secrets > New client secret**.

The **Add a client secret** screen appears.

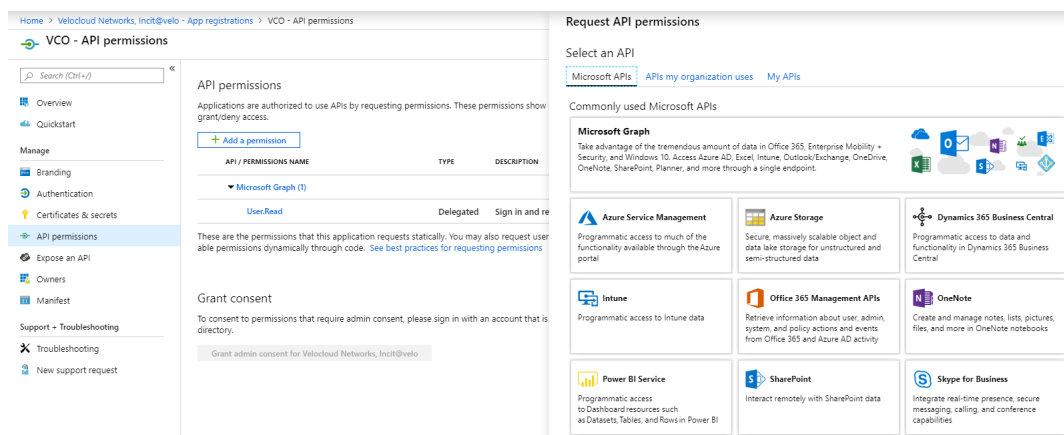


- i Provide details such as description and expiry value for the secret and click **Add**.

The client secret will be created for the application. Note down the new client secret value to be used during the SSO configuration in VCO.

- j To configure permissions for your VCO application, click on your VCO application and go to **API permissions > Add a permission**.

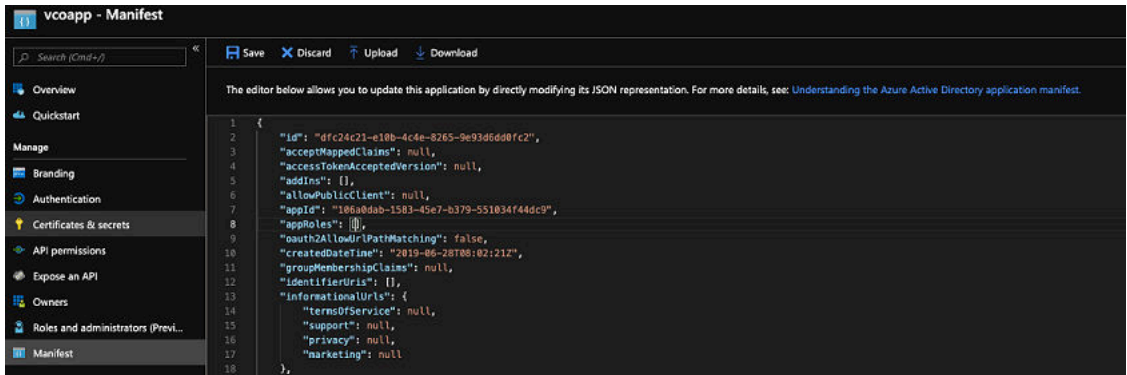
The **Request API permissions** screen appears.



- k Click **Microsoft Graph** and select **Application permissions** as the type of permission for your application.
- l Under **Select permissions**, from the **Directory** drop-down menu, select **Directory.Read.All** and from the **User** drop-down menu, select **User.Read.All**.
- m Click **Add permissions**.

- n To add and save roles in the manifest, click on your VCO application and from the application **Overview** screen, click **Manifest**.

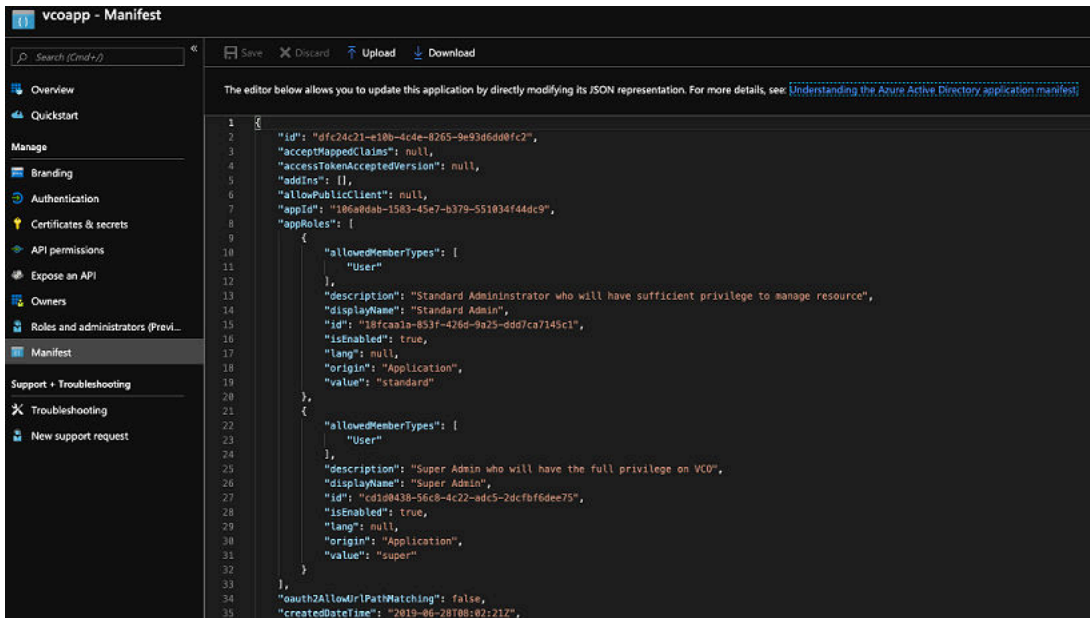
A web-based manifest editor opens, allowing you to edit the manifest within the portal. Optionally, you can select **Download** to edit the manifest locally, and then use **Upload** to reapply it to your application.



- o In the manifest, search for the `appRoles` array and add one or more role objects as shown in the following example and click **Save**.

Sample role objects

```
{
  "allowedMemberTypes": [
    "User"
  ],
  "description": "Standard Adminstrator who will have sufficient privilege to
manage resource",
  "displayName": "Standard Admin",
  "id": "18fcaa1a-853f-426d-9a25-ddd7ca7145c1",
  "isEnabled": true,
  "lang": null,
  "origin": "Application",
  "value": "standard"
},
{
  "allowedMemberTypes": [
    "User"
  ],
  "description": "Super Admin who will have the full privilege on VCO",
  "displayName": "Super Admin",
  "id": "cd1d0438-56c8-4c22-adc5-2dcfbf6dee75",
  "isEnabled": true,
  "lang": null,
  "origin": "Application",
  "value": "superuser"
}
```



Note Make sure to set id to a newly generated GUID value.

- 3 To assign groups and users to your VCO application:
 - a Go to **Azure Active Directory > Enterprise applications**.
 - b Search and select your VCO application.
 - c Click **Users and groups** and assign users and groups to the application.
 - d Click **Submit**.

Results

You have completed setting up an OIDC-based application in AzureAD for SSO.

What to do next

Configure Single Sign On in VCO.

Create a New Guest User in AzureAD

To create a new guest user, perform the steps on this procedure.

Procedure

- 1 Go to **Azure Active Directory > Users > All users**.
- 2 Click **New guest user**.
The **New Guest User** pop-up window appears.
- 3 In the **Email address** text box, enter the email address of the guest user and click **Invite**.
The guest user immediately receives a customizable invitation that lets them to sign into their Access Panel.

- 4 Guest users in the directory can be assigned to apps or groups.

Configure VMware CSP for Single Sign On

To configure VMware Cloud Services Platform (CSP) for Single Sign On (SSO), perform the steps on this procedure.

Prerequisites

Sign in to [VMware CSP console](#) (staging or production environment) with your VMware account ID. If you are new to VMware Cloud and do not have a VMware account, you can create one as you sign up. For more information, see How do I Sign up for VMware CSP section in [Using VMware Cloud](#) documentation.

Procedure

- 1 Contact the VMware SD-WAN Support Provider for receiving a Service invitation URL link to register your VCO application to VMware CSP. For information on how to contact the Support Provider, see <https://kb.vmware.com/s/article/53907> and https://www.vmware.com/support/contacts/us_support.html.

Your Support Provider will create and share:

- a Service invitation URL that needs to be redeemed to your Customer organization
- a Service definition uuid and Service role name to be used for Role mapping in Orchestrator

- 2 Redeem the Service invitation URL to your existing Customer Organization or create a new Customer Organization by following the steps in the UI screen.

You need to be a Organization Owner to redeem the Service invitation URL to your existing Customer Organization.

- 3 After redeeming the Service invitation, when you sign in to [VMware CSP console](#), you can view your application tile under **My Services** area in the **VMware Cloud Services** page.

The Organization you are logged into is displayed under your username on the menu bar. Make a note of the Organization ID by clicking on your username, to be used during Orchestrator configuration. A shortened version of the ID is displayed under the Organization name. Click the ID to display the full Organization ID.

- 4 Log in to [VMware CSP console](#) and create an OAuth application. For steps, see [Use OAuth 2.0 for Web Apps](#). Make sure to set Redirect URI to the URL displayed in **Configure Authentication** screen in VCO.

Once OAuth application is created in VMware CSP console, make a note of IDP integration details such as Client ID and Client Secret. These details will be needed for SSO configuration in Orchestrator.

- 5 Log in to your VCO application as Super Admin user and configure SSO using the received IDP integration details as follows.

- a Click **Administration > System Settings**

The **System Settings** screen appears.

- b Click the **General Information** tab and in the **Domain** text box, enter the domain name for your enterprise, if it is not already set.

Note To enable SSO authentication for the VCO, you must set up the domain name for your enterprise.

- c Click the **Authentication** tab and from the **Authentication Mode** drop-down menu, select **SSO**.

- d From the **Identity Provider template** drop-down menu, select **VMwareCSP**.

- e In the **Organization Id** text box, enter the Organization ID (that you have noted down in Step 3) in the following format: `/csp/gateway/am/api/orgs/<full organization ID>`

- f In the **OIDC well-known config URL** text box, enter the OpenID Connect (OIDC) configuration URL (<https://console.cloud.vmware.com/csp/gateway/am/api/.well-known/openid-configuration>) for your IDP.

The VCO application auto-populates endpoint details such as Issuer, Authorization Endpoint, Token Endpoint, and User Information Endpoint for your IDP.

- g In the **Client Id** text box, enter the client ID that you have noted down from the OAuth application creation step.
- h In the **Client Secret** text box, enter the client secret code that you have noted down from the OAuth application creation step.
- i To determine user's role in VCO, select either **Use Default Role** or **Use Identity Provider Roles**.
- j On selecting the **Use Identity Provider Roles** option, in the **Role Attribute** text box, enter the name of the attribute set in the VMware CSP to return roles.
- k In the **Role Map** area, map the VMwareCSP-provided roles to each of the VCO roles, separated by using commas.

Roles in VMware CSP will follow this format: `external/<service definition uuid>/<service role name mentioned during service template creation>`. Use the same Service definition uuid and Service role name that you have received from your Support Provider.

- 6 Click **Save Changes** to save the SSO configuration.

7 Click **Test Configuration** to validate the entered OpenID Connect (OIDC) configuration.

Configure Authentication Save Changes ?

Operator Authentication

Authentication Mode:

Identity Provider template:

Organization Id:

OIDC well-known config URL:

Issuer:

Authorization Endpoint:

Token Endpoint:

User Information Endpoint:

Client Id:

Client Secret:

Scopes:

Use Default Role Use Identity Provider Roles

Role Attribute:

Role Map

Operator Superuser:

Operator Standard Admin:

Operator Support:

Operator Business:

Remember to set <https://13.52.173.235/login/ssologin/openidCallback> as an allowed redirect URL with your IDP application/client

The user is navigated to the VMware CSP website and allowed to enter the credentials. On IDP verification and successful redirect to VCO test call back, a successful validation message will be displayed.

Results

You have completed integrating VCO application in VMware CSP for SSO and can access the VCO application logging in to the VMware CSP console.

What to do next

- Within the organization, manage users by adding new users and assigning appropriate role for the users. For more information, see [Manage Users](#).

Configure Edge Licensing

13

Partner Superusers, Partner Standard Administrators, Partner Business Specialist, and Partner Customer Support users can assign and manage license types to Enterprise customers.

This chapter includes the following topics:

- [Edge Licenses and License Types](#)
- [Generate an Edge Licensing Report](#)

Edge Licenses and License Types

This section provides an overview of Edge Licenses.

Enabling the Edge License Feature

Operators enable the Edge License feature for partners. If the Edge License feature is not enabled, contact your Operator.

Edge License Types

The Edge license type consists of the following attributes:

Attribute	Description
Bandwidth	10M, 30M, 50M, 100M, 200M, 500M, 1G, 2G, 5G, 10G
Editions (from lowest level to highest level)	Standard, Enterprise, Premium
Region	North America Europe Middle East, LATAM, APJC
Term	1 Year, 3 Years, 5 Years

Edge License Type Catalog

Partner Superusers, Partner Standard Administrators, Partner Business Specialist, and Partner Customer Support can assign license types to Enterprise customers from a catalog of 270 license types. These users will get access to the catalog of 270 license types automatically during a VCO installation or when upgrading to the 3.3 release. These license types will be displayed in the **Edge Licensing** screen. To use the Edge License feature, the above-mentioned users must enable the Edge License system property.

Considerations for Assigning Edge License Types

Note Assigning a license type to an Edge does NOT change or limit the functionality of the Edge in anyway. The Edge License feature does NOT enforce license types on to the Edge, but merely introduces the ability to attach license types. The intent is to ensure license types can be attached to Edges and can be reported when necessary.

When assigning Edge License Types, consider the following issues:

Issue	Considerations
Mixing License Types	<ul style="list-style-type: none"> Standard License Type: No mixing of license types Enterprise License Type: Can mix with the Premium license type Premium License Type: Can mix with the Enterprise license type
Upgrading an Edge license Types	<ul style="list-style-type: none"> A Standard license type can be upgraded to an Enterprise or Premium license type An Enterprise license type can be upgraded to a Premium license type
Downgrading License Types	<ul style="list-style-type: none"> License Types cannot be downgraded. Once a higher edition license type is assigned, it cannot be downgraded to a lower edition.

Generate an Edge Licensing Report

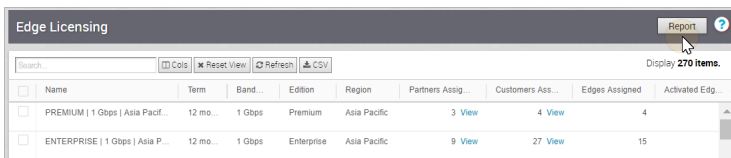
Partner Superusers, Partner Standard Administrators, Partner Business Specialist, and Partner Customer Support users can generate a report listing both the number of license types that are assigned to Edges and Edges that do not have any license types assigned to them.

To generate a report:

Note Partners on-boarding new customers will be able to assign a license type to every customer.

To generate an Edge Licensing Report:

- 1 From the VCO navigation panel, go to **Administration > Edge Licensing**.
- 2 From the **Edge Licensing** screen, click the **Report** button.



The Excel spreadsheet report automatically downloads.

Install the VeloCloud Partner Gateway

14

This document describes the steps needed to install and deploy VeloCloud Gateway (VCG) as a Partner Gateway. It also covers how to configure the VRF/VLAN and BGP configuration necessary on the VeloCloud Orchestrator (VCO).

This chapter includes the following topics:

- [Installation Overview](#)
- [Hypervisor Minimum Hardware Requirements](#)
- [VeloCloud Gateway Installation Procedures](#)
- [Post-Installation Tasks](#)
- [Upgrade VeloCloud Gateway](#)
- [Custom Configurations](#)
- [SNMP Integration](#)
- [Custom Firewall Rules](#)

Installation Overview

This section provides an overview of VeloCloud Partner Gateway installation.

About Partner Gateways

Partner Gateways are Gateways tailored to an on-premise operation in which the Gateway is installed and deployed with two interfaces.

- One interface is facing the private and/or public WAN network and is dedicated to receiving VCMP encapsulated traffic from the remote edges, as well as standard IPsec traffic from non-VeloCloud sites.
- Another interface is facing the datacenter and provides access to resources or networks attached to a PE router, which the Partner Gateway is connected to. The PE router typically affords access to shared managed services that are extended to the branches, or access to a private (MPLS / IP-VPN) core network in which individual customers are separated.

What's Provided?

The following distributions are provided:

Provided	Description	Example
VMware	Gateway OVA package.	velocloud-vcg-2.4.0-R24-20170428-GA.ova
KVM	Gateway qcow2 disk image.	velocloud-vcg-2.4.0-R24-20170428-GA.qcow2

Hypervisor Minimum Hardware Requirements

The VeloCloud Gateway runs on a standard hypervisor (KVM or VMware ESXi).

Minimum Server Requirements

To run the hypervisor:

- 10 Intel CPU's at 2.0 Ghz or higher. The CPU must support the AES-NI, SSSE3, SSE4 and RDTSC instruction sets.
- 20+ GB (16 GB is required for VC Gateway VM memory)
- 100 GB magnetic or SSD based, persistent disk volume
- 2 x 1 Gbps (or higher) network interface. The physical NIC card should use the Intel 82599/82599ES chipset (for SR-IOV & DPDK support).

Reference Hardware Specifications:

Hardware	Specification
HP DL380G9	http://www.hp.com/hpinfo/newsroom/press_kits/2014/ComputeEra/HP_ProLiantDL380_DataSheet.pdf
NIC card with 82599/82599ES chipset	https://www.hpe.com/h20195/v2/GetPDF.aspx/c04111506.pdf

Supported Hypervisor Versions

Hypervisor	Supported Versions
VMware	ESXi 5.5U3 or later. In order to use SR-IOV, the vCenter and the vSphere Enterprise Plus license are required.
KVM	Ubuntu 14.04 LTS and 16.04 LTS

VCG Virtual Hardware Specification

For VMware, the OVA already specifies the minimum virtual hardware specification. For KVM, an example XML file will be provided. The minimum virtual hardware specifications are:

- 8 vCPUs

- 8 GB of memory
- Minimum of 2 vNICs:
 - One vNIC is the public (outside) interface, which must be an untagged interface.
 - One vNIC is the private (inside) interface that must be tagged. This is the interface facing the PE router or L3 switch.
- Optional vNIC (if a separate management/OAM interface is required)
- 32 GB of virtual disk

Firewall/NAT Requirements

Note These requirements apply if the VeloCloud Gateway is deployed behind a Firewall and/or NAT device.

- The firewall needs to allow outbound traffic from the VeloCloud Gateway to TCP/443 (for communication with VeloCloud Orchestrator).
- The firewall needs to allow inbound traffic from the Internet to UDP/2426 (VCMP), UDP/4500, and UDP/500. If NAT is not used, then the firewall needs to also allow IP/50 (ESP).
- If NAT is used, the above ports must be translated to an externally reachable IP address. Both the 1:1 NAT and port translations are supported.

Git Repository with Templates and Samples

The following Git repository contains templates and samples.

```
git clone https://bitbucket.org/velocloud/deployment.git
```

VeloCloud Gateway Installation Procedures

This section describes the VeloCloud Gateway installation procedures.

In general, installing the VCG involves the following steps:

- 1 Create VCG on VCO and make a note of the activation key.
- 2 Configure VCG on VCO.
- 3 Create the cloud-init file.
- 4 Create the VM in VMware or KVM.
- 5 Boot the VCG VM and ensure the VCG cloud-init initializes properly. At this stage, the VCG should already activate itself against the VCO.
- 6 Verify connectivity and disable cloud-init.

Important VCG supports both the virtual switch and SR-IOV. This guide specifies the SR-IOV as an optional configuration step.

Pre-Installation Considerations

The VeloCloud Partner Gateway provides different configuration options. The worksheet below should be prepared before the installation of the Gateway.

This section explains this worksheet.

Worksheet

VCG	<ul style="list-style-type: none"> ■ Version ■ OVA/QCOW2 file location ■ Activation Key ■ VCO (IP ADDRESS/vco-fqdn-hostname) ■ Hostname
Hypervisor address/cluster name	Address/Cluster name
Storage	Root volume datastore (>40GB recommended)
	Note It is recommended that on a Partner Gateway Host, the free disk space in the /tmp/partition directory is at least twice the size of memory (RAM).
CPU Allocation	CPU Allocation for KVM/VMware.
Installation Selections	DPDK—This is optional and enabled by default for higher throughput. If you choose to disable DPDK, contact VMware Customer Support .
OAM Network (Optional See Custom Configurations)	<ul style="list-style-type: none"> ■ DHCP ■ OAM IPv4 Address ■ OAM IPv4 Netmask ■ DNS server - primary ■ DNS server - secondary ■ Static Routes
ETH0 – Internet Facing Network	<ul style="list-style-type: none"> ■ IPv4 Address ■ IPv4 Netmask ■ IPv4 Default gateway ■ DNS server - primary ■ DNS server - secondary
Handoff (ETH1) - Network	<ul style="list-style-type: none"> ■ MGMT VRF IPv4 Address ■ MGMT VRF IPv4 Netmask ■ MGMT VRF IPv4 Default gateway ■ DNS server - primary ■ DNS server - secondary ■ Handoff (QinQ (0x8100), QinQ (0x9100), none, 802.1Q, 802.1ad) ■ C-TAG ■ S-TAG

Console access	<ul style="list-style-type: none"> ■ Console_Password ■ SSH: <ul style="list-style-type: none"> ■ Enabled (yes/no) ■ SSH public key
NTP (Optional see Custom Configuration Section)	<ul style="list-style-type: none"> ■ Public NTP: <ul style="list-style-type: none"> ■ server 0.ubuntu.pool.ntp.org ■ server 1.ubuntu.pool.ntp.org ■ server 2.ubuntu.pool.ntp.org ■ server 3.ubuntu.pool.ntp.org ■ Internal NTP server - 1 ■ Internal NTP server - 2

VCG Section

Most of the VCG section is self-explanatory.

- VCG
- Version - Should be same or lower than VCO
 - OVA/QCOW2 file location - Plan ahead the file location and disk allocation
 - Activation Key
 - VCO (IP ADDRESS/vco-fqdn-hostname)
 - Hostname - Valid Linux Hostname "RFC 1123"

Creating a Gateway and Getting the Activation Key

- 1 Go to **Operator > Gateway Pool** and create a new VeloCloud Gateway pool. For running VeloCloud Gateway in the Service Provider network, check the **Allow Partner Gateway** checkbox. This will enable the option to include the partner gateway in this gateway pool.

- 2 Go to **Operator > Gateway** and create a new gateway and assign it to the pool. The IP address of the gateway entered here must match the **public IP address** of the gateway. If unsure, you can run `curl ipinfo.io/ip` from the VCG which will return the public IP of the VCG.

Create New Gateway ✕

Properties

- * Gateway Name:
- * IP Address:
- Service State:
- Initial Gateway Pool:

Site Contact

- * Name:
- * Email:

- 3 Make a note of the activation key and add it to the worksheet.

- ▲ Monitor Customers
- ▲ Manage Customers
- ▲ System Software Packages
- ▲ System Properties

This Gateway has been provisioned with activation key **Y4RN-YWPX-49K8-543X**.

Configure Gateways -
My Gateway #1

Enable Partner Gateway Mode

- 1 Go to **Operator > Gateways** and select the VeloCloud Gateway. Check the **Partner Gateway** checkbox to enable the Partner Gateway.

Gateways - SP-VCG1 Save Changes

Properties

- * Name:
- Description:
- Service State:
- Status: **Never Activated**
- IP Address:
- Gateway Authentication Mode:

Gateway Roles

- Control Plane
- Data Plane
- Partner Gateway**
- Secure VPN Gateway

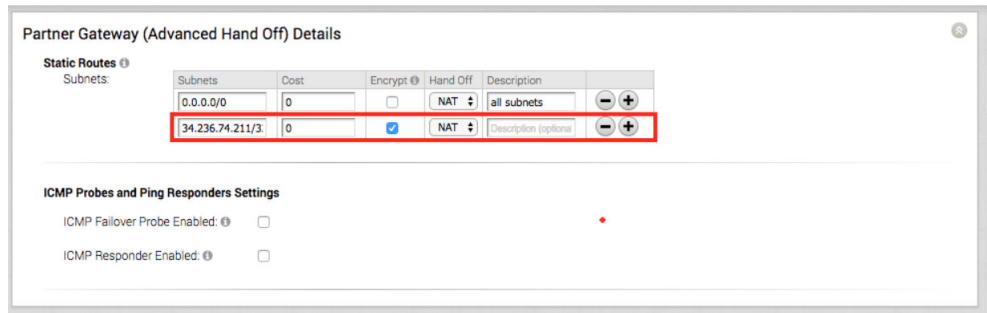
There are additional parameters that can be configured. The most common are the following:

Advertise 0.0.0.0/0 with no encrypt



This option will enable the Partner Gateway to advertise a path to Cloud traffic for the SAAS Application. Since the Encrypt Flag is off, it will be up to the customer configuration on the business policy to use this path or not.

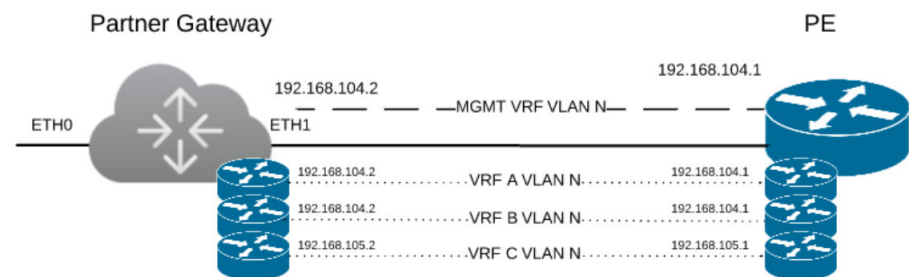
The second recommend option is to **advertise the VCO IP as a /32 with encrypt**



This will force the traffic that is sent from the Edge to the VCO to take the Gateway Path. This is recommended since it introduces predictability to the behavior that the VCE takes to reach the VCO.

Networking

Important The following procedure and screenshots focus on the most common deployment, which is the 2-ARM installation for the Gateway. The addition of an OAM network is considered in the section titled, [OAM Interface and Static Routes](#).



The diagram above is a representation of the VeloCloud Gateway in a 2-ARM deployment. In this example, we assume eth0 is the interface facing the public network (Internet) and eth1 is the interface facing the internal network (handoff or VRF interface).

Note A **Management VRF** is created on the VCG and is used to send a periodic ARP refresh to the default gateway IP to check that the handoff interface is physically up and speed up the failover time. It is recommended that a dedicated VRF is set up on the PE router for this purpose. Optionally, the same management VRF can also be used by the PE router to send an IP SLA probe to the VCG to check for VCG status (VCG has a stateful ICMP responder that will respond to ping only when its service is up). If a dedicated Management VRF is not set up, then you can use one of the customer VRFs as a Management VRF, although this is not recommended.

For the Internet Facing network, you only need the basic network configuration.

- ETH0 – Internet Facing Network
- IPv4_Address
 - IPv4_Netmask
 - IPv4_Default_gateway
 - DNS_server_primary
 - DNS_server_secondary
-

For the Handoff interface, you must know which type of handoff you want to configure and the Handoff configuration for the Management VRF.

- ETH1 – HANDOFF Network
- MGMT_IPv4_Address
 - MGMT_IPv4_Netmask
 - MGMT_IPv4_Default gateway
 - DNS_Server_Primary
 - DNS_Server_Secondary
 - Handoff (QinQ (0x8100), QinQ (0x9100), none, 802.1Q, 802.1ad)
 - C_TAG_FOR_MGMT_VRF
 - S_TAG_FOR_MGMT_VRF
-

Console Access

- Console access
- Console_Password
 - SSH:
 - Enabled (yes/no)
 - SSH public key
-

In order to access the Gateway, a console password and/or a SSH public key must be created.

Cloud-Init Creation

The configuration options for the gateway that we defined in the worksheet are used in the cloud-init configuration. The cloud-init config is composed of two main configuration files, the metadata file and the user-data file. The meta-data contains the network configuration for the Gateway, and the user-data contains the Gateway Software configuration. This file provides information that identifies the instance of the VeloCloud Gateway being installed.

Below are the templates for both Meta_data and User_data files.

Fill the templates with the information in the worksheet. All #_VARIABLE_# need to be replaced, also check any #ACTION#

Important The template assumes you are using static configuration for the interfaces. It also assumes that you are either using SR-IOV for all interfaces or none. See section titled, [OAM - SR-IOV with vmxnet3 or SR-IOV with VIRTIO](#) for this. The templates are also available in the git repository at: [git clone https://bitbucket.org/velocloud/deployment.git](https://bitbucket.org/velocloud/deployment.git) **It is recommended that you get the templates from repository instead of copying and pasting from this document.** <https://bitbucket.org/velocloud/deployment>

meta-data file:

```
instance-id: #_Hostname_#
local-hostname: #_Hostname_#
```

```
network-interfaces: |
  auto eth0
    iface eth0 inet static
      address #_IPv4_Address_#
      mac_address #_mac_Address_#
      netmask #_IPv4_Netmask_#
      gateway #_IPv4_Gateway_#
      dns-nameservers
        #_DNS_server_primary_#
        #_DNS_server_secondary_#
  auto eth1
    iface eth1 inet static
      metric '13'
      address #_MGMT_IPv4_Address_#
      mac_address #_MGMT_mac_Address_#
      netmask #_MGMT_IPv4_Netmask_#
      gateway #_MGMT_IPv4_Gateway_#
      dns-nameservers
        #_DNS_server_primary_#
        #_DNS_server_secondary_#
```

user-data file:

```
#cloud-config
hostname: #_Hostname_#
password: #_Console_Password_#
chpasswd:
  expire: false

ssh_authorized_keys:
  - #_SSH_public_Key_#
ssh_pwauth: true
```



```

velocloud:
  vcg:
    activation_code: #_Activation_Key_#
    vco: #_VCO_#

runcmd:
- "echo \[\]\\" > /opt/vc/etc/vc_blocked_subnets.json"
- "sed -iorig \"s/wan=\\\".*wan=\\\"eth0 eth1\\\"/\\" /etc/config/gatewayd-tunnel"
- /var/lib/cloud/scripts/per-boot/config_gateway
- "sleep 10"
- "/opt/vc/bin/vc_procmon restart"

write_files:
-
  content: |
    #!/usr/bin/python
    import json
    ### EDIT GATEWAYD ###
    with open("/etc/config/gatewayd", "r") as jsonFile:
        data = json.load(jsonFile)
        data["global"]["vcmp.interfaces"] = ["eth0"]
        data["global"]["wan"] = ["eth1"]
        # NOTE FOR HAND OFF IT CAN BE "QinQ (0x8100)" "QinQ (0x9100)" "none" "802.1Q" "802.1ad"
        data["vrf_vlan"]["tag_info"][0]["mode"] = "#_Handoff_"
        data["vrf_vlan"]["tag_info"][0]["interface"] = "eth1"
        data["vrf_vlan"]["tag_info"][0]["c_tag"] = "#_C_TAG_FOR_MGMT_VRF_#"
        data["vrf_vlan"]["tag_info"][0]["s_tag"] = "#_S_TAG_FOR_MGMT_VRF_"
    with open("/etc/config/gatewayd", "w") as jsonFile:
        jsonFile.write(json.dumps(data,sort_keys=True,indent=4, separators=(",", ":")))
    ### EDIT DPDK ###
    with open("/opt/vc/etc/dpdk.json", "r") as jsonFile:
        data = json.load(jsonFile)
        #SET 0 or 1 for enabled or DISABLED example data["dpdk_enabled"] = 0
        data["dpdk_enabled"] = #_DKDP_ENABLED_(1)_OR_DISABLED_(0)_#
    with open("/opt/vc/etc/dpdk.json", "w") as jsonFile:
        jsonFile.write(json.dumps(data,sort_keys=True,indent=4, separators=(",", ":")))
  path: /var/lib/cloud/scripts/per-boot/config_gateway
  permissions: "0755"

final_message: "==== Cloud-init completed ====="

power_state:
  condition: true

```

```
delay: "+1"
message: "Bye Bye"
mode: reboot
timeout: 30
```

Important

- VMware recommends to have a proper fully qualified domain name (FQDN) configured for all production Orchestrators so proper TLS certificates may be issued for them.
- If activation using the Orchestrator's IP address is the only option, use the following example which instructs the Edge to bypass TLS verification.

```
commands.getoutput("/opt/vc/bin/ activate.py -s myvco.example.com -i #_activation_key_#")
```

- This configuration is not recommended for production use and we highly encourage you to reactivate against the Orchestrator's hostname at the soonest possible.

Note Always validate user-data and metadata, using <http://www.yamllint.com/> for example. - The metadata should also be a valid network configuration under the network-interface section, this section will be the /etc/network/interfaces once the cloud-init completes. - Sometimes when working with the Windows/Mac copy paste feature, there is a danger of introducing Smart Quotes which can corrupt the files. **Run this to make sure you are smart quote free**

```
sed s/[\"'\"']/\"'/g /tmp/user-data > /tmp/user-data_new
```

Create 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 vcg01-cidata.iso, is created with the following command on a Linux system:

```
genisoimage -output vcg01-cidata.iso -volid cidata -joliet -rock user-data meta-data
```

If you are on a MAC OSX, use the command below instead:

```
mkisofs -output vcg01-cidata.iso -volid cidata -joliet -rock {user-data,meta-data}
```

This iso file which we will call #CLOUD_INIT_ISO_FILE# is going to be used in both OVA and VMware installations.

Install VeloCloud Gateway

This section describes how to install VeloCloud Gateway on VMware and KVM.

KVM provides multiple ways to provide networking to virtual machines. VMware SD-WAN recommends the following options:

- SR-IOV
- Linux Bridge

- OpenVSwitch Bridge

If you decide to use SR-IOV mode, enable SR-IOV on KVM and VMware. For steps, see:

- [Enable SR-IOV on KVM](#)
- [Enable SR-IOV on VMware \(Optional\)](#)

To install VeloCloud Gateway:

- On KVM, see [Install VeloCloud Gateway on KVM](#).
- On VMware, see [Install VeloCloud Gateway on VMware](#).

Enable SR-IOV on VMware (Optional)

This section describes how to enable SR-IOV on VMware. This step is optional.

Prerequisites

This requires a specific NIC card. As of today, only the following chipset is certified by VeloCloud to work with the VCG.

- Intel 82599/82599ES
- X550 (under experimenting as this requires the latest Intel ixgbev driver on the VCG VM and Malicious Driver Detection disabled on the ESXi host ixgbe driver)

Instructions to Enable SR-IOV

To enable SR-IOV on VMware:

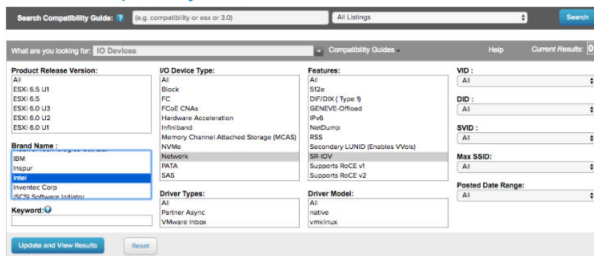
- 1 Make sure that your NIC card supports SR-IOV. Check the VMware Hardware Compatibility List (HCL) at <https://www.vmware.com/resources/compatibility/search.php?deviceCategory=io>

Brand Name: Intel

I/O Device Type: Network

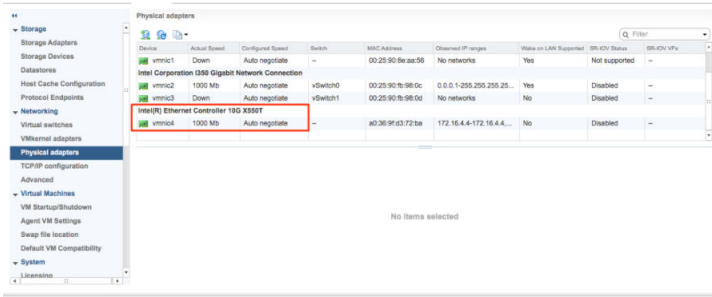
Features: SR-IOV

VMware Compatibility Guide



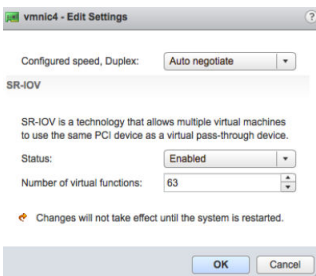
The following VMware KB article provides details of how to enable SR-IOV on the supported NIC: <https://kb.vmware.com/s/article/2038739>

- 2 Once you have a support NIC card, go to the specific VMware host, select the **Configure** tab, and then choose **Physical adapters**.

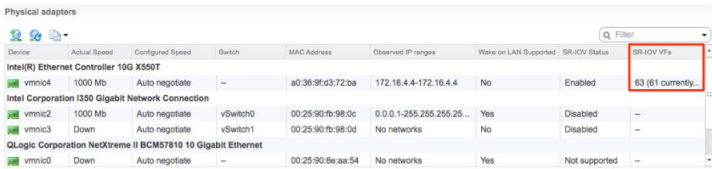


3 Select **Edit Settings**. Change **Status** to **Enabled** and specify the number of virtual functions required. This number varies by the type of NIC card.

4 Reboot the hypervisor.



5 If SR-IOV is successfully enabled, the number of Virtual Functions (VFs) will show under the particular NIC after ESXi reboots.



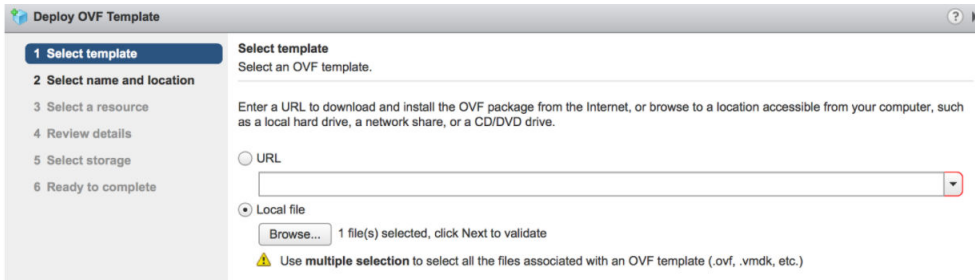
Install VeloCloud Gateway on VMware

This section describes how to install the VCG OVA on VMware.

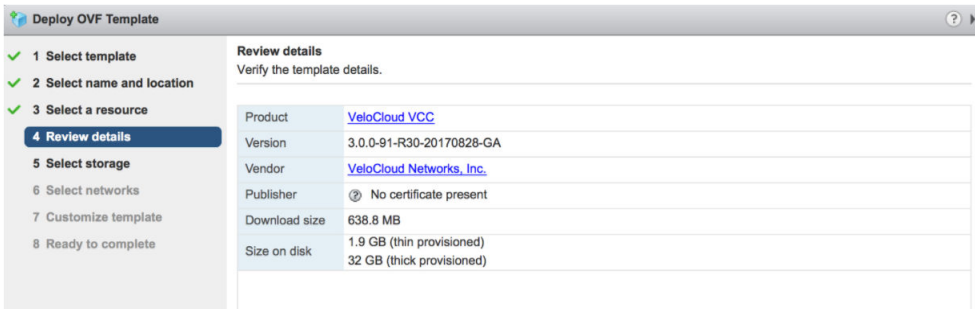
Important When you are done with the OVA installation, DO NOT start the VM until you have the cloud-init iso file and mount as CD-ROM to the VCG VM. Otherwise, you will need to re-deploy the VM again.

To install the VCG OVA on VMware:

1 Select the ESXi host, go to **Actions**, and then **Deploy OVF Template**. Select the VCG OVA file provided by VeloCloud and click **Next**.

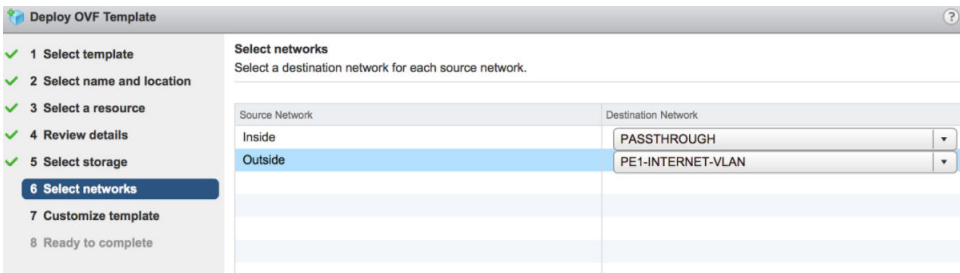


Review the template details in Step 4 (**Review details**) of the **Deploy OVA/OVF Template** wizard as shown in the image below.

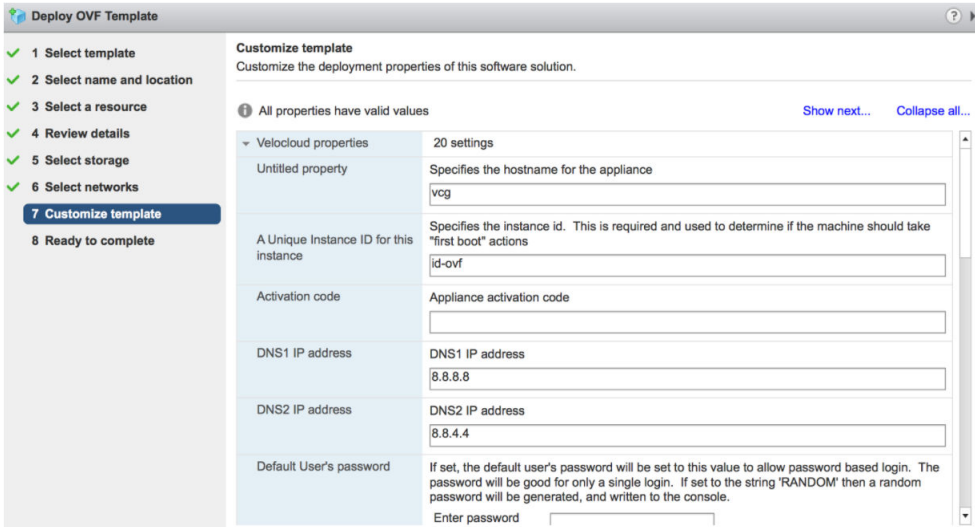


2 For the **Select networks** step, the OVA comes with two pre-defined networks (vNICs).

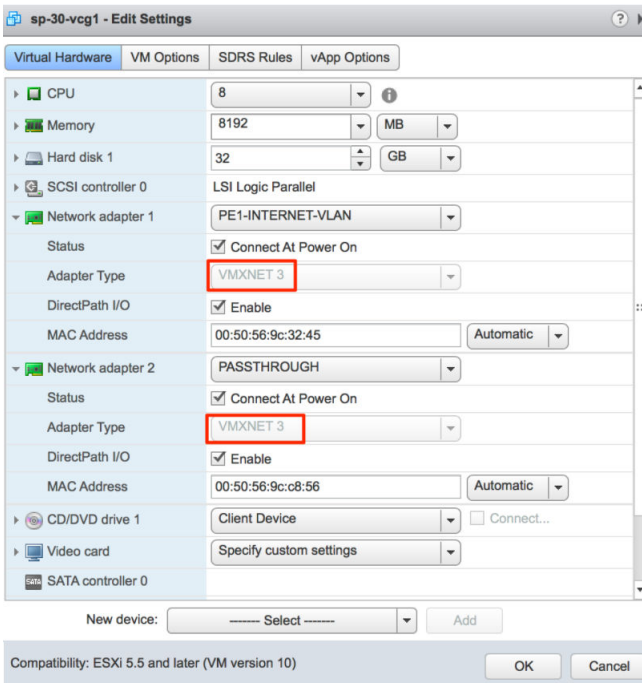
vNIC	Description
Inside	This is the vNIC facing the PE router and is used for handoff traffic to the MPLS PE or L3 switch. This vNIC is normally bound to a portgroup that does a VLAN pass-through (VLAN=4095 in vswitch configuration).
Outside	This is the vNIC facing the Internet. This vNIC expects a non-tagged L2 frame and is normally bound to a different portgroup from the Inside vNIC.



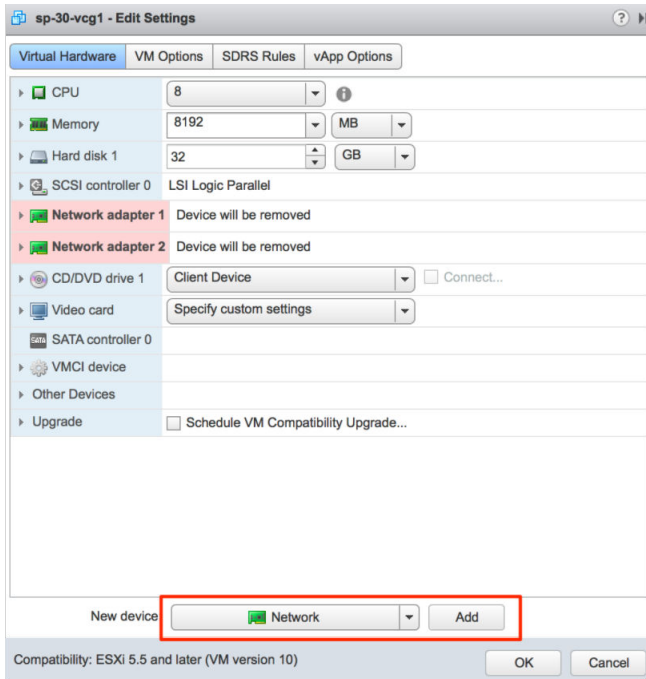
3 For the **Customize template** step, do not change anything. This is when you use vApp to configure the VM. We will not use vApp in this example. Click **Next** to continue with deploying the OVA.



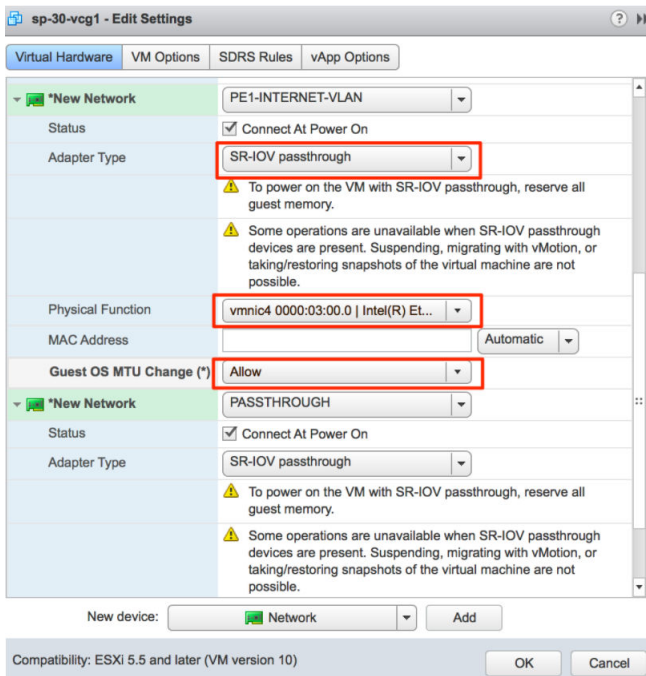
- Once the VM is successfully deployed, return to the VM and click **Edit Settings** . Two vNICs are created with adapter type = vmxnet3.



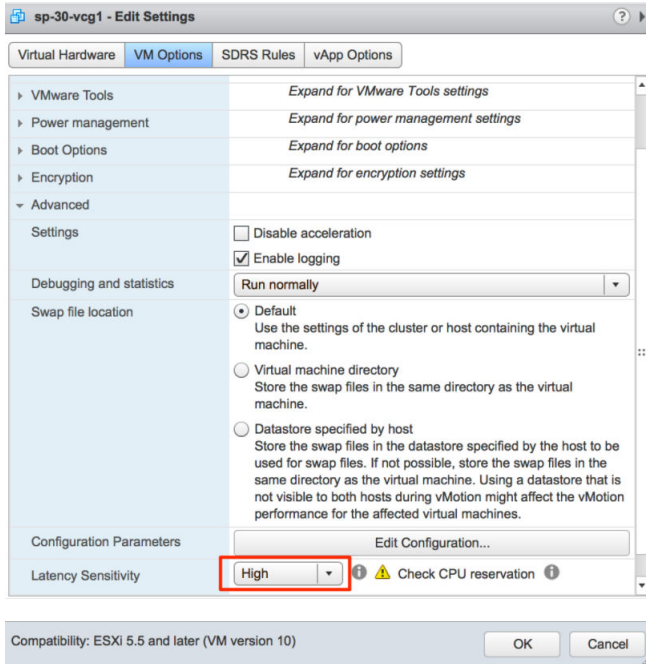
- (Optional for SR-IOV) This step is required only if you plan to use SR-IOV. Because the OVA by default creates the two vNICs as vmxnet3, we will need to remove the two vNICs and re-add them as SR-IOV.



When adding the two new SR-IOV vNICs, use the same portgroup as the original two vmxnet3 vNICs. Make sure the **Adapter Type** is **SR-IOV passthrough**. Select the correct physical port to use and set the **Guest OS MTU Change** to **Allow**. After you add the two vNICs, click **OK**.

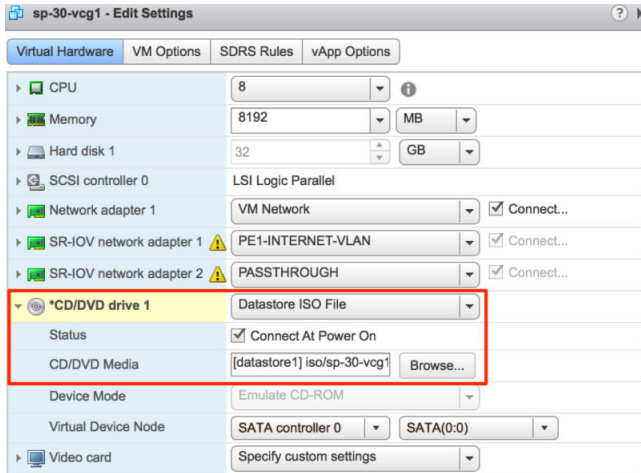


- Because VCG is a real-time application, you need to configure the **Latency Sensitivity** to **High**. For more information about how to configure the VM for real-time application, see <https://www.vmware.com/content/dam/digitalmarketing/vmware/en/pdf/techpaper/latency-sensitive-perf-vsphere55-white-paper.pdf>.



- 7 Refer to *Cloud-init Creation*. The Cloud-init file is packaged as a CD-ROM (iso) file. You need to mount this file as a CD-ROM.

Note You must upload this file to the datastore.



- 8 Start the VM.

Enable SR-IOV on KVM

This section describes how to enable the SR-IOV on KVM.

To enabling the SR-IOV on KVM:

- 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 the prompt by checking that Intel has the correct cpu flag.

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

- 2 Add the options on Bboot.

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

(in /etc/default/grub)

- a After this, use the Run command: `update-grub` and `update-initramfs -u`.
- b Reboot
- c 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
...
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->

Download on left 5.2.1

- a Download ixgbe from intel. Follow compile options.
- b 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 ixgbev
```

- d Remember to do `update-initramfs -u` and reboot.
- e Use `modinfo` to see if it is properly 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, you should see the interfaces.

Install VeloCloud Gateway on KVM

This section describes how to install the VCG qcow on KVM.

Pre-Installation Considerations

KVM provides multiple ways to provide networking to virtual machines. The networking in libvirt should be provisioned before the VM configuration. There are multiple ways to configure networking in KVM. For a full configuration of options on how to configure Networks on libvirt, please see the following link:

<https://libvirt.org/formatnetwork.html>

From the full list of options, the following are recommended by VeloCloud:

- SR-IOV (This mode is required for the VCG to deliver the maximum throughput specified by VeloCloud)
- OpenVSwitch Bridge

Validating SR-IOV (Optional)

If you decided to use SR-IOV, you can quickly verify if your host machine has it enabled.

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 decide to use SR-IOV mode, enable SR-IOV on KVM. To enable the SR-IOV on KVM, see [Enable SR-IOV on KVM](#).

Installation Steps

- 1 Copy the QCOW and the Cloud-init files created in the Cloud-Init Creation section to a new empty directory.
- 2 Create the Network pools that you are going to use for the device. Provided below is a sample of a pool using SR-IOV and a sample of a pool using OpenVswitch.

Using SR-IOV

```
git ./vcg/templates/KVM_NETWORKING_SAMPLES/template_outside_sriov.xml
```

```
<network>
  <name>public_interface</name> <!--This is the network name-->
  <forward mode='hostdev' managed='yes'>
    <pf dev='eth1' /> <!--Use the netdev name of your SR-IOV devices PF here-->
    <address type='pci' domain='0x0000' bus='0x06' slot='0x12' function='0x6' />
    <address type='pci' domain='0x0000' bus='0x06' slot='0x13' function='0x0' />
    <address type='pci' domain='0x0000' bus='0x06' slot='0x13' function='0x2' />
  </forward>
</network>
```

```
velocloud@KVMperf1:/images/automation/Local_Settings$ cat public.xml
<network>
  <name>public_interface</name> <!--This is the name of the network you created-->
  <forward mode='hostdev' managed='yes'>
    <pf dev='eth1' /> <!--Use the netdev name of your SR-IOV devices PF here-->
  </forward>
</network>

velocloud@KVMperf1:/images/automation/Local_Settings$ virsh net-define public.xml
Network public_interface defined from public.xml

velocloud@KVMperf1:/images/automation/Local_Settings$ virsh net-autostart public_interface
Network public_interface marked as autostarted

velocloud@KVMperf1:/images/automation/Local_Settings$ virsh net-start public_interface
Network public_interface started

velocloud@KVMperf1:/images/automation/Local_Settings$ virsh net-list
Name                State    Autostart  Persistent
-----
default             active   yes        yes
hole                active   no         no
lan                 active   no         no
ovs-net             active   no         no
passthrough         active   no         no
public_interface    active   yes        yes
```

Create a network for inside_interface.

```
git ./vcg/templates/KVM_NETWORKING_SAMPLES/template_inside_sriov.xml
```

```
<network>
  <name>inside_interface</name> <!--This is the network name-->
  <forward mode='hostdev' managed='yes'>
    <pf dev='eth2' /> <!--Use the netdev name of your SR-IOV devices PF here-->
    <address type='pci' domain='0x0000' bus='0x06' slot='0x12' function='0x0' />
    <address type='pci' domain='0x0000' bus='0x06' slot='0x12' function='0x2' />
    <address type='pci' domain='0x0000' bus='0x06' slot='0x12' function='0x4' />
  </forward>
</network>
```

Using OpenVSwitch

```
git ./vcg/templates/KVM_NETWORKING_SAMPLES/template_outside_openswitch.xml
```

```
<?xml version="1.0" encoding="UTF-8"?>
<network>
  <name>public_interface</name>
  <!--This is the network name-->
  <model type="virtio" />
  <forward mode="bridge" />
  <bridge name="publicinterface" />
  <virtualport type="openswitch" />
  <vlan trunk="yes">
    <tag id="50" />
    <!--Define all the VLANS for this Bridge -->
    <tag id="51" />
    <!--Define all the VLANS for this Bridge -->
  </vlan>
</network>
```

```
root@KVMperf1:/home/velocloud# ovs-vsctl add-br publicinterface
root@KVMperf1:/home/velocloud# cat public.xml

<network>
<name>public_interface</name> <!--This is the network name-->
  <model type='virtio'/>
<forward mode="bridge"/>
<bridge name="publicinterface"/>
<virtualport type='openswitch'/></virtualport>
<vlan trunk='yes'>
  <tag id='50'/> <!--Define all the VLANS for this Bridge -->
  <tag id='51'/> <!--Define all the VLANS for this Bridge -->
</vlan>
</network>
```

```
root@KVMperf1:/home/velocloud# virsh net-create public.xml
Network public_interface created from public.xml

root@KVMperf1:/home/velocloud#
```

Create a network for inside_interface:

```
git ./vcg/templates/KVM_NETWORKING_SAMPLES/template_inside_openswitch.xml
```

```
<network>
  <name>inside_interface</name> <!--This is the network name-->
  <model type='virtio'/>
  <forward mode="bridge"/>
  <bridge name="insideinterface"/>
  <virtualport type='openswitch'/></virtualport>
  <vlan trunk='yes'/></vlan>
  <tag id='200'/> <!--Define all the VLANS for this Bridge -->
  <tag id='201'/> <!--Define all the VLANS for this Bridge -->
  <tag id='202'/> <!--Define all the VLANS for this Bridge -->
</network>
```

- 3 Edit the VM XML. There are multiple ways to create a Virtual Machine in KVM. We are going to use the traditional way where we define the VM in an XML file and create it using libvirt. Below is a template that you can use for the XML file. You can create the XML file using:

```
vi my_vm.xml
```

Copy the template below and replace the sections in bold.

```
<?xml version="1.0" encoding="UTF-8"?>
<domain type="kvm">
  <name>#domain_name#</name>
  <memory unit="KiB">8388608</memory>
  <currentMemory unit="KiB">8388608</currentMemory>
  <vcpu>8</vcpu>
  <cputune>
    <vcpupin vcpu="0" cpuset=" 0" />
    <vcpupin vcpu="1" cpuset=" 1" />
    <vcpupin vcpu="2" cpuset=" 2" />
    <vcpupin vcpu="3" cpuset=" 3" />
    <vcpupin vcpu="4" cpuset=" 4" />
    <vcpupin vcpu="5" cpuset=" 5" />
    <vcpupin vcpu="6" cpuset=" 6" />
    <vcpupin vcpu="7" cpuset=" 7" />
  </cputune>
  <resource>
    <partition>/machine</partition>
  </resource>
  <os>
    <type>hvm</type>
  </os>
  <features>
    <acpi />
    <apic />
    <paе />
  </features>
  <cpu mode="host-passthrough" />
  <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>
    <disk type="file" device="disk">
      <driver name="qemu" type="qcow2" />
      <source file=" #folder#/#qcow_root#" />
      <target dev="hda" bus="ide" />
      <alias name="ide0-0-0" />
      <address type="drive" controller="0" bus="0" target="0" unit="0" />
    </disk>
    <disk type="file" device="cdrom">
      <driver name="qemu" type="raw" />
      <source file=" #folder#/#Cloud_ INIT_ ISO#" />
      <target dev="sdb" bus="sata" />
      <readonly />
    </disk>
  </devices>
</domain>
```

```

<alias name="sata1-0-0" />
<address type="drive" controller="1" bus="0" target="0" unit="0" />
</disk>
<controller type="usb" index="0">
  <alias name="usb0" />
  <address type="pci" domain="0x0000" bus="0x00" slot="0x01" function="0x2" />
</controller>
<controller type="pci" index="0" model="pci-root">
  <alias name="pci.0" />
</controller>
<controller type="ide" index="0">
  <alias name="ide0" />
  <address type="pci" domain="0x0000" bus="0x00" slot="0x01" function="0x1" />
</controller>
<interface type="network">
  <source network="public_interface" />
  <vlan>
    <tag id="#" #public_vlan#" />
  </vlan>
  <alias name="hostdev1" />
  <address type="pci" domain="0x0000" bus="0x00" slot="0x11" function="0x0" />
</interface>
<interface type="network">
  <source network="inside_interface" />
  <alias name="hostdev2" />
  <address type="pci" domain="0x0000" bus="0x00" slot="0x12" function="0x0" />
</interface>
<serial type="pty">
  <source path="/dev/pts/3" />
  <target port="0" />
  <alias name="serial0" />
</serial>
<console type="pty" tty="/dev/pts/3">
  <source path="/dev/pts/3" />
  <target type="serial" port="0" />
  <alias name="serial0" />
</console>
<memballoon model="none" />
</devices>
<seclabel type="none" />
</domain>

```

4 Launch the VM.

- a Verify the basic networks are created and **active**.

```

velocloud@KVMperf2:/tmp/VeloCloudGateway$ virsh net-list
Name          State    Autostart  Persistent
-----
default       active   yes        yes
inside_interface active   no         no
passthrough   active   no         no
public_interface active   no         no
velocloud@KVMperf2:/tmp/VeloCloudGateway$

```

```

velocloud@KVMperf2:/tmp/VeloCloudGateway$ ls -lrt
total 2107400
-rw-r--r-- 1 velocloud velocloud 2157576192 Dec  6 12:20 vcg-root.img
-rw-rw-r-- 1 velocloud velocloud    1990 Dec  6 12:25 user-data
-rw-rw-r-- 1 velocloud velocloud     336 Dec  6 12:29 meta-data
-rw-rw-r-- 1 velocloud velocloud  374784 Dec  6 12:31 vcg-test.iso
-rw-rw-r-- 1 velocloud velocloud    2674 Dec  6 12:34 test_vcg.xml
-rw-rw-r-- 1 velocloud velocloud     219 Dec  6 12:37 public.xml
-rw-rw-r-- 1 velocloud velocloud     219 Dec  6 12:38 private.xml
velocloud@KVMperf2:/tmp/VeloCloudGateway$

```

Main Files

- vcg-root (qcow file)
- vcg-test.iso (cloud-init)
- test_vcg.xml (XML file that defines the VM)

Define VM

```

velocloud@KVMperf2:/tmp/VeloCloudGateway$ virsh define test_vcg.xml
Domain test_vcg defined from test_vcg.xml

```

Set VM to autostart

```

velocloud@KVMperf2:/tmp/VeloCloudGateway$ virsh define test_vcg.xml
Domain test_vcg defined from test_vcg.xml

```

Start VM

```

velocloud@KVMperf2:/tmp/VeloCloudGateway$ virsh define test_vcg.xml
Domain test_vcg defined from test_vcg.xml

```

- 5 Console into the VM.

```

virsh list
Id Name State
-----
25 test_vcg running
velocloud@KVMperf2$ virsh console 25
Connected to domain test_vcg
Escape character is ^]

```

Special Consideration for KVM Host

- Disable GRO (Generic Receive Offload) on physical interfaces (to avoid unnecessary re-fragmentation in VCG).

```

ethtool -K <interface> gro off tx off

```

- Disable CPU C-states (power states affect real-time performance). Typically, this can be done as part of kernel boot options by appending `processor.max_cstate=1` or just disable in the BIOS. For more information, see https://docs.fedoraproject.org/en-US/Fedora/13/html/Virtualization_Guide/chap-Virtualization-KVM_guest_timing_management.html.

- For production deployment, vCPUs must be pinned to the instance. No oversubscription on the cores should be allowed to take place. For more information, see https://docs.fedoraproject.org/en-US/Fedora/13/html/Virtualization_Guide/ch25s06.html.

Post-Installation Tasks

This section describes post-installation and installation verification steps.

If everything worked as expected in the installation, you can now login to the VM.

- 1 If everything works as expected, you should see the login prompt on the console. You should see the prompt name as specified in cloud-init.

```

Ubuntu 14.04.5 LTS sp-30-vcg1 tty1
sp-30-vcg1 login: * Starting execute cloud user/final scripts      [ OK ]

Ubuntu 14.04.5 LTS sp-30-vcg1 tty1
sp-30-vcg1 login: _

```

- 2 You can also take a look at `/var/log/cloud-init.log`. If you see the message below, it is likely that the cloud init runs successfully.

```

Nov 24 00:28:50 sp-30-vcg1 [CLOUDINIT] helpers.py[DEBUG]: Running config-final-message using lock (<cloudinit.helpers.DummyLock object at 0x7f5246d0bc10>)
Nov 24 00:28:50 sp-30-vcg1 [CLOUDINIT] util.py[DEBUG]: Reading from /proc/uptime (quiet=False)
Nov 24 00:28:50 sp-30-vcg1 [CLOUDINIT] util.py[DEBUG]: Read 13 bytes from /proc/uptime
Nov 24 00:28:50 sp-30-vcg1 [CLOUDINIT] util.py[DEBUG]: === Cloud-init completed ===
Nov 24 00:28:50 sp-30-vcg1 [CLOUDINIT] util.py[DEBUG]: Writing to /var/lib/cloud/instance/boot-finish
Nov 24 00:28:50 sp-30-vcg1 [CLOUDINIT] helpers.py[DEBUG]: config-power-state-change already ran (frequency=once-per-instance)
Nov 24 00:28:50 sp-30-vcg1 [CLOUDINIT] helpers.py[DEBUG]: Running config-velocloud using lock (<cloudinit.helpers.DummyLock object at 0x7f5246d0b9d0>)
Nov 24 00:28:50 sp-30-vcg1 [CLOUDINIT] cc_velocloud.py[DEBUG]: in Velocloud vcg velocloud
Nov 24 00:28:50 sp-30-vcg1 [CLOUDINIT] cc_velocloud.py[DEBUG]: No activation configuration
Nov 24 00:28:50 sp-30-vcg1 [CLOUDINIT] cloud-init[DEBUG]: Ran 12 modules with 0 failures
Nov 24 00:28:50 sp-30-vcg1 [CLOUDINIT] util.py[DEBUG]: Creating symbolic link from '/run/cloud-init/result.json' => './../var/lib/cloud/data/result.json'
Nov 24 00:28:50 sp-30-vcg1 [CLOUDINIT] util.py[DEBUG]: Reading from /proc/uptime (quiet=False)
Nov 24 00:28:50 sp-30-vcg1 [CLOUDINIT] util.py[DEBUG]: Read 13 bytes from /proc/uptime
Nov 24 00:28:50 sp-30-vcg1 [CLOUDINIT] util.py[DEBUG]: cloud-init mode 'modules' took 0.427 seconds

```

- 3 Verify that the VeloCloud Gateway is registered with VCO.

```

root@vcg1:/home/vcadmin# /opt/vc/bin/is_activated.py
True
root@vcg1:/home/vcadmin#

```

- 4 Verify Outside Connectivity.


```
root@vcg1:/home/vcadmin# ping 8.8.8.8
PING 8.8.8.8 (8.8.8.8) 56(84) bytes of data:
64 bytes from 8.8.8.8: icmp_seq=1 ttl=57 time=7.06 ms
^C
--- 8.8.8.8 ping statistics ---
1 packets transmitted, 1 received, 0% packet loss, time 0ms
rtt: min/avg/max/mdev = 7.060/7.060/7.060/0.000 ms
root@vcg1:/home/vcadmin#
```

- 5 Verify that the MGMT VRF is responding to ARPs.

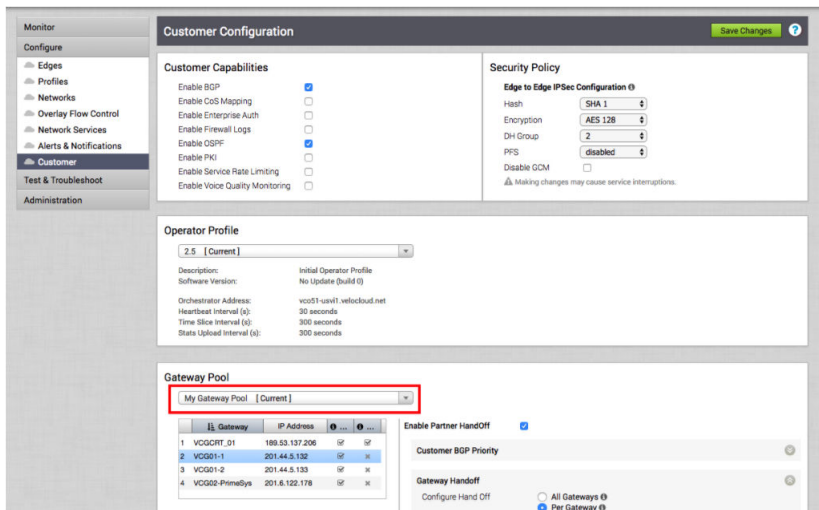
```
ubuntu@ubuntu:~$ sudo /opt/vc/bin/tcpdump.sh -i eth1
tcpdump: WARNING: tcpdump: no IPv4 address assigned
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on tcpdump, link-type EN10MB (Ethernet), capture size 65535 bytes
08:35:50.013411 ARP, Request who-has 27.0.0.2 tell 27.0.0.1, length 28
08:35:50.013420 ARP, Reply 27.0.0.2 is-at 54:7f:ee:da:c4:7c (oui Unknown), length 42
08:35:55.013411 ARP, Request who-has 27.0.0.2 tell 27.0.0.1, length 28
08:35:55.013420 ARP, Reply 27.0.0.2 is-at 54:7f:ee:da:c4:7c (oui Unknown), length 42
^C
4 packets captured
4 packets received by filter
0 packets dropped by kernel
```

- 6 Remove cloud-init so it doesn't run again.

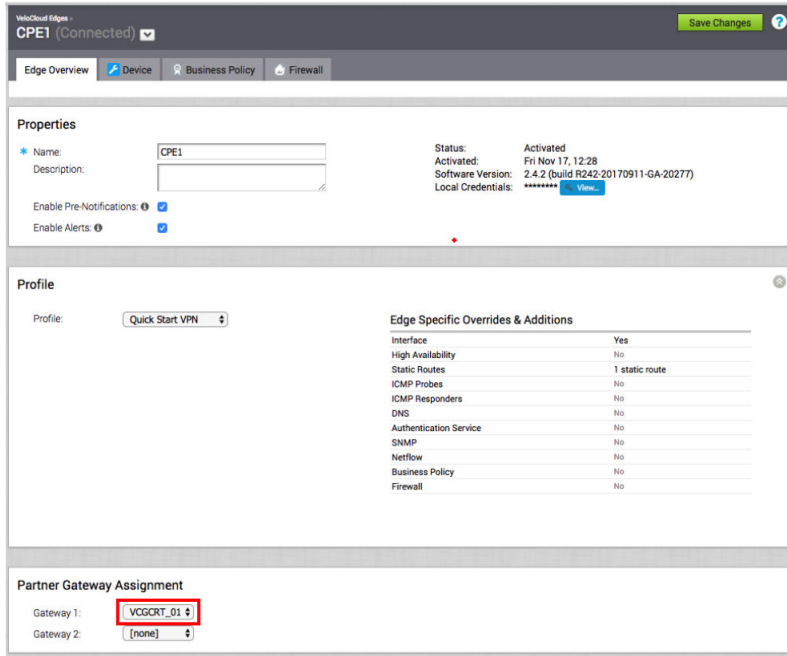
Note If you have deployed OVA on VeloCloudvSphere with vAPP properties, you must disable cloud-init prior to upgrading to versions 4.0.1 or 4.1.0. This is to ensure that the customization settings such as network configuration or password are not lost during the upgrade.

```
apt-get purge cloud-init
```

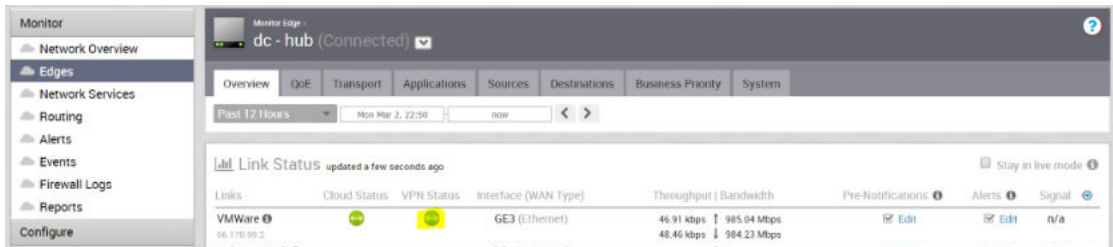
- 7 Associate the new gateway pool, (created in the section titled, “ *Creating a Gateway and getting the Activation Key*”) with the customer.



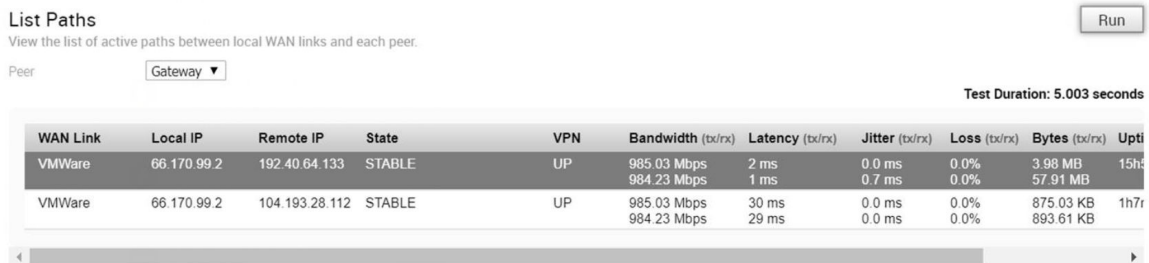
- 8 Associate the Gateway with an Edge.



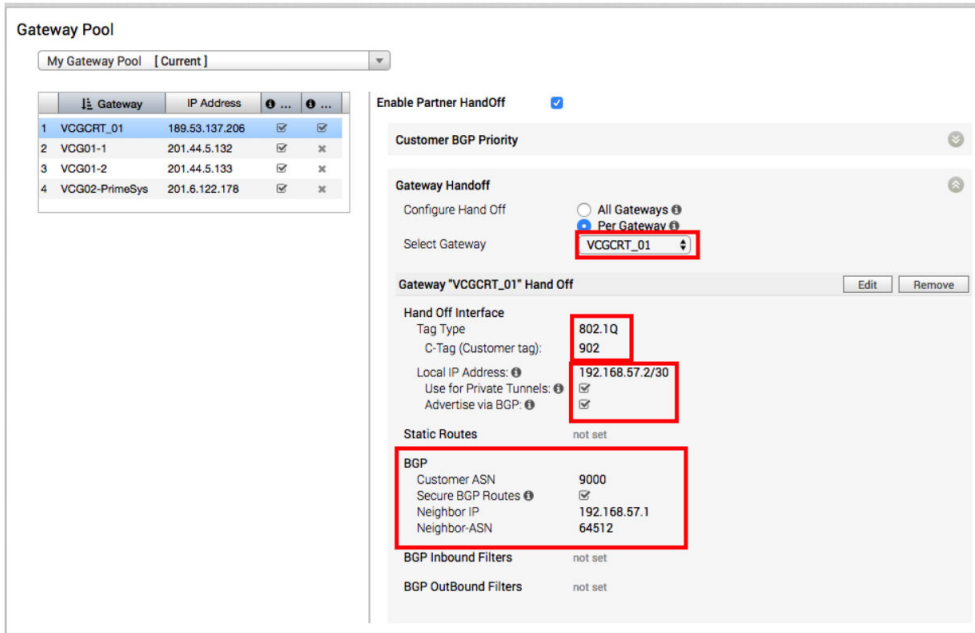
- Verify that the Edge is able to establish a tunnel with the Gateway on the Internet side. From the VMware SD-WAN Orchestrator, go to **Monitor > Edges > Overview**.



From the VMware SD-WAN Orchestrator, go to **Test & Troubleshoot > Remote Diagnostics > [Edge] > List Paths**, and click **Run** to view the list of active paths.



- Configure the Handoff interface.



11 Verify that the BGP session is up.

BGP Gateway Neighbor State Auto refresh: Paused									
Gateway	Neighbor IP	State	State Changed Time	Msg Received	Msg Sent	Events	Up/Down	Prefix Received	
<input type="checkbox"/> VCGCRT_01	192.168.57.33	ESTABLISHED	Fri Nov 17, 10:34:28 13 days ago	20724	18899	7 View	01w6d01h	8	

Upgrade VeloCloud Gateway

This section describes how to upgrade a VeloCloud Gateway installation.

To upgrade a VeloCloud Gateway installation:

- 1 Download the VeloCloud Gateway Update package.
- 2 Upload the image to the VeloCloud Gateway system (using, for example, the scp command). Copy the image to the following location on the system: `/var/lib/velocloud/software_update/vcg_update.tar`.
- 3 Connect to the VeloCloud Gateway console and run:

```
sudo /opt/vc/bin/vcg_software_update
```

Custom Configurations

This section describes custom configurations.

NTP Configuration

NTP configuration involves editing the `/etc/ntp.conf` file.

Userdata

This section describes userdata.

```
#cloud-config
hostname: #_Hostname_#
password: #_Console_Password_#
chpasswd: {expire: False}
ssh_pwauth: True
ssh_authorized_keys:
- #_SSH_public_Key_#
runcmd:
- 'echo "[]" > /opt/vc/etc/vc_blocked_subnets.json'
- 'sed -iorig "s/wan=\.*\/wan=\`eth0 eth1\`/" /etc/config/gatewayd-tunnel'
- '/var/lib/cloud/scripts/per-boot/config_gateway'
- 'sleep 10'
- '/opt/vc/bin/vc_procmon restart'
write_files:
- path: "/etc/ntp.conf"
  permissions: '0644'
  content: |
# Use servers from the NTP Pool Project. Approved by Ubuntu Technical Board
# on 2011-02-08 (LP: #104525). See http://www.pool.ntp.org/join.html for
# more information.
server #_NTP_SERVER_1_#
server #_NTP_SERVER_2_#
server 1.ubuntu.pool.ntp.org iburst
server 2.ubuntu.pool.ntp.org iburst
server 3.ubuntu.pool.ntp.org iburst
- path: "/var/lib/cloud/scripts/per-boot/config_gateway"
  permissions: '0777'
  content: |
#!/usr/bin/python
import json
import commands
is_activated = commands.getoutput("/opt/vc/bin/is_activated.py")
if "True" in str(is_activated):
print "Gateway already activated"
exit
commands.getoutput("/opt/vc/bin/activate.py -s #_VCO_# #_Activation_Key_#")
### EDIT GATEWAYD ###
with open("/etc/config/gatewayd", "r") as jsonFile:
data = json.load(jsonFile)
data["global"]["vcmp.interfaces"] = ["eth0"]
data["global"]["wan"] = ["eth1"]
# NOTE FOR HAND OFF IT CAN BE "QinQ (0x8100)" "QinQ (0x9100)" "none" "802.1Q" "802.1ad"
data["vrf_vlan"]["tag_info"][0]["mode"] = "#_Handoff_"
data["vrf_vlan"]["tag_info"][0]["interface"] = "eth1"
data["vrf_vlan"]["tag_info"][0]["c_tag"] = "#_C_TAG_FOR_MGMT_VRF_#"
data["vrf_vlan"]["tag_info"][0]["s_tag"] = "#_S_TAG_FOR_MGMT_VRF_"
with open("/etc/config/gatewayd", "w") as jsonFile:
jsonFile.write(json.dumps(data,sort_keys=True,indent=4, separators=(",", ": ")))
### EDIT DPKD ###
with open("/opt/vc/etc/dpdk.json", "r") as jsonFile:
data = json.load(jsonFile)
```

```
#SET 0 or 1 for enabled or DISABLED example data["dpdk_enabled"] = 0
data["dpdk_enabled"] = #_DKDP_ENABLED_OR_DISABLED_#
with open("/opt/vc/etc/dpdk.json", "w") as jsonFile:
jsonFile.write(json.dumps(data,sort_keys=True,indent=4, separators=(",", ": ")))
final_message: "==== Cloud-init completed ====="
power_state:
delay: "+1"
mode: reboot
message: Bye Bye
timeout: 30
condition: True
```

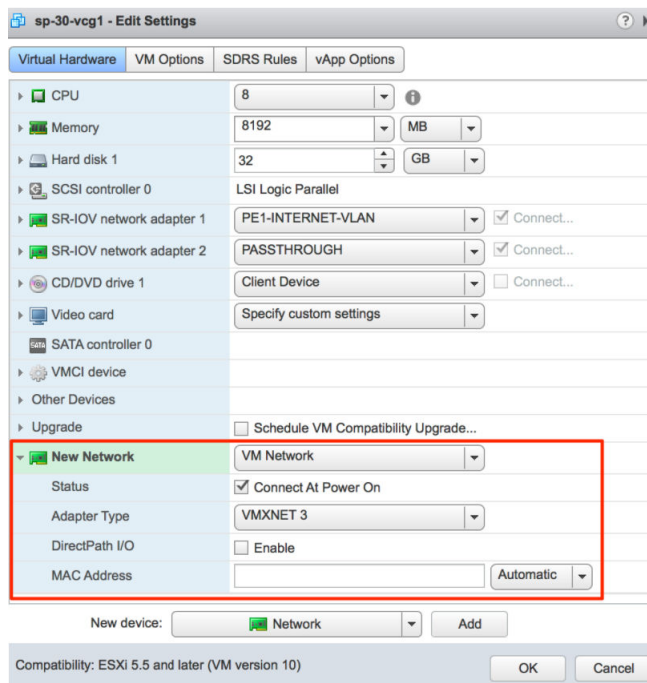
OAM Interface and Static Routes

If Gateways are to be deployed with an OAM interface, complete the following steps.

- 1 Add an additional interface to the VM (ETH2).

VMware

If a dedicated vNIC for Management/OAM is desired, add another vNIC of type vmxnet3. You must repeat the previous step, which is to click **OK** and then **Edit Settings** again so you can make a note of the vNIC MAC address.



KVM

If a dedicated vNIC for Management/OAM is desired, make sure you have a libvirt network named oam-network. Then add the following lines to your XML VM structure:

```
...
</controller>
<interface type='network'>
```

```

<source network='public_interface'/>
<vlan><tag id='#public_vlan#'/></vlan>
<alias name='hostdev1'/>
<address type='pci' domain='0x0000' bus='0x00' slot='0x11' function='0x0'/>
</interface>
<interface type='network'>
<source network='inside_interface'/>
<alias name='hostdev2'/>
<address type='pci' domain='0x0000' bus='0x00' slot='0x12' function='0x0'/>
</interface>
<interface type='network'>
<source network='oam_interface'/>
<vlan><tag id='#oam_vlan#'/></vlan>
<alias name='hostdev2'/>
<address type='pci' domain='0x0000' bus='0x00' slot='0x13' function='0x0'/>
</interface>
<serial type='pty'>
<source path='/dev/pts/3'/>
<target port='0'/>
<alias name='serial0'/>
</serial>

```

2 Configure the meta-data file with the additional interface.

```

instance-id: #_Hostname_#
local-hostname: #_Hostname_#
network-interfaces: |
auto eth0
iface eth0 inet static
address #_IPv4_Address_#
netmask #_IPv4_Netmask_#
gateway #_IPv4_Gateway_#
dns-nameservers #_DNS_server_primary_# #_DNS_server_secondary_#
auto eth1
iface eth1 inet static
metric '13'
address #_MGMT_IPv4_Address_#
netmask #_MGMT_IPv4_Netmask_#
gateway #_MGMT_IPv4_Gateway_#
dns-nameservers #_DNS_server_primary_# #_DNS_server_secondary_#
auto eth2
iface eth2 inet static
address #_OAM_IPv4_Address_#
netmask #_OAM_IPv4_Netmask_#
up route add -net 10.0.0.0 netmask 255.0.0.0 gw #_OAM_IPv4_Gateway_#
up route add -net 192.168.0.0 netmask 255.255.0.0 gw #_OAM_IPv4_Gateway_#
dns-nameservers #_DNS_server_primary_# #_DNS_server_secondary_#

```

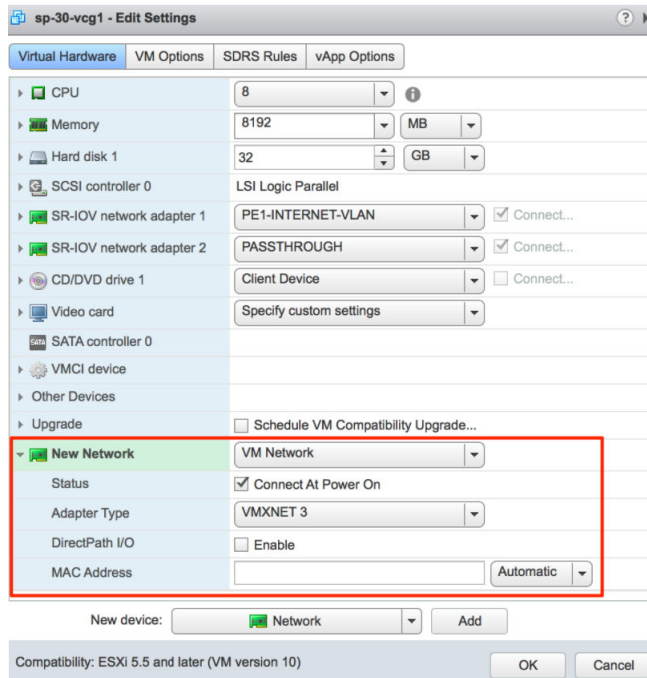
OAM - SR-IOV with vmxnet3 or SR-IOV with VIRTIO

It is possible in some installations to mix and match and provide different interface types for the Gateway. This generally happens if you have an OAM without SR-IOV. This custom configuration requires additional steps since this causes the interfaces to come up out of order.

- 1 Record the MAC address of each interface.

VMWare

After creating the machine, go to **Edit Settings** and copy the Mac address.



KVM

After defining the VM, perform the following command:

```
velocloud@KVMperf1:/tmp$ virsh dumpxml vcg_1 | egrep 'interface|network|mac address'
<interface type='hostdev' managed='yes'>
  <mac address='52:54:00:85:5e:98'/>
</interface>
<interface type='hostdev' managed='yes'>
  <mac address='52:54:00:80:3f:36'/>
</interface>
velocloud@KVMperf1:/tmp$
```

- 2 Edit the user-data and lock the mac address to the interface order. This is done by adding the additional lines in bold:

Userdata

```
#cloud-config
hostname: #_Hostname_#
password: #_Console_Password_#
chpasswd: {expire: False}
ssh_pwauth: True
ssh_authorized_keys:
```

```

- #_SSH_public_Key_#
runcmd:
- 'echo "[]" > /opt/vc/etc/vc_blocked_subnets.json'
- 'sed -iorig "s/wan=\\".*\\/wan=\\"eth0 eth1\\"/" /etc/config/gatewayd-tunnel'
- '/var/lib/cloud/scripts/per-boot/config_gateway'
- 'sleep 10'
- '/opt/vc/bin/vc_procmon restart'
write_files:
- path: "/etc/udev/rules.d/70-persistent-net.rules"
  permissions: '0644'
  content: |
    SUBSYSTEM=="net", ACTION=="add", DRIVERS=="?*", ATTR{address}=="#_ETH0_MAC_ADDRESS_#",
ATTR{type}=="1", KERNEL=="eth*", NAME="eth0"
    SUBSYSTEM=="net", ACTION=="add", DRIVERS=="?*", ATTR{address}=="#_ETH1_MAC_ADDRESS_#",
ATTR{type}=="1", KERNEL=="eth*", NAME="eth1"
    # NOTE ETH2 IS OAM IF NO OAM PRESENT THEM REMOVE
    SUBSYSTEM=="net", ACTION=="add", DRIVERS=="?*", ATTR{address}=="#_ETH2_MAC_ADDRESS_#",
ATTR{type}=="1", KERNEL=="eth*", NAME="eth2"
- path: "/var/lib/cloud/scripts/per-boot/config_gateway"
  permissions: "0777"
  content: |
#!/usr/bin/python
import json
import commands
is_activated = commands.getoutput("/opt/vc/bin/is_activated.py")
if "True" in str(is_activated):
print "Gateway already activated"
exit
commands.getoutput("/opt/vc/bin/activate.py -s #_VCO_# #_Activation_Key_#")

### EDIT GATEWAYD ###
with open("/etc/config/gatewayd", "r") as jsonFile:
data = json.load(jsonFile)
data["global"]["vcmp.interfaces"] = ["eth0"]
data["global"]["wan"] = ["eth1"]
# NOTE FOR HAND OFF IT CAN BE "QinQ (0x8100)" "QinQ (0x9100)" "none" "802.1Q" "802.1ad"
data["vrf_vlan"]["tag_info"][0]["mode"] = "#_Handoff_"
data["vrf_vlan"]["tag_info"][0]["interface"] = "eth1"
data["vrf_vlan"]["tag_info"][0]["c_tag"] = "#_C_TAG_FOR_MGMT_VRF_#"
data["vrf_vlan"]["tag_info"][0]["s_tag"] = "#_S_TAG_FOR_MGMT_VRF_"
with open("/etc/config/gatewayd", "w") as jsonFile:
jsonFile.write(json.dumps(data,sort_keys=True,indent=4, separators=(",", ": ")))

### EDIT DPDK ###
with open("/opt/vc/etc/dpdk.json", "r") as jsonFile:
data = json.load(jsonFile)
#SET 0 or 1 for enabled or DISABLED example data["dpdk_enabled"] = 0
data["dpdk_enabled"] = #_DKDP_ENABLED_OR_DISABLED_#
with open("/opt/vc/etc/dpdk.json", "w") as jsonFile:
jsonFile.write(json.dumps(data,sort_keys=True,indent=4, separators=(",", ": ")))
final_message: "==== Cloud-init completed ====="
power_state:
delay: "+1"

```



```

mode: reboot
message: Bye Bye
timeout: 30
condition: True

```

Special Consideration When Using 802.1ad Encapsulation

It seems certain that 802.1ad devices do not populate the outer tag EtherType with 0x88A8. Special change is required in user data to interoperate with these devices.

Assuming a Management VRF is configured with S-Tag: 20 and C-Tag: 100, edit the vrf_vlan section in /etc/config/gatewayd as follows. Also, define resp_mode to 1 so that the VCG will relax its check to allow Ethernet frames that have incorrect EtherType of 0x8100 in the outer header.

```

#cloud-config
hostname: #_Hostname_#
password: #_Console_Password_#
chpasswd: {expire: False}
ssh_pwauth: True
ssh_authorized_keys:
- #_SSH_public_Key_#
runcmd:
- 'echo "[]" > /opt/vc/etc/vc_blocked_subnets.json'
- 'sed -i.orig "s/wan=\\.*\\/wan=\\\"eth0 eth1\\\"/" /etc/config/gatewayd-tunnel'
- '/var/lib/cloud/scripts/per-boot/config_gateway'
- 'sleep 10'
- '/opt/vc/bin/vc_procmon restart'
write_files:
- path: "/var/lib/cloud/scripts/per-boot/config_gateway"
permissions: '0777'
content: |
#!/usr/bin/python
import json
import commands
is_activated = commands.getoutput("/opt/vc/bin/is_activated.py")
if "True" in str(is_activated):
print "Gateway already activated"
exit
commands.getoutput("/opt/vc/bin/activate.py -s #_VCO_# #_Activation_Key_#")

### EDIT GATEWAYD ###
with open("/etc/config/gatewayd", "r") as jsonFile:
data = json.load(jsonFile)
data["global"]["vcmp.interfaces"] = ["eth0"]
data["global"]["wan"] = ["eth1"]
# NOTE FOR HAND OFF IT CAN BE "QinQ (0x8100)" "QinQ (0x9100)" "none" "802.1Q" "802.1ad"
data["vrf_vlan"]["tag_info"][0]["resp_mode"] = "1"
data["vrf_vlan"]["tag_info"][0]["mode"] = "#_Handoff_"
data["vrf_vlan"]["tag_info"][0]["interface"] = "eth1"
data["vrf_vlan"]["tag_info"][0]["c_tag"] = "#_C_TAG_FOR_MGMT_VRF_#"
data["vrf_vlan"]["tag_info"][0]["s_tag"] = "#_S_TAG_FOR_MGMT_VRF_"
with open("/etc/config/gatewayd", "w") as jsonFile:
jsonFile.write(json.dumps(data,sort_keys=True,indent=4, separators=(",", " ": )))

```

```

### EDIT DPDK ###
with open("/opt/vc/etc/dpdk.json", "r") as jsonFile:
    data = json.load(jsonFile)
#SET 0 or 1 for enabled or DISABLED example data["dpdk_enabled"] = 0
data["dpdk_enabled"] = #_DKDP_ENABLED_OR_DISABLED_#
with open("/opt/vc/etc/dpdk.json", "w") as jsonFile:
    jsonFile.write(json.dumps(data,sort_keys=True,indent=4, separators=(",", ": ")))
final_message: "==== Cloud-init completed ====="
power_state:
delay: "+1"
mode: reboot
message: Bye Bye
timeout: 30
condition: True

```

SNMP Integration

This section describes how to configure SNMP integration.

To configure SNMP integration:

- 1 Edit `/etc/snmp/snmpd.conf`. Add the following lines to the config with source IP of the systems that will be connecting to SNMP service.

The following example will configure access to all counters from localhost via community string `vc-vcg` and from `10.0.0.0/8` with community string `myentprisecommunity` using SNMPv2c version. For more information, see the Net-SNMP documentation.

```

agentAddress udp:161
# com2sec sec.name source community
com2sec local localhost vc-vcg
com2sec myenterprise 10.0.0.0/8 myentprisecommunity# group access.name sec.model sec.name
group rogroup v2c local
group rogroup v2c myenterpriseview all included .1 80
# access access.name context sec.model sec.level match read write notif
access rogroup "" any noauth exact all none none#sysLocation Sitting on the Dock of the Bay
#sysContact Me <me@example.org>sysServices 72master agentx#
# Process Monitoring
## At least one 'gwd' process
proc gwd
# At least one 'mgd' process
proc mgd#
# Disk Monitoring
#
# 100MBs required on root disk, 5% free on /var, 10% free on all other disks
disk / 100000
disk /var 5%
includeAllDisks 10%#
# System Load
#
# Unacceptable 1-, 5-, and 15-minute load averages
load 12 10 5

```

- 2 Edit `/etc/snmp/snmpd.conf`. Add the following lines to the config with the source IP of the systems that will be connecting to SNMP service:

```
# WARNING: only add targeted rules for addresses and ports
# do not add blanket drop or accept rules since VCG will append its own rules
# and that may prevent it from functioning properly
*filter
:INPUT ACCEPT [0:0]
-A INPUT -p udp -m udp --source 127.0.0.1 --dport 161 -m comment --comment "allow SNMP port" -j
ACCEPT
-A INPUT -p udp -m udp --source 10.0.0.0/8 --dport 161 -m comment --comment "allow SNMP port" -j
ACCEPT
:FORWARD ACCEPT [0:0]
:OUTPUT ACCEPT [0:0]
COMMIT
```

- 3 Restart snmp and iptables services:

```
service snmpd restart
service iptables-persistent restart
service vc_process_monitor restart
```

Custom Firewall Rules

This section describes how to modify custom firewall rules.

To modify local firewall rules, edit the following file: `/etc/iptables/rules.v4`

Important Add only targeted rules for addresses and ports. Do NOT add blanket drop or accept rules. VCG will append its own rules to the table and, because the rules are evaluated in order, that may prevent Gateway software from functioning properly.

```
*filter
:INPUT ACCEPT [0:0]
-A INPUT -p udp -m udp --source 127.0.0.1 --dport 161 -m comment --comment "allow SNMP port" -j
ACCEPT
:FORWARD ACCEPT [0:0]
:OUTPUT ACCEPT [0:0]
COMMIT
Restart iptables service:
service iptables-persistent restart
service vc_process_monitor restart
```