Orchestrator Guide

VeloCloud SD-WAN Deployment and Monitoring

Version 5.2

Arista.com

Headquarters	Support	Sales
5453 Great America Parkway Santa Clara, CA 95054 USA +1-408-547-5500	+1-408-547-5502 +1-866-476-0000	+1-408-547-5501 +1-866-497-0000
www.arista.com/en/	support@arista.com	sales@arista.com

Contents

Chapter 1

Arista VeloCloud Orchestrator Deployment and Monitoring Guide

The Arista VeloCloud SD-WAN Orchestrator Deployment and Monitoring Guide provides guidance on how to install, run, and monitor the Orchestrator.

Chapter 2

Overview

The VeloCloud Orchestrator Deployment and Monitoring Guide provides the following information:

- How to install the VeloCloud Orchestrator
- How to setup Disaster Recovery
- How to upgrade the VeloCloud Orchestrator
- How to back up the VeloCloud Orchestrator application Data
- How to monitor the VeloCloud Orchestrator application
- How to tune various system properties (depending on the scale of the deployment)

Chapter 3

Install VeloCloud Orchestrator

This section discusses the VeloCloud Orchestrator installation.

3.1 Prerequisites

This section discusses the prerequisites that must be met before installing the VeloCloud Orchestrator.

3.1.1 Instance Requirements

Arista recommends installation of the Orchestrator and Gateway applications as a virtual machine (i.e., guest instance) on an existing hypervisor.

The VeloCloud Orchestrator requires the following minimal guest instance specifications:

- 8 Intel vCPU's at 2.5 Ghz or higher

  Note: Although we recommend using Intel Xeon processors, similar Intel or AMD processors having the same or greater CPU frequency are also acceptable.

- 64 GB of memory
- Required Minimum IOPS: 5,000 IOPS
- VeloCloud Orchestrator requires 4 SSD based persistent volumes (expandable through LVM if needed)
  - 192GB x 1 - Root
  - 1TB x 1 - Store
  - 500GB x 1 - Store2
  - 1TB x 1 - Store3
- 1 Gbps NIC
- Ubuntu x64 server VM compatibility
- Single public IP address (Can be made available through NAT)

3.1.2 Upstream Firewall Configuration

The upstream firewall needs to be configured to allow inbound HTTP (TCP/80) as well as HTTPS (TCP/443). If a stateful firewall is in place, established connections that are outbound originated should also be allowed to facilitate upgrades and security updates.

3.1.3 External Services

The VeloCloud Orchestrator relies on several external services. Before proceeding with an installation, ensure that licenses are available for each of the services.

Google Maps

Google Maps is used for displaying Edges and data centers on a map. No account needs to be created with Google to utilize the functionality. However, Internet access must be available to the VeloCloud Orchestrator instance in order for the service to be available.

The service is limited to 25,000 map loads each day, for more than 90 consecutive days. Arista does not anticipate exceeding these limits for nominal use of the VeloCloud Orchestrator.

Twilio

Twilio is used for SMS-based alerting to enterprise customers to notify them of Edge or link outage events. An account needs to be created and funded at http://www.twilio.com.

The account can be provisioned in the VeloCloud Orchestrator through the Operator Portal's **System Properties** page. The account will be provisioned through a system property, as described later in the guide.

MaxMind

MaxMind is a geolocation service. It is used to automatically detect Edge and Gateway locations and ISP names based on IP address. If this service is deactivated, then geolocation information will need to be updated manually. The account can be provisioned in the VeloCloud Orchestrator through the Operator Portal's **System Properties** page.

For additional information, see Configure System Properties.

3.2 Installation Procedures

This section discusses installation.

3.2.1 Cloud-init Preparation

This section discusses how to use the cloud-init package to handle the early initialization of instances.

About cloud-init

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

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

The VeloCloud Orchestrator supports cloud-init and all essential configurations can be packaged in an ISO image.

Create the Cloud-init meta-data File

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

```
instance-id: vco01 local-hostname: vco-01
```

Additionally, you can specify network interface information (if the network is not configured via DHCP, for example):

```
instance-id: vco01 local-hostname: vco-01 network-interfaces: | auto eth0 iface eth0 inet static
 address 10.0.1.2 network 10.0.1.0 netmask 255.255.255.0 broadcast 10.0.1.255 gateway 10.0.1.1
```

Create the Cloud-init User-data File

The second installation configuration option file is the user data file. This file provides information about users on the system. Create it with a text editor and call it user-data. This file will be used to enable access to the installation of VeloCloud Orchestrator. The following is an example of what the user-data file will look like:

```
#cloud-config password: Velocloud123 chpasswd: {expire: False} ssh_pwauth: True ssh_authorize
d_keys: - ssh-rsa AAA...SDvz user1@yourdomain.com - ssh-rsa AAB...QTuo user2@yourdomain.com vco:
 super_users: list: | user1@yourdomain.com:password1 remove_default_users: True system_proper
ties: list: | mail.smtp.port:34 mail.smtp.host:smtp.yourdomain.com service.maxmind.enable:True
 service.maxmind.license:todo_license service.maxmind.userid:todo_user service.twilio.phoneNumber
:222123123 network.public.address:222123123 write_files: - path: /etc/nginx/velocloud/ssl/
server.crt permissions: '0644' content: "-----BEGIN CERTIFICATE-----\nMI….ow==\n-----END
 CERTIFICATE-----\n" - path: /etc/nginx/velocloud/ssl/server.key permissions: '0600' content:
 "-----BEGIN RSA PRIVATE KEY-----\nMII...D/JQ==\n-----END RSA PRIVATE KEY-----\n" - path: /etc/
nginx/velocloud/ssl/velocloudCA.crt
```

This user-data file enables the default user, *vcadmin*, to login either with a password or with an SSH key. The use of both methods is possible, but not required. The password login is enabled by the password and chpasswd lines.

- The password contains the plain-text password for the vcadmin user.
- The chpasswd line turns off password expiration to prevent the first login from immediately prompting for a change of password. This is optional.

Note: If you set a password, it is recommended that you change it when you first log in because the password has been stored in a plain text file.

The `ssh_pwauth` line enables SSH login. The `ssh_authorized_keys` line begins a block of one or more authorized keys. Each public SSH key listed on the ssh-rsa lines will be added to the `vcadmin ~/.ssh/authorized_keys` file.

In this example, two keys are listed. For this example, the key has been truncated. In a real file, the entire public key must be listed. Note that the ssh-rsa lines must be preceded by two spaces, followed by a hyphen, followed by another space.

The *vco* section specifies configured VeloCloud Orchestrator services.

super_users contains list of Arista Super Operator accounts and corresponding passwords.

The *system_properties* section allows to customize Orchestrator System Properties. See System Properties for details regarding system properties configuration.

The *write_files* section allows to replace files on the system. By default, VeloCloud Orchestrator web services are configured with self-signed SSL certificate. If you would like to provide different SSL certificate, the above example replaces the server.crt and `server.key` files in the `/etc/nginx/velocloud/ssl/ folder` with user-supplied files.

Note: The server.key file must be unencrypted. Otherwise, the service will fail to start without the key password.

Create an 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 vco01-cidata.iso, is created with the following command on a Linux system:

```
genisoimage -output vco01-cidata.iso -volid cidata -joliet -rock user-data meta-data
```

Transfer the newly created ISO image to the datastore on the host running Arista.

3.2.2 Install on VMware

VMware vSphere provides a means of deploying and managing virtual machine resources. This section explains how to run the Orchestrator using the VMware vSphere Client.

Deploy OVA Template

Note: This procedure assumes familiarity with VMware vSphere and is not written with reference to any specific version of VMware vSphere.

1. Log in to the vSphere Client.
2. Select **File** > **Deploy OVF Template**.
3. Respond to the prompts with information specific to your deployment.

Table 1: OVF- Options and Descriptions

Option	Description
Source	Type a URL or navigate to the OVA package location.
OVF template details	Verify that you pointed to the correct OVA template for this installation.
Name and location	Name of the virtual machine.
Storage	Select the location to store the virtual machine files.
Provisioning	Select the provisioning type. "thin" is recommended for database and binary log volumes.
Network mapping	Select the network for each virtual machine to use.
	Important: Uncheck **Power On After Deployment**. Selecting it will start the virtual machine and it should be started later after the cloud-init ISO has been attached.

4. Select **Finish**.

> **Note:** Depending on your network speed, this deployment can take several minutes or more.

Attach ISO Image as a CD/DVD to Virtual Machine

1. Right-click the newly-added Orchestrator VM and select **Edit Settings**.
2. From the **Virtual Machine Properties** window, select **CD/DVD Drive**.
3. Select the **Use an ISO image** option.
4. Browse to find the ISO image you created earlier (we called ours vco01-cidata.iso), and then select it. The ISO can be found in the datastore that you uploaded it to, in the folder that you created.
5. Select **Connect on Power On**.
6. Select **OK** to exit the **Properties** screen.

Run the Orchestrator Virtual Machine

To start up the Orchestrator virtual machine:

1. Select to highlight it, then select the **Power On** button.
2. Select the **Console** tab to watch as the virtual machine boots up.

> **Note:** If you configured Orchestrator as described here, you should be able to log into the virtual machine with the user name vcadmin and password that you defined when you created the cloud-init ISO.

3.2.3 Install on KVM

This section discusses how to run the Orchestrator using the libvirt. This deployment was tested in Ubuntu 18.04 LTS.

Images

For KVM deployment, Arista provides the Orchestrator in four qcow images.

- ROOTFS
- STORE
- STORE2
- STORE3

The images are thin provisioned on deployment.

Start by copying the images to the KVM server. In addition, you must copy the cloud-init iso build as described in the previous section.

XML Sample

Note: For the images in the *images/vco* folder, you will need to edit from the XML.

```
<domain type='kvm' id='49'> <name>vco</name> <uuid>b0ff25bc-72b8-6ccb-e777-fdc0f4733e05</
uuid> <memory unit='KiB'>12388608</memory> <currentMemory unit='KiB'>12388608</currentMemory>
 <vcpu>2</vcpu> <resource> <partition>/machine</partition> </resource> <os> <type>hvm</type>
 </os> <features> <acpi/> <apic/> <pae/> </features> <cpu mode='custom' match='exact'> <model
 fallback='allow'>SandyBridge</model> <vendor>Intel</vendor> <feature policy='require' name='vme'/
> <feature policy='require' name='dtes64'/> <feature policy='require' name='invpcid'/> <feature
 policy='require' name='vmx'/> <feature policy='require' name='erms'/> <feature policy='requi
re' name='xtpr'/> <feature policy='require' name='smep'/> <feature policy='require' name='pbe'/
> <feature policy='require' name='est'/> <feature policy='require' name='monitor'/> <feature
 policy='require' name='smx'/> <feature policy='require' name='abm'/> <feature policy='requi
re' name='tm'/> <feature policy='require' name='acpi'/> <feature policy='require' name='fma'/
> <feature policy='require' name='osxsave'/> <feature policy='require' name='ht'/> <feature
 policy='require' name='dca'/> <feature policy='require' name='pdcm'/> <feature policy='requi
re' name='pdpe1gb'/> <feature policy='require' name='fsgsbase'/> <feature policy='require'
 name='f16c'/> <feature policy='require' name='ds'/> <feature policy='require' name='tm2'/
> <feature policy='require' name='avx2'/> <feature policy='require' name='ss'/> <feature
 policy='require' name='bmi1'/> <feature policy='require' name='bmi2'/> <feature policy='require'
 name='pcid'/> <feature policy='require' name='ds_cpl'/> <feature policy='require' name='movbe'/
> <feature policy='require' name='rdrand'/> </cpu> <clock offset='utc'/> <on_poweroff>destroy</
on_poweroff> <on_reboot>restart</on_reboot> <on_crash>restart</on_crash> <devices> <emulator>/us
r/bin/kvm-spice</emulator> <disk type='file' device='disk'> <driver name='qemu' type='qcow2'/>
 <source file='/images/vco/rootfs.qcow2'/> <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='disk'> <driver name='qemu' type='qcow2'/> <source file='/ images/vco/store.qcow2'/>
 <target dev='hdb' bus='ide'/> <alias name='ide0-0-1'/> <address type='drive' controller='0'
 bus='0' target='0' unit='1'/> </disk> <disk type='file' device='disk'> <driver name='qemu'
 type='qcow2'/> <source file='/ images/vco/store2.qcow2'/> <target dev='hdc' bus='ide'/> <alias
 name='ide0-0-2'/> <address type='drive' controller='0' bus='1' target='0' unit='0'/> </disk>
 <disk type='file' device='disk'> <driver name='qemu' type='qcow2' /> <source file='/images/vco/
store3.qcow2' /> <target dev='hdd' bus='ide' /> <alias name='ide0-0-3' /> <address type='drive'
 controller='0' bus='1' target='0' unit='1' /> </disk> <disk type='file' device='cdrom'> <driver
 name='qemu' type='raw'/> <source file='/ images/vco/seed.iso'/> <target dev='sdb' bus='sata'/>
 <readonly/> <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='direct'> <source dev='eth0' mode='vepa'/> </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='virtio'> <alias name='balloon0'/> <address
 type='pci' domain='0x0000' bus='0x00' slot='0x03' function='0x0'/> </memballoon> </devices>
 <seclabel type='none' /> <!-- <seclabel type='dynamic' model='apparmor' relabel='yes'/> --> </
domain>
```

Create the VM

To create the VM using the standard virsh commands:

```
virsh define vco.xml virsh start vco.xml
```

3.2.4 Install on AWS

This section discusses how to install VeloCloud Orchestrator on AWS.

Minimum Instance Requirements

See the first section of the Orchestrator Installation, titled Instance Requirements, and select an AWS instance type matching these requirements. Both CPU and Memory requirements must be satisfied. Example: use c4.2xlarge or larger; r4.2xlarge or larger

Request an AMI Image

Request an AMI ID from Arista. It will be shared with the customer account. Have an Amazon AWS account ID ready when requesting AMI access.

Installation

1. Launch the EC2 instance in AWS cloud.

   Example: http://docs.aws.amazon.com/efs/latest/ug/gs-step-one-create-ec2-resources.html
2. Configure the security group to allow inbound HTTP (TCP/80) as well as HTTPS (TCP/443).
3. After the instance is launched, point the web browser to the Operator login URL: `https://<name>/operator`.

3.3 Initial Configuration Tasks

Complete the following initial configuration tasks:

- Configure system properties
- Set up initial operator profile
- Set up operator accounts
- Create gateways
- Setup gateway pools
- Create customer account / partner account

3.3.1 Install an SSL Certificate

This section discusses how to install an SSL certificate.

To install an SSL certificate:

1. Login into the Orchestrator CLI console through SSH. If you configured the Orchestrator as described here, you should be able to log into the virtual machine with the user name *vcadmin* and password that you defined when you created the cloud-init ISO.
2. Generate the Orchestrator private key.

```
openssl genrsa -out server.key 2048
```

> **Note:** Do not encrypt the key. It must remain unencrypted on the Orchestrator system.

3. Generate a certificate request. Customize **`-subj`** according to your organization information.

```
openssl req -new -key server.key -out server.csr -subj "/C=US/ST=California/L=Mountain View/
O=Velocloud Networks Inc./OU=Development/CN=vco.velocloud.net"
```

Table 2: Subject- Options and Descriptions

Option	Description
C	country
ST	state
L	locality (city)
O	company
OU	department (optional)
CN	Orchestrator fully qualified domain name

4. Send `server.csr` to a Certificate Authority for signing. You should get back the SSL certificate (`server.crt`). Ensure that it is in the PEM format.
5. Install the certificate (which requires root access). Orchestrator SSL certificates are located in `/etc/nginx/velocloud/ssl/`.

```
cp server.key server.crt /etc/nginx/velocloud/ssl/ chmod 600 /etc/nginx/velocloud/ssl/s
erver.key
```

6. Restart `nginx`.

```
systemctl restart nginx
```

3.3.2 Configure System Properties

This section discusses how to configure System Properties, which provide a mechanism to control the system-wide behavior of the VeloCloud SD-WAN.

System Properties can be set initially using the cloud-init config file. For additional information, see Cloud-init Preparation. The following properties need to be configured to ensure proper operation of the service.

System Name

Enter a fully qualified Arista domain name in the `network.public.address` system property.

Google Maps

Google Maps is used for displaying edges and data centers on a map. Maps may fail to display without a license key. The Orchestrator will continue to function properly, but browser maps will not be available in this case.

1. Login into https://console.developers.google.com.
2. Create a new project, if one is not already created.
3. Locate the button **Enable API**. Select the **Google Maps APIs** and enable both **Google Maps JavaScript API** and **Google Maps Geolocation API**.
4. On the left side of the screen, click the **Credentials** link.
5. Under the Credentials page, click **Create Credentials**, then select **API key**. Create an API key.
6. Set the `service.client.googleMapsApi.key` system property to API key.
7. Set `service.client.googleMapsApi.enable` to "**true**."

Twilio

Twilio is a messaging service that allows you to receive alerts via SMS. It is optional. The account details can be entered into Arista through the Operator Portal's **System Properties** page. The properties are called:

- **`service.twilio.enable`** allows the service to be deactivated in the event that no Internet access is available to the Arista
- **`service.twilio.accountSid`**
- **`service.twilio.authToken`**
- **`service.twilio.phoneNumber`** in (nnn)nnn-nnnn format

Obtain the service at https://www.twilio.com.

MaxMind

MaxMind is a geolocations service. It is used to automatically detect Edge and Gateway locations and ISP names based on an IP address. If this service is deactivated, then geolocation information will need to be updated manually. The account details can be entered into the Arista through the Operator Portal's **System Properties page**. You can configure:

- **`service.maxmind.enable`** allows the service to be deactivated in the event that no Internet access is available to the Arista
- **`service.maxmind.userid`** holds the user identification supplied by MaxMind during the account creation
- **`service.maxmind.license`** holds the license key supplied by MaxMind

Obtain the license at: https://www.maxmind.com/en/geoip-api-web-services.

Email

Email services can be used for both sending the Edge activation messages as well as for alarms and notifications. It is not required, but it is strongly recommended that you configure this as part of Arista operations. The following system properties are available to configure the external email service used by the Orchestrator:

- **`mail.smtp.auth.pass`**- SMTP user password.
- **`mail.smtp.auth.user`**- SMTP user for authentication.
- **`mail.smtp.host`**- relay server for email originated from Arista.
- **`mail.smtp.port`**- SMTP port.
- **`mail.smtp.secureConnection`**- use SSL for SMTP traffic.

3.4 Upgrade VeloCloud Orchestrator

This section discusses how to upgrade the VeloCloud Orchestrator.

To upgrade the Orchestrator:

1. Upload the image to the Orchestrator system using any file transfer tool available in your infrastructure, for example "*scp*." Copy the image to the following location on the system: `/var/lib/velocloud/software_update/vco_update.tar`.
2. Connect to the SD-WAN Orchestrator console and run:

```
sudo /opt/vc/bin/vco_software_update
```

Note: If you configured the Orchestrator as described here, you should be able to log into the virtual machine with the user name vcadmin and the password that you defined when you created your the cloud-init configuration files.

For instructions on how to upgrade the SD-WAN Orchestrator with DR deployment, see the topic Upgrade an Orchestrator.

3.5 Expand Disk Size

All storage volumes are configured as LVM devices. They can be resized online by providing the underlying virtualization technology to support online disk expansion. Disks are expanded automatically via cloud-init when the VM boots.

To expand disks after boot:

1. Login into the VeloCloud Orchestrator system console.
2. Identify the physical disks that support the database volume.

```
vgs -o +devices store
```

Example:

```
root@vco:~# vgs -o +devices db_data \ VG #PV #LV #SN Attr VSize VFree Devices store 1 1 0 wz--
n- 500.00g 125.00g /dev/sdb(0)
```

3. Identify the physical disk attachment.

```
lshw -class volume
```

Example:

```
/dev/sdb is attached to scsi@2:0.1.0 (Host: scsi2 Channel: 00 Id: 01 Lun: 00)
```

```
root@vco:~# lshw -class volume *-volume description: EXT4 volume vendor: Linux physical id:
 1 bus info: scsi@2:0.0.0,1 logical name: /dev/sda1 logical name: / version: 1.0 serial:
 9d212247-77c4-4f98-a5c2-7f8470fa2da8 size: 10239MiB capacity: 10239MiB capabilities: primary
 bootable journaled extended_attributes large_files huge_files dir_nlink recover extents ext4
 ext2 initialized configuration: created=2016-02-22 20:49:38 filesystem=ext4 label=cloudim
g-rootfs lastmountpoint=/ modified=2016-02-22 21:18:58 mount.fstype=ext4 mount.options
=rw,relatime,data=ordered mounted=2016-10-06 23:22:04 state=mounted *-disk:1 description:
 SCSI Disk physical id: 0.1.0 bus info: scsi@2:0.1.0 logical name: /dev/sdb serial: v5V2zm-
Lvbh-Mfx3-W8ki-COI9-DAtP-RXndhu size: 500GiB capacity: 500GiB capabilities: lvm2 configuration
: sectorsize=512 *-disk:2 description: SCSI Disk physical id: 0.2.0 bus info: scsi@2:0.2.0
 logical name: /dev/sdc serial: fTQFJ2-giAV-WsXL-1Wha-V305-oQkV-qqS3SA size: 100GiB capacity:
 100GiB capabilities: lvm2 configuration: sectorsize=512
```

4. On the hypervisor host, locate the disk attached to the VM using bus information. Example: SCSI(0:1)
5. Extend the virtual disk. For instructions, see the KB article *Increasing the disk size on a Virtual Machine.*
6. View the disk input/output statistics. These statistics are displayed twice, at an interval of 10 seconds.

```
sar -d -p 10 2
```

> **Note:** This step is optional.

7. View detailed device utilization statistics, that provides insights into individual storage device performance.

```
iostat -d -x
```

> **Note:** This step is optional.

8. Re-login into the VeloCloud Orchestrator system console.
9. Re-scan the block device for the resized physical volume.

Example:

```
echo 1 > /sys/block/$DEVICE/device/rescan
```

Example:

```
echo 1 > /sys/block/sdb/device/rescan
```

10. Resize the LVM physical disk.

```
pvresize /dev/sdb
```

11. Determine the amount of free space in the database volume group.

```
vgdisplay store |grep Free
```

Example:

```
root@vco:~# vgdisplay store |grep Free Free PE / Size 34560 / 135.00 GiB
```

12. Extend the database logical volume.

```
lvextend -r -L+#G /dev/store/data
```

Example:

```
root@vco1:~# lvextend -r -L+1G /dev/store/data Size of logical volume store/data changed
 from 400.00 GiB (102400 extents) to 401.00 GiB (102656 extents). Logical volume store/data
 successfully resized. resize2fs 1.44.1 (24-Mar-2018) Filesystem at /dev/mapper/store-data is
 mounted on /store; on-line resizing required old_desc_blocks = 50, new_desc_blocks = 51 The
 filesystem on /dev/mapper/store-data is now 105119744 (4k) blocks long.
```

13. View the new size of the volume.

```
df -h /dev/store/data
```

Example:

```
root@vco:~# df -h /dev/store/data Filesystem Size Used Avail Use% Mounted on /dev/mapper/store-
data 379G 1.2G 359G 1% /store
```

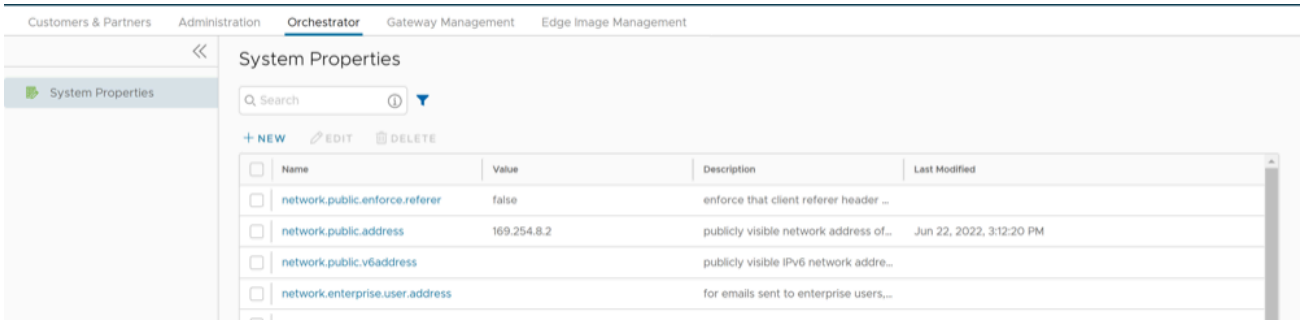
Chapter 4

System Properties

Arista provides System Properties to configure various features and options available in the Orchestrator portal.

In the **Operator** portal, navigate to the **System Properties** page, which lists the available pre-defined system properties. See List of System Properties, which lists some of the system properties that you can modify as an Operator.

Figure 4-1: System Properties



To configure the system properties:

1. Select **New System Property** to add a new property.

2. In the **New System Property** window, configure the following parameters:

Figure 4-2: New System Property

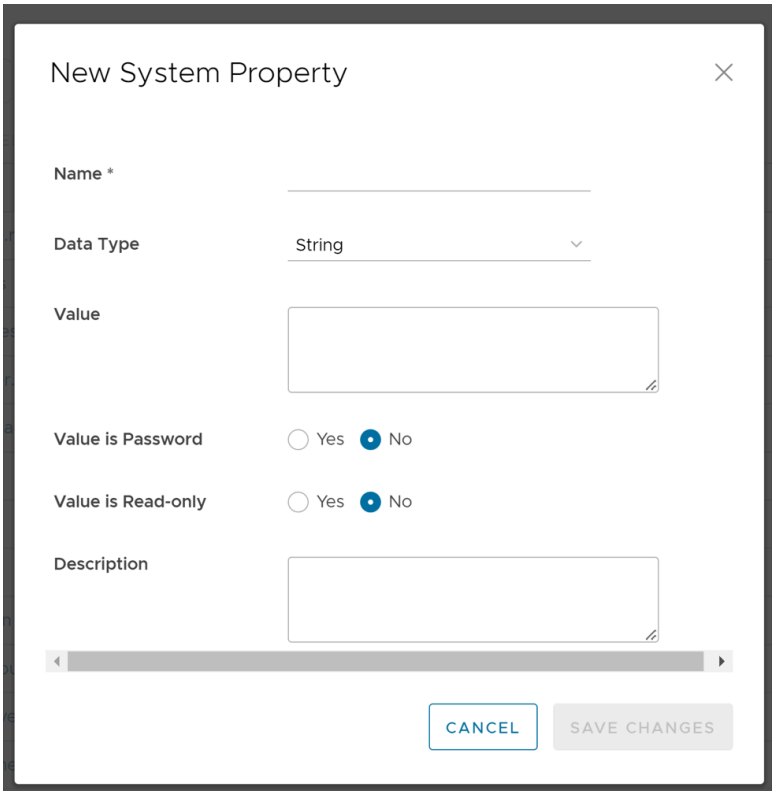


Table 3: New System Property- Options and Descriptions

Option	Description
Name	Enter the **Name** for the new system property.
Data Type	Choose the required **Data Type** from the drop-down menu.
Value	Enter the **Value** for the property according to the data type.
Value is Password	Select **Yes** or **No** as required.
Value is Read-only	Select **Yes** or **No** for as required.
Description	Enter the **Description** for the new system property

3. Select **Save Changes**.
4. You can use the **Search** field to find a specific system property.

   See the section List of System Properties, which lists some of the system properties that you can modify as an Operator.

> **Note:** It is recommended to contact Arista Support before making changes to the system properties.

4.1 List of System Properties

As an Operator, you can add or modify the values of the system properties.

The following tables describe some of the system properties. As an Operator, you can set the values for these properties.

- Alert Emails
- Alerts
- Bastion Orchestrator Configuration
- Certificate Authority
- Customer Configuration
- Data Retention
- Edges
- Edge Activation
- Edge Management
- Enhanced Firewall Services
- LAN-Side NAT Rules
- Monitoring
- Notifications
- Password Reset and Lockout
- Rate Limiting APIs
- Remote Diagnostics
- Security Service Edge
- Segmentation
- Self-service Password Reset
- Syslog Forwarding
- TACACS Services
- Two-factor Authentication
- Tunnel Parameters for Edges
- VNF Configuration
- VPN
- Warning Banner
- Zscaler

Table 4: Alert Emails

System Property	Description
vco.alert.mail.to	When an alert is triggered, a notification is sent immediately to the list of Email addresses provided in the Value field of this system property. You can enter multiple Email IDs separated by commas. If the property does not contain any value, then the notification is not sent. The notification is meant to alert Arista Support/Operations personnel of impending issues before notifying the customer.
vco.alert.mail.cc	When alert emails are sent to any customer, a copy is sent to the Email addresses provided in the Value field of this system property. You can enter multiple Email IDs separated by commas.
mail.*	There are multiple system properties available to control the Alert Emails. You can define the Email parameters like SMTP properties, username, password, and so on.

Table 5: Alerts

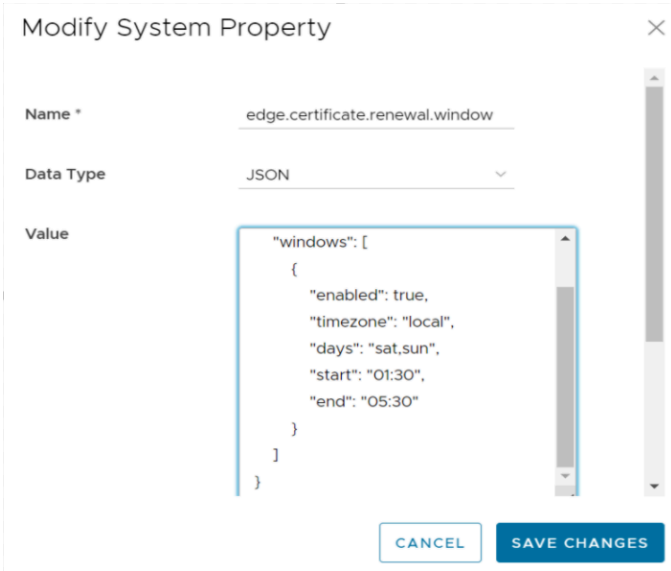
System Property	Description
vco.alert.enable	Globally activates or deactivates the generation of alerts for both Operators and Enterprise customers.
vco.enterprise.alert.enable	Globally activates or deactivates the generation of alerts for Enterprise customers.
vco.operator.alert.enable	Globally activates or deactivates the generation of alerts for Operators.

Table 6: Bastion Orchestrator Configuration

System Property	Description
session.options.enableBastionOrchestrator	Enables the Bastion Orchestrator feature. For additional information, see *Bastion Orchestrator Configuration Guide*.
vco.bastion.private.enable	Enables the Orchestrator to be the Private Orchestrator of the Bastion pair.
vco.bastion.public.enable	Enables the Orchestrator to be the Public Orchestrator of the Bastion pair.

Table 7: Edge Certificate

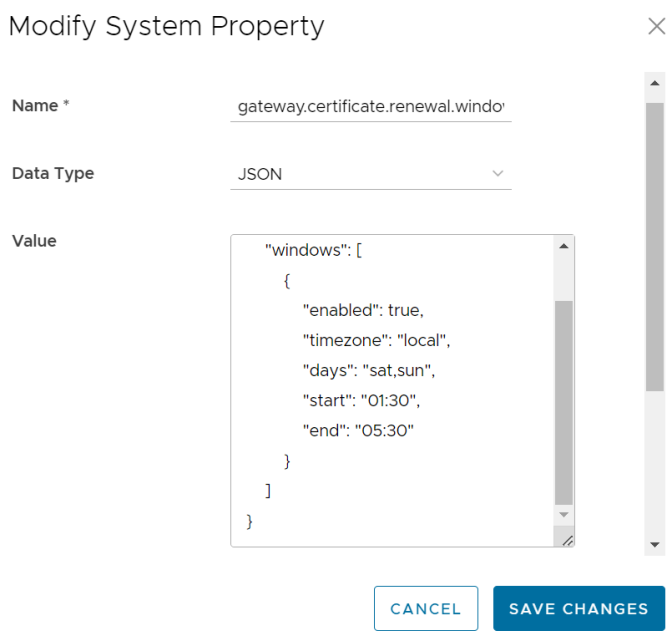
System Property	Description

edge.certificate.renewal.window	This optional system property allows the Operator to define one or more maintenance windows during which the Edge certificate renewal is enabled. Certificates scheduled for renewal outside of the windows will be deferred until the current time falls within one of the enabled windows.
	
	Enable System Property:
	
	To enable this system property, type "true" for "enabled" in the first part of the Value text area in the Modify System Property dialog box. An example of the first part of this system property when it is enabled is shown below.
	
	Operators can define multiple windows to restrict the days and hours of the day during which Edge renewals are enabled. Each window can be defined by a day, or a list of days (separated by a comma), and a start and end time. Start and end times can be specified relative to an Edge's local time zone, or relative to UTC. See image below for an example.
	
	Figure 4-3: Modify System Property
	
	 
	
	Note: If attributes are not present, the default is enabled "false."
	
	When defining window attributes, adhere to the following:
	
	• Use IANA time zones, not PDT or PST (e.g. America/Los_Angeles) See https://en.wikipedia.org/wiki/List_of_tz_database_time_zones for additional information.
	
	• Use UTC for days (e.g. SAT, SUN).
	  • Separated by comma.
	  • Days in three letters in English.
	  • Not case sensitive.
	
	• Use Military 24 hour time format only (HH:MM) for start times (e.g. 01:30) and end times (e.g. 05:30).

System Property	Description

	If the above-mentioned values are missing, the attribute defaults in each window definition are as follow:
	• If enabled is missing, the default value = false.
	• If timezone is missing, the default = 'local.'
	• If one of either 'days' or end and start times are missing, the defaults are as follows:
	  • If 'days' is missing, the start/end is applied to each day of the week (Mon, Tue, Wed, Thur, Fri, Sat, Sun).
	  • If end and start times are missing, then any time in the specified day will match (start = 00:00 and end = 23:59).
	Note: One of either 'days' or end and start times must be present. However, if they are missing, the defaults will be as indicated above.
	
	Deactivate System Property:
	
	This system property is deactivated by default, which means the certificate will automatically renew after it expires. "Enabled" will be set to "false in the first part of the Value text area in the Modify System Property dialog box. An example of this property when it is deactivated is shown below.
	`{ "enabled": false, "windows": [ {`
	
	Note: This system property requires that PKI be enabled.

System Property	Description

gateway.certificate.renewal.window	This optional system property allows the Operator to define one or more maintenance windows during which the Gateway certificate renewal is enabled. Certificates scheduled for renewal outside of the windows will be deferred until the current time falls within one of the enabled windows.
	
	Enable System Property:
	
	To enable this system property, type "true" for "enabled" in the first part of the Value text area in the Modify System Property dialog box. See image below for an example.
	
	Operators can define multiple windows to restrict the days and hours of the day during which edge renewals are enabled. Each window can be defined by a day, or list of days (separated by a comma), and a start and end time. Start and end times can be specified relative to an edge's local timezone, or relative to UTC. See image below for an example.
	
	Figure 4-4: Modify System Property
	
	 
	
	Note: If attributes are not present, the default is enabled "false."
	
	When defining window attributes, adhere to the following:
	• Use IANA time zones, not PDT or PST (e.g. America/Los_Angeles) See https://en.wikipedia.org/wiki/List_of_tz_database_time_zones for additional information.
	• Use UTC for days (e.g. SAT, SUN).
	  • Separated by comma.
	  • Days in three letters in English.
	  • Not case sensitive.
	• Use Military 24 hour time format only (HH:MM) for start times (e.g. 01:30) and end times (e.g. 05:30).

System Property	Description

	If the above-mentioned values are missing, the attribute defaults in each window definition are as follow:
	• If enabled is missing, the default value = false.
	• If timezone is missing, the default = 'local."
	• If one of either 'days' or end and start times are missing, the defaults are as follows:
	  • If 'days' is missing, the start/end is applied to each day of the week (Mon, Tue, Wed, Thur, Fri, Sat, Sun).
	  • If end and start times are missing, then any time in the specified day will match (start = 00:00 and end = 23:59).
	Note: One of either 'days' or (end and start) must be present. However, if they are missing, the defaults will be as indicated above.
	
	Deactivate System Property:
	
	This system property is deactivated by default, which means the certificate will automatically renew after it expires. "Enabled" will be set to "false in the first part of the Value text area in the Modify System Property dialog box. An example of this property when it is deactivated is shown below.
	
	`{ "enabled": false, "windows": [ {`
	
	Note: This system property requires that PKI be enabled.

Table 8: Customer Configuration

System Property	Description
session.options.enableServiceLicenses	This system property allows Operator users to manage Service Configuration under **Global Settings** > **Customer Configuration**, and is set to **True**, by default.

Table 9: Data Retention

System Property	Description
retention.highResFlows.days	This system property enables Operators to configure high resolution flow stats data retention anywhere between 1 and 90 days.
retention.lowResFlows.months	This system property enables Operators to configure low resolution flow stats data retention anywhere between 1 and 365 days.
session.options.maxFlowstatsRetentionDays	This property enables Operators to query more than two weeks of flows stats data.
retentionWeeks.enterpriseEvents	Enterprise events retention period (-1 sets retention to the maximum time period allowed)
retentionWeeks.operatorEvents	Operator events retention period (-1 sets retention to the maximum time period allowed)
retentionWeeks.proxyEvents	Proxy events retention period (-1 sets retention to the maximum time period allowed)
retentionWeeks.firewallLogs	Firewall logs retention period (-1 sets retention to the maximum time period allowed)
retention.linkstats.days	Link stats retention period (-1 sets retention to the maximum time period allowed)
retention.linkquality.days	Link quality events retention period (-1 sets retention to the maximum time period allowed)
retention.healthstats.days	Edge health stats retention period (-1 sets retention to the maximum time period allowed)
retention.pathstats.days	Path stats retention period (-1 sets retention to the maximum time period allowed)

Table 10: Edges

SD-WAN Data	Date Retention Period
Enterprise Events	1 year
Enterprise Alerts	1 year
Operator Events	1 year
Enterprise Proxy Events	1 year
Link Stats	1 year
Link QoE	1 year
Path Stats	2 weeks
Flow Stats (Low Resolution)	1 year – 1 hour rollup
Flow Stats (High Resolution)	2 weeks – 5 minute rollup
Edge Health Stats	1 year

Table 11: Edge Activation

System Property	Description
edge.offline.limit.sec	If the Orchestrator does not detect a heartbeat from an Edge for the specified duration, then the state of the Edge is moved to OFFLINE mode.
edge.link.unstable.limit.sec	When the Orchestrator does not receive link statistics for a link for the specified duration, the link is moved to UNSTABLE mode.
edge.link.disconnected.limit.sec	When the Orchestrator does not receive link statistics for a link for the specified duration, the link is disconnected.
edge.deadbeat.limit.days	If an Edge is not active for the specified number of days, then the Edge is not considered for generating Alerts.
vco.operator.alert.edgeLinkEvent.enable	Globally activates or deactivates Operator Alerts for Edge Link events.
vco.operator.alert.edgeLiveness.enable	Globally activates or deactivates Operator Alerts for Edge Liveness events.

Table 12: Edge Management

System Property	Description
edge.activation.key.encode.enable	Base64 encodes the activation URL parameters to obscure values when the Edge Activation Email is sent to the Site Contact.
edge.activation.trustedIssuerReset.enable	Resets the trusted certificate issuer list of the Edge to contain only the Orchestrator Certificate Authority. All TLS traffic from the edge are restricted by the new issuer list.
network.public.certificate.issuer	Set the value of `network.public.certificate.issuer` equal to the PEM encoding of the issuer of Orchestrator server certificate, when `edge.activation.trustedIssuerReset.enable` is set to True. This will add the server certificate issuer to the trusted issuer of the Edge, in addition to the Orchestrator Certificate Authority.

Table 13: Edge Link Down Limit

System Property	Description
edge.link.show.limit.sec	Allows to set the **Edge Link Down Limit** value for each Edge.

Table 14: NTICS

System Property	Description
ntics.public address	Specifies the hostname that is used to access the NSX Threat Intelligent Cloud Service (NTICS).
gsm.public.address	Specifies the Public address of Global Services Manager (GSM).
gsm.authentication.key	Specifies the mTLS key to authenticate with GSM.
gsm.authentication.cert	Specifies the mTLS certificate to authenticate with GSM.
gsm.authentication.passphrase	Specifies the mTLS passphrase to authenticate with GSM.

Table 15: LAN-Side NAT Rules

System Property	Description
session.options.enableLansidePortRules	Allows to configure the parameters **Inside Port** and **Outside Port** under **Device Settings tab** > **Routing and NAT** > **LAN-Side NAT Rules** for an Edge or Profile.

Table 16: Monitoring

System Property	Description
vco.monitor.enable	Globally activates or deactivates monitoring of Enterprise and Operator entity states. Setting the Value to **False** prevents the Orchestrator from changing entity states and triggering alerts.
vco.enterprise.monitor.enable	Globally activates or deactivates monitoring of Enterprise entity states.
vco.operator.monitor.enable	Globally activates or deactivates monitoring of Operator entity states.

Table 17: Live Data

System Property	Description
edge.liveData.enterFlowLiveMode.delay.seconds	How long the Edge will wait before giving up on capturing the count configured by edge.liveData.enterFlowLiveMode.delay.seconds. The default value is five seconds. The allowed range is 5 - 59 seconds. The invalid input defaults to zero seconds.
edge.liveData.enterFlowLiveMode.flow.count	How many flows the Edge will return if met within the configured time controlled by edge.liveData.enterFlowLiveMode.flow.count.The default value is 1000. The allowed range is 1000 - 4999 total flows. The invalid input defaults to one flow.

Table 18: Alerts and Notifications

System Property	Description
vco.notification.enable	Globally activates or deactivates the delivery of Alert notifications to both Operator and Enterprises.
vco.enterprise.notification.enable	Globally activates or deactivates the delivery of Alert notifications to the Enterprises.
vco.operator.notification.enable	Globally activates or deactivates the delivery of Alert notifications to the Operator.

Table 19: Object Groups

System Property	Description
vco.object.groups.max.count.per.enterprise	Maximum allowed number of object groups per Enterprise. The default value is 2000.
vco.object.groups.max.count.per.edge	Maximum allowed number of object group associations per Edge and its Profile. The default value is 1000.

Table 20: Password

System Property	Description
vco.enterprise.resetPassword.token.expirySeconds	Duration of time, after which the password reset link for an enterprise user expires.
vco.enterprise.authentication.passwordPolicy	Defines the password strength, history, and expiration policy for customer users. Edit the JSON template in the Value field to define the following: **strength** • **minlength**: Minimum password character length. The default minimum password length is 8 characters. • **maxlength**: Maximum password character length. The default maximum password length is 32 characters. • **requireNumber**: The password must contain at least one numeric character. Numeric requirement is enabled by default. • **requireLower**: The password must contain at least one lowercase character. Lowercase requirement is enabled by default. • **requireUpper**: The password must contain at least one uppercase character. Uppercase requirement is not enabled by default. • **requireSpecial**: The password must contain at least one special character (for example, _@!). The special character requirement is not enabled by default. • **excludeTop**: Password must not match a list of the most used passwords. Default value is 1000, representing the top 1000 most used passwords, and is configurable to a maximum of 10,000 of the most used passwords. • **maxRepeatingCharacters**: Password must not include a configurable number of repeated characters. For example, if maxRepeatingCharacters is set to ‘2’ then the Orchestrator would reject any password with 3 or more repetitive characters, like “Passwordaaa”. The default value of -1 signifies that this feature is not enabled. • **maxSequenceCharacters**: Password must not include a configurable number of sequential characters. For example, if maxSequenceCharacters is set to ‘3’ then the Orchestrator would reject any password where 4 or more characters which are sequential, like “Password1234”. The default value of -1 signifies that this feature is not enabled.

System Property	Description

	• **disallowUsernameCharacters**: Password must not match a configurable portion of the user's ID. For example, if disallowUsernameCharacters is set to 5, if a user with username username@domain.com attempts to configure a new password that includes ‘usern’ or ‘serna’, or any five-character string that matches a section of the user’s username, that new password would be rejected by the Orchestrator. The default value of -1 signifies that this feature is not enabled.
	• **variationValidationCharacters**: New password must vary from the old password by a configurable number of characters. The Orchestrator uses the Levenshtein distance between two words to determine the variation between the new and old password. The Levenshtein distance is the minimum number of single-character edits (insertions, deletions, or substitutions) required to change one word into another.
	• If variationValidationCharacters is set to 4, then the Levenshtein distance between the new and old password must be 4 or greater. In other words, the new password must have 4 or more variations from the old password. For example, if the old password used was "kitten" and the new password is "sitting", the Levenshtein distance for these is 3, since it requires only three edits to change kitten into sitting:
	  • kitten → sitten (substitution of "s" for "k")
	  • sitten → sittin (substitution of "i" for "e")
	  • sittin → sitting (insertion of "g" at the end).
	Since the new password only varies by 3 characters from the old, “sitting” would be rejected as a new password to replace “kitten”. The default value of -1 signifies that this feature is not enabled.
	expiry:
	• **enable**: Set this to **true** to enable automatic expiry of customer user passwords.
	• **days**: Enter the number of days that an customer password may be used before forced expiration.
	history:
	• **enable**: Set this to **true** to enable recording of customer users' previous Passwords.
	• **count**: Enter the number of previous Passwords to be saved in the history. When a customer user tries to change the password, the system does not allow the user to enter a password that is already saved in the history.

enterprise.user.lockout.defaultAttempts	Number of times the enterprise user can attempt to login. If the login fails for the specified number of times, the account is locked.
enterprise.user.lockout.defaultDurationSeconds	Duration of time, in seconds, in which the Enterprise user account is locked. For example, if set to 300, the Enterprise user account will get locked if four incorrect login attempts are made within 300 seconds. If set to 60, the Enterprise user account will get locked if four incorrect attempts are made within one minute.  **Note:** The number of attempts is configurable via the `enterprise.user.lockout.defaultAttempts` system property.

System Property	Description
enterprise.user.lockout.enabled	Activates or deactivates the lockout option for the enterprise login failures.
vco.operator.resetPassword.token.expirySeconds	Duration of time, after which the password reset link for an Operator user expires.
vco.operator.authentication.passwordPolicy	Defines the password strength, history, and expiration policy for Operator users. Edit the JSON template in the Value field to define the following: **strength** • **minlength**: Minimum password character length. The default minimum password length is 8 characters. • **maxlength**: Maximum password character length. The default maximum password length is 32 characters. • **requireNumber**: The password must contain at least one numeric character. Numeric requirement is enabled by default. • **requireLower**: The password must contain at least one lowercase character. Lowercase requirement is enabled by default. • **requireUpper**: The password must contain at least one uppercase character. Uppercase requirement is not enabled by default. • **requireSpecial**: The password must contain at least one special character (for example, _@!). The special character requirement is not enabled by default. **Note:** Starting from the 4.5 release, the use of the special character "<" in the password is no longer supported. In cases where users have already used "<" in their passwords in previous releases, they must remove it to save any changes on the page.

System Property	Description

	• **excludeTop**: Password must not match a list of the most used passwords. Default value is 1000, representing the top 1000 most used passwords, and is configurable to a maximum of 10,000 of the most used passwords.
	
	• **maxRepeatingCharacters**: Password must not include a configurable number of repeated characters. For example, if maxRepeatingCharacters is set to ‘2’ then the Orchestrator would reject any password with 3 or more repetitive characters, like “Passwordaaa”. The default value of -1 signifies that this feature is not enabled.
	
	• **maxSequenceCharacters** : Password must not include a configurable number of sequential characters. For example, if maxSequenceCharacters is set to ‘3’ then the Orchestrator would reject any password where 4 or more characters which are sequential, like “Password1234”. The default value of -1 signifies that this feature is not enabled.
	
	• **disallowUsernameCharacters**: Password must not match a configurable portion of the user's ID. For example, if disallowUsernameCharacters is set to 5, if a user with username username@domain.com attempts to configure a new password that includes ‘usern’ or ‘serna’, or any five-character string that matches a section of the user’s username, that new password would be rejected by the Orchestrator. The default value of -1 signifies that this feature is not enabled.
	
	• **variationValidationCharacters**: New password must vary from the old password by a configurable number of characters. The Orchestrator uses the Levenshtein distance between two words to determine the variation between the new and old password. The Levenshtein distance is the minimum number of single-character edits (insertions, deletions, or substitutions) required to change one word into another.
	
	• If variationValidationCharacters is set to 4, then the Levenshtein distance between the new and old password must be 4 or greater. In other words, the new password must have 4 or more variations from the old password. For example, if the old password used was "kitten" and the new password is "sitting", the Levenshtein distance for these is 3, since it requires only three edits to change kitten into sitting:
	  • kitten → sitten (substitution of "s" for "k")
	  • sitten → sittin (substitution of "i" for "e")
	  • sittin → sitting (insertion of "g" at the end).
	
	Since the new password only varies by 3 characters from the old, “sitting” would be rejected as a new password to replace “kitten”. The default value of -1 signifies that this feature is not enabled.
	
	expiry:
	
	• **enable**: Set this to **true** to enable automatic expiry of Operator user passwords.
	
	• **days**: Enter the number of days that an Operator password may be used before forced expiration.
	
	history:
	
	• **enable**: Set this to **true** to enable recording of Operator users' previous Passwords.
	
	• **count**: Enter the number of previous Passwords to be saved in the history. When a Operator user tries to change the password, the system does not allow the user to enter a password that is already saved in the history.

System Property	Description
operator.user.lockout.defaultAttempts	Number of times the Operator user can attempt to login. If the login fails for the specified number of times, the account is locked.
operator.user.lockout.defaultDurationSeconds	Duration of time, in seconds, in which an Operator user account is locked. For example, if set to 300, the Operator user account will get locked if four incorrect login attempts are made within 300 seconds. If set to 60, the Operator user account will get locked if four incorrect attempts are made within one minute.  **Note:** The number of attempts is configurable via the `operator.user.lockout.defaultAttempts` system property.
operator.user.lockout.enabled	Activates or deactivates the lockout option for the Operator login failures.

Table 21: API

System Property	Description
vco.api.rateLimit.enabled	Allows Operator Super users activate or deactivate the rate limiting feature at the system level. By default, the value is **False**.  **Note:** The rate-limiter is not enabled in earnest, that is, it will not reject API requests that exceed the configured limits, unless the `vco.api.rateLimit.mode.logOnly` setting is deactivated.
vco.api.rateLimit.mode.logOnly	Allows Operator Super user to use rate limit in a **LOG_ONLY** mode. When the value is set as **True** and if a rate limit exceeds, this option logs only the error and fires respective metrics allowing clients to make requests without rate limiting. When the value is set to **False**, the request API is restricted with defined policies and HTTP 429 is returned.

System Property	Description

vco.api.rateLimit.rules.global	Allows to define a set of globally applicable policies used by the rate-limiter, in a JSON array. By default, the value is an empty array.
	
	Each type of user (Operator, Partner, and Customer) can make up to 500 requests for every 5 seconds. The number of requests is subject to change based on the behavior pattern of the rate limited requests.
	
	The JSON array consists of the following parameters:
	
	Types: The type objects represent different contexts in which the rate limits are applied. The following are the different type objects that are available:
	• SYSTEM: Specifies a global limit shared by all the users.
	• OPERATOR_USER: A limit that can be set in general for all the Operator users.
	• ENTERPRISE_USER: A limit that can be set in general for all the Enterprise users.
	• MSP_USER: A limit that can be set in general for all the MSP users.
	• ENTERPRISE: A limit that can be shared between all users of an Enterprise and is applicable to all the Enterprises in the network.
	• PROXY: A limit that can be shared between all users of a Proxy and is applicable to all proxies.
	
	Policies: Add rules to the policies to apply the requests that match the rule, by configuring the following parameters:
	• Match: Enter the type of requests to be matched:
	  • All: Rate-limit all requests matching one of the type objects.
	  • METHOD: Rate-limit all requests matching the specified method name.
	  • METHOD_PREFIX: Rate-limit all requests matching the specified method group.
	• Rules: Enter the values for the following parameters:
	  • maxConcurrent: Number of jobs that can be performed at the same time.
	  • reservoir: Number of jobs that can be performed before the limiter stops performing jobs.
	  • reservoirRefreshAmount: Value to set the reservoir to when reservoirRefreshInterval is in use.
	  • reservoirRefreshInterval: For every millisecond of reservoirRefreshInterval, the reservoir value will be automatically updated to the value of reservoirRefreshAmount. The reservoirRefreshInterval value should be a multiple of 250 (5000 for Clustering).

System Property	Description
	Enabled: Each type limit can be activated or deactivated by including the enabled key in APIRateLimiterTypeObject. By default, the value of enabled is True, even if the key is not included. You need to include "enabled": false key to deactivate the individual type limits. The following example shows a sample JSON file with default values: `[{ "type": "OPERATOR_USER", "policies": [{ "match": { "type": "ALL" }, "rules": { "reservoir": 500, "reservoirRefreshAmount": 500, "reservoirRefreshInterval": 5000 } }] }, { "type": "MSP_USER", "policies": [{ "match": { "type": "ALL" }, "rules": { "reservoir": 500, "reservoirRefreshAmount": 500, "reservoirRefreshInterval": 5000 } }] }, { "type": "ENTERPRISE_USER", "policies": [{ "match": { "type": "ALL" }, "rules": { "reservoir": 500, "reservoirRefreshAmount": 500, "reservoirRefreshInterval": 5000 } }] }]` **Note:** It is recommended not to change the default values of the configuration parameters.
vco.api.rateLimit.rules.enterprise.default	Comprises the default set of Enterprise-specific policies applied to newly created Customers. The Customer-specific properties are stored in the Enterprise property **vco.api.rateLimit.rules.enterprise**.
vco.api.rateLimit.rules.enterpriseProxy.default	Comprises the default set of Enterprise-specific policies applied to newly created Partners. The Partner-specific properties are stored in the Enterprise proxy property **vco.api.rateLimit.rules.enterpriseProxy**.

For additional information on Rate limiting, see Rate Limiting API Requests.

Table 22: DNS

System Property	Description
network.public.address	Specifies the browser origin address/DNS hostname that is used to access the Orchestrator UI.
network.portal.websocket.address	Allows to set an alternate DNS hostname/address to access the Orchestrator UI from a browser, if the browser address is not the same as the value of network.public.address system property. As remote diagnostics now uses a WebSocket connection, to ensure web security, the browser origin address that is used to access the Orchestrator UI is validated for incoming requests. In most cases, this address is same as the network.public.address system property. In rare scenarios, the Orchestrator UI can be accessed using another DNS hostname/address that is different from the value set in the network.public.address system property. In such cases, you can set this system property to the alternate DNS hostname/address. By default, this value is not set.
session.options.websocket.portal.idle.timeout	Allows to set the total amount of time (in seconds) the browser WebSocket connection is active in an idle state. By default, the browser WebSocket connection is active for 300 seconds in an idle state.

Table 23: Segments

System Property	Description
enterprise.capability.enableSegmentation	Activates or deactivates the segmentation capability for Enterprise users.
enterprise.segments.system.maximum	Specifies the maximum number of segments allowed for any Enterprise user. Ensure that you change the value of this system property to 128 if you want to enable 128 segments on Orchestrator for an Enterprise user.
enterprise.segments.maximum	Specifies the default value for the maximum number of segments allowed for a new or existing Enterprise user. The default value for any Enterprise user is 16.  **Note:** This value must be less than or equal to the number defined in the system property, enterprise.segments.system.maximum. It is not recommended for you to change the value of this system property if you want to enable 128 segments for an Enterprise user. Instead, you can enable **Customer Capabilities** in the **Customer Configuration** page to configure the required number of segments. For instructions, refer to the *Configure Customer Capabilities* section in the *Arista VeloCloud SD-WAN Operator Guide*.
enterprise.subinterfaces.maximum	Specifies the maximum number of sub-interfaces that can be configured for an Enterprise user. The default value is 32.
enterprise.vlans.maximum	Specifies the maximum number of VLANs that can be configured for an Enterprise user. The default value is 32.
session.options.enableAsyncAPI	When the segment scale is increased to 128 segments for any Enterprise user, to prevent UI timeouts, you can enable Async APIs support on the UI by using this system property. The default value is true.
session.options.asyncPollingMilliSeconds	Specifies the Polling interval for Async APIs on the UI. The default value is 5000 milliseconds.
session.options.asyncPollingMaxCount	Specifies the maximum number of calls to get Status API from the UI. The default value is 10.
vco.enterprise.events.configuration.diff.enable	Activates or deactivates configuration diff event logging. Whenever the number of segments for an Enterprise user is greater than 4, the configuration diff event logging will be deactivated. You can enable configuration diff event logging using this system property.

Table 24: Two-factor Authentication

System Property	Description
vco.enterprise.resetPassword.twoFactor.mode	Defines the mode for the second level for password reset authentication, for all the Enterprise users. Currently, only the SMS mode is supported.
vco.enterprise.resetPassword.twoFactor.required	Activates or deactivates the two-factor authentication for password reset of Enterprise users.
vco.enterprise.selfResetPassword.enabled	Activates or deactivates self-service password reset for Enterprise users.
vco.enterprise.selfResetPassword.token.expirySeconds	Duration of time, after which the self-service password reset link for an Enterprise user expires.
vco.operator.resetPassword.twoFactor.required	Activates or deactivates the two-factor authentication for password reset of Operator users.
vco.operator.selfResetPassword.enabled	Activates or deactivates self-service password reset for Operator users.
vco.operator.selfResetPassword.token.expirySeconds	Duration of time, after which the self-service password reset link for an Operator user expires.

Table 25: Syslog

System Property	Description
log.syslog.backend	Backend service syslog integration configuration.
log.syslog.portal	Portal service syslog integration configuration.
log.syslog.upload	Upload service syslog integration configuration.
log.syslog.lastFetchedCRL.backend	Keeps the last updated CRL as PEM formatted string for service syslog and updated regularly.
log.syslog.lastFetchedCRL.portal	Keeps the last updated CRL as PEM formatted string for service syslog and updated regularly.
log.syslog.lastFetchedCRL.upload	Keeps the last updated CRL as PEM formatted string for service syslog and updated regularly.

Table 26: TACACS

System Property	Description
session.options.enableTACACS	Activates or deactivates the TACACS services for Enterprise users.

Table 27: Two-factor Authentication

System Property	Description
vco.enterprise.authentication.twoFactor.enable	Activates or deactivates the two-factor authentication for Enterprise users.
vco.enterprise.authentication.twoFactor.mode	Defines the mode for the second level authentication for Enterprise users. Currently, only SMS is supported as the second level authentication mode.
vco.enterprise.authentication.twoFactor.require	Defines the two-factor authentication as mandatory for Enterprise users.
vco.operator.authentication.twoFactor.enable	Activates or deactivates the two-factor authentication for Operator users.
vco.operator.authentication.twoFactor.mode	Defines the mode for the second level authentication for Operator users. Currently, only SMS is supported as the second level authentication mode.
vco.operator.authentication.twoFactor.require	Defines the two-factor authentication as mandatory for Operator users.

Table 28: IPv6 Certificate Authentication

System Property	Description
session.options.enableNsdPkiIPv6Config	Activates **Certificate** Authentication mode and **IPv6** Local Identification Type.

Table 29: VNF

System Property	Description

edge.vnf.extraImageInfos	Defines the properties of a VNF Image.
	You can enter the following information for a VNF Image, in JSON format in the **Value** field:
	`[ { "vendor": " Vendor Name", "version": " VNF Image Version", "checksum": " VNF Checksum Value", "checksumType": " VNF Checksum Type" } ]`
	Example of JSON file for Check Point Firewall Image:
	`[ { "vendor": "checkPoint", "version": "r80.40_no_workaround_46", "checksum": "bc9b06376cdbf210cad8202d728f1602b79cfd7d", "checksumType": "sha-1" } ]`
	Example os JSON file for Fortinet Firewall Image:
	`[ { "vendor": "fortinet", "version": "624", "checksum": "6d9e2939b8a4a02de499528c745d76bf75f9821f", "checksumType": "sha-1" } ]`

edge.vnf.metric.record.limit	Defines the number of records to be stored in the database.
enterprise.capability.edgeVnfs.enable	Allows VNF deployment on supported Edge models.
enterprise.capability.edgeVnfs.securityVnf.checkPoint	Activates Check Point Networks Firewall VNF.
enterprise.capability.edgeVnfs.securityVnf.fortinet	Activates Fortinet Networks Firewall VNF.
enterprise.capability.edgeVnfs.securityVnf.paloAlto	Activates Palo Alto Networks Firewall VNF.
session.options.enableVnf	Activates VNF feature.
vco.operator.alert.edgeVnfEvent.enable	Activates or deactivates Operator alerts for Edge VNF events globally.
vco.operator.alert.edgeVnfInsertionEvent.enable	Activates or deactivates Operator alerts for Edge VNF Insertion events globally.
edge.vnf.extraImageInfos.	Allows selection of the Check Point VNF image.

Table 30: VPN Tunnel

System Property	Description
vpn.disconnect.wait.sec	The time interval for the system to wait before disconnecting a VPN tunnel.
vpn.reconnect.wait.sec	The time interval for the system to wait before reconnecting a VPN tunnel.

Table 31: Warning Message

System Property	Description
login.warning.banner.message	This optional system property allows the Operator to configure and display a Security Administrator-specified advisory notice and consent warning message regarding the use of Orchestrator. The warning message is displayed in the Orchestrator prior to user login. For instructions about how to configure this system property, see the topic *Configure Advisory Notice and Consent Warning Message for SD-WAN Orchestrator*.

Table 32: Zscalar Settings

System Property	Description
session.options.enableZscalerProfileAutomation	Enables to configure Zscaler settings at the Profile level.

Chapter 5

Configure Orchestrator Disaster Recovery

This section provides disaster recovery (DR) instructions for Orchestrator.

5.1 Orchestrator Disaster Recovery Overview

The Orchestrator Disaster Recovery (DR) feature prevents the loss of stored data and resumes Orchestrator services in the event of system or network failure.

Orchestrator DR involves setting up an active/standby Orchestrator pair with data replication and a manually-triggered failover mechanism.

- The recovery time objective (RTO), therefore, is dependent on explicit action by the operator to trigger promotion of the standby.
- The recovery point objective (RPO), however, is essentially zero, regardless of the recovery time, because all configuration is instantaneously replicated. Monitoring data that would have been collected during the outage is cached on the Edges and Gateways pending promotion of the standby.

Note: DR is mandatory. For licensing and pricing, contact the Arista Sales team for support.

Active/Standby Pair

In an Orchestrator DR deployment, two identical Orchestrator systems are configured as an active / standby pair. The operator can view the state of DR readiness through the web UI on either of the servers. Edges and Gateways are aware of both Orchestrators, and while they receive configuration changes only from the active Orchestrator, they periodically send DR heartbeats to both systems to report their view of both servers and to query the DR system status. When the operator triggers a failover, the Edges and Gateways are informed of the change in their next DR heartbeat.

DR States

From the view of an operator, and of the edges and gateways, an Orchestrator has one of four DR states:

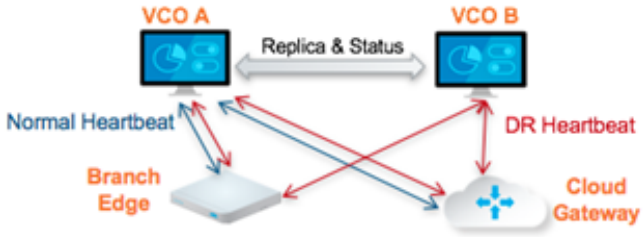
Table 33: DR States

DR State	Description
Standalone	No DR configured.
Active	DR configured, acting as the primary Orchestrator server.
Standby	DR configured, acting as an inactive replica Orchestrator server.
Zombie	DR formerly configured and active but no longer acting as the active or standby.

Run-time Operation

When DR is configured, the standby server runs in a limited mode, blocking all API calls except those related to the DR status and the DR heartbeats. When the operator invokes a failover, the standby is promoted to become fully operational as a Standalone server. The server that was formerly active is automatically transitioned to a Zombie state if it is responsive and visible from the promoted standby. In the Zombie state, management configuration services are blocked and any contact from Edges and Gateways that have not transitioned to the new active Orchestrator are redirected to the promoted server.

Figure 5-1: Run-time Operation



Phases	VCO A Role	VCO B Role
Initial	Standalone	Standalone
Pairing	Active	Standby
Failover	Zombie	Standalone

5.2 Set Up Orchestrator Replication

Two installed Orchestrator instances are required to initiate replication.

- The selected standby is put into a STANDBY_CANDIDATE state, enabling it to be configured by the active server.
- The active server is then given the address and credentials of the standby and it enters the ACTIVE_CONFIGURING state.

When a STANDBY_CONFIG_RQST is created from Active to Standby, the two servers synchronize through the state transitions.

The two Orchestrators for Disaster Recovery (DR) that will be established, must have the same time. Before you initiate Orchestrator replication, ensure you check the following NTP configurations:

- The Gateway time zone must be set to **Etc/UTC**. Use the following command to view the NTP time zone.

```
vcadmin@vcg1-example:~$ cat /etc/timezone Etc/UTC vcadmin@vcg1-example:~$
```

  If the time zone is incorrect, use the following commands to update the time zone.

```
echo "Etc/UTC" | sudo tee /etc/timezone sudo dpkg-reconfigure --frontend noninteractive tzdata
```

- The NTP offset must be less than or equal to 15 milliseconds. Use the following command to view the NTP offset.

```
sudo ntpqvcadmin@vcg1-example:~$ sudo ntpq -p remote refid st t when poll reach delay offset
 jitter ============================================================================== *ntp1-
us1.prod.v 74.120.81.219 3 u 474 1024 377 10.171 -1.183 1.033 ntp1-eu1-old.pr .INIT. 16 u -
 1024 0 0.000 0.000 0.000 vcadmin@vcg1-example:~$
```

If the offset is incorrect, use the following commands to update the NTP offset.

```
sudo systemctl stop ntp sudo ntpdate <server> sudo systemctl start ntp
```

- By default, a list of NTP Servers are configured in the `/etc/ntpd.conf` file. The Orchestrators on which DR need to be established must have Internet to access the default NTP Servers and ensure the time is in sync on both the Orchestrators. Customers can also use their local NTP server running in their environment to sync time.

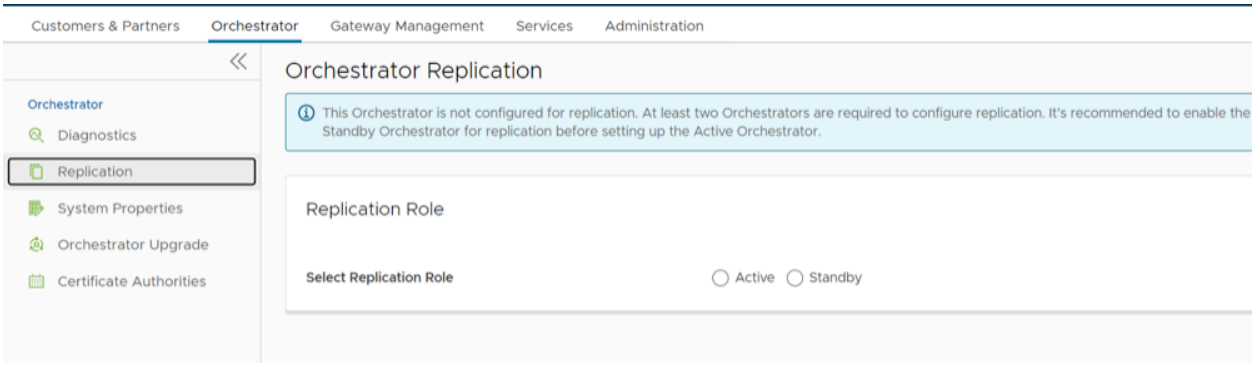
Note: Before you set up your Standby Orchestrator to begin the Replication process, you must enable the `network.public.address` system property.

5.2.1 Set Up the Standby Orchestrator

To set up Orchestrator replication, perform the following steps:

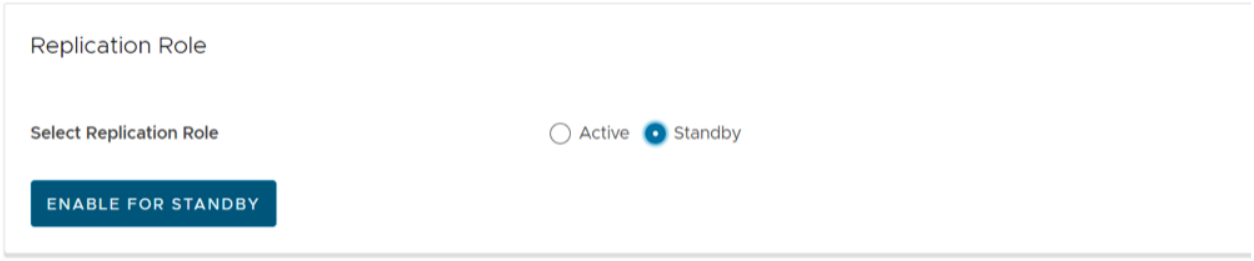
1. Select **Replication** from the Navigation panel to display the **Orchestrator Replication** screen.

   Figure 5-2: Orchestrator Replication



2. Enable the Standby Orchestrator by selecting the **Standby (Replication Role)** radio button.

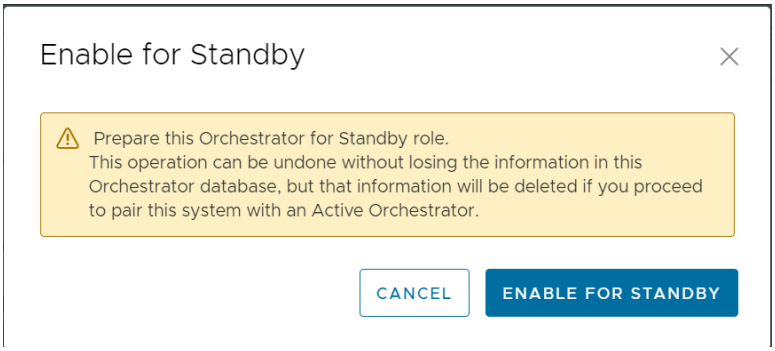
   Figure 5-3: Replication Role



3. Select the **Enable for Standby** button.

The **Prepare this Orchestrator for Standby Role** dialog appears.

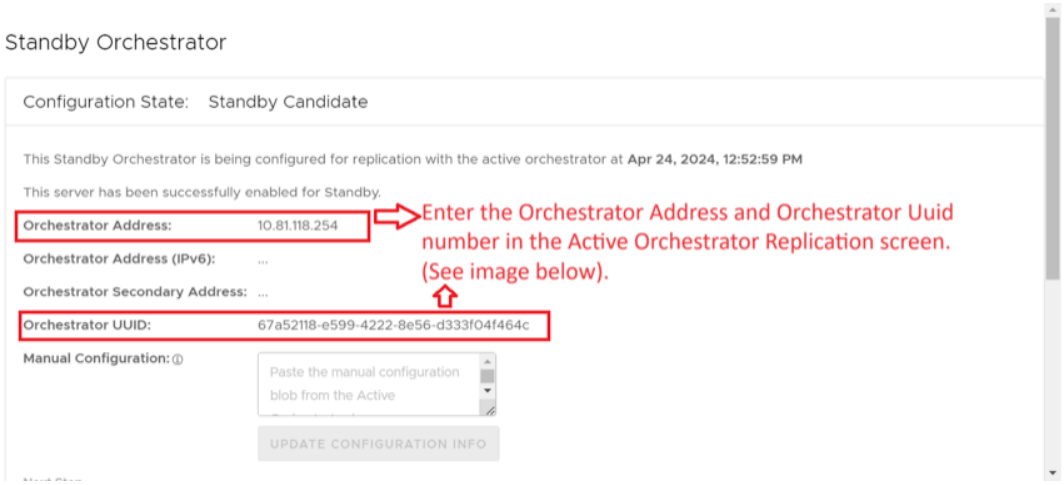
Figure 5-4: Enable for Standby



4. Select the **Enable for Standby** button again.

   The **Orchestrator Success** message appears across the top of the screen indicating that the Orchestrator has been enabled for Standby, and that the Orchestrator will restart in Standby mode.
5. Select **OK**.

Figure 5-5: Standby Orchestrator



After the Standby Orchestrator has been configured for replication, configure the Active Orchestrator. For additional information, see Set Up the Active Orchestrator.

5.2.2 Set Up the Active Orchestrator

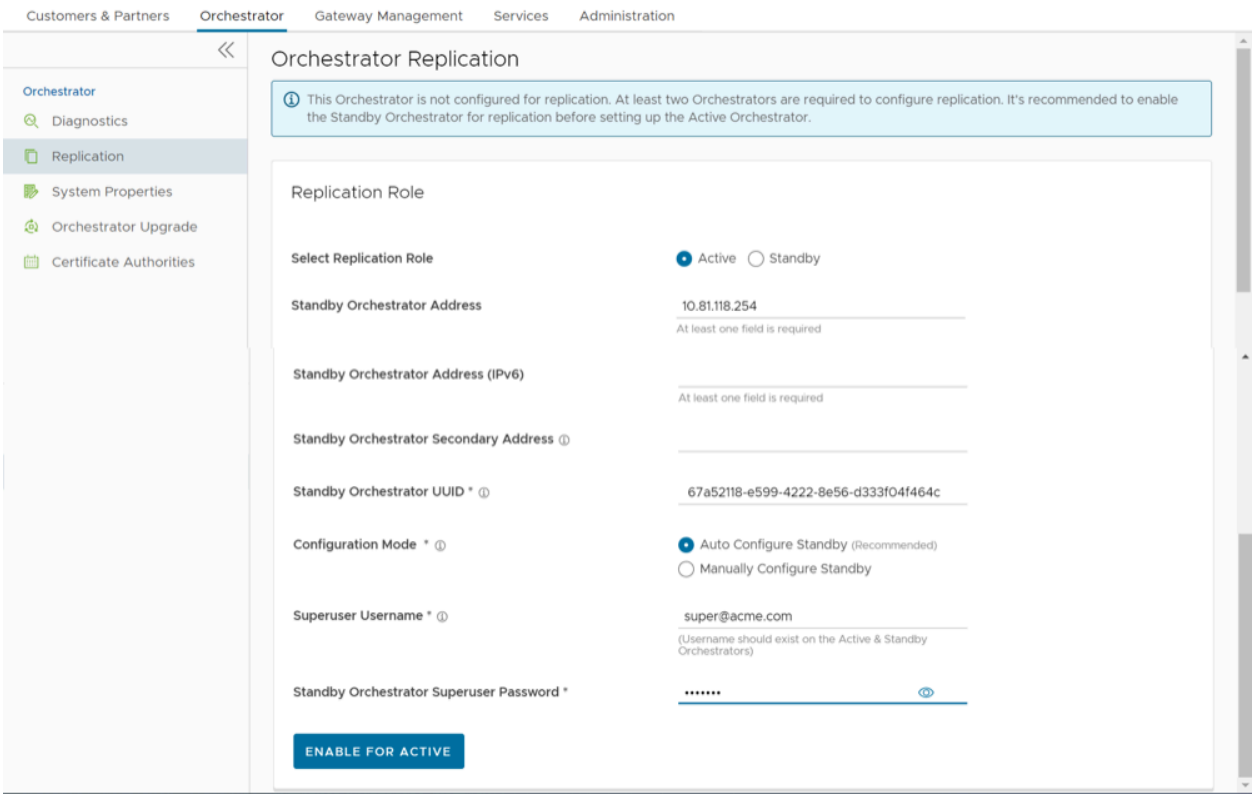
To configure the second Orchestrator to be the Active Orchestrator:

1. Select **Replication** from the Navigation panel.

   The **Orchestrator Replication** screen appears.
2. Choose the **Active Replication Role**.
3. Type in the **Standby Orchestrator Address** and the **Standby Orchestrator UUID**.

The Orchestrator Address and Uuid are displayed in the **Standby Orchestrator** screen.

Figure 5-6: Orchestrator Replication



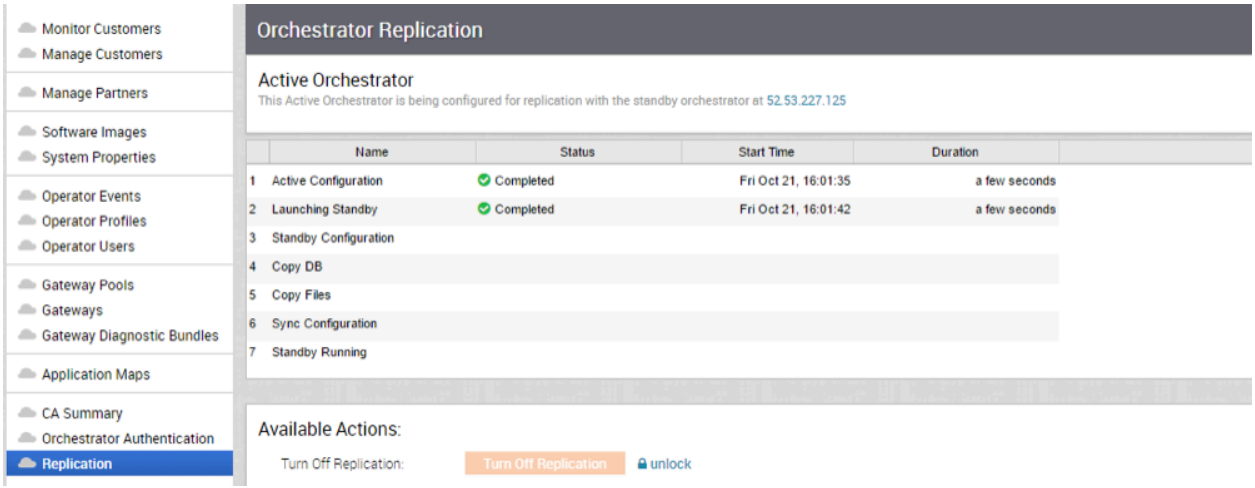
4. Type in the username and password for the Orchestrator Superuser to be used for replication.

> **Note:**
>
> - This Superuser should already exist on both systems.
> - Starting from the 4.5 release, the use of the special character "<" in the password is no longer supported. In cases where users have already used "<" in their passwords in previous releases, they must remove it to save any changes on the page.

5. Select the **Make Active** button.

The **Active Orchestrator** screen displays showing a status of the current state.

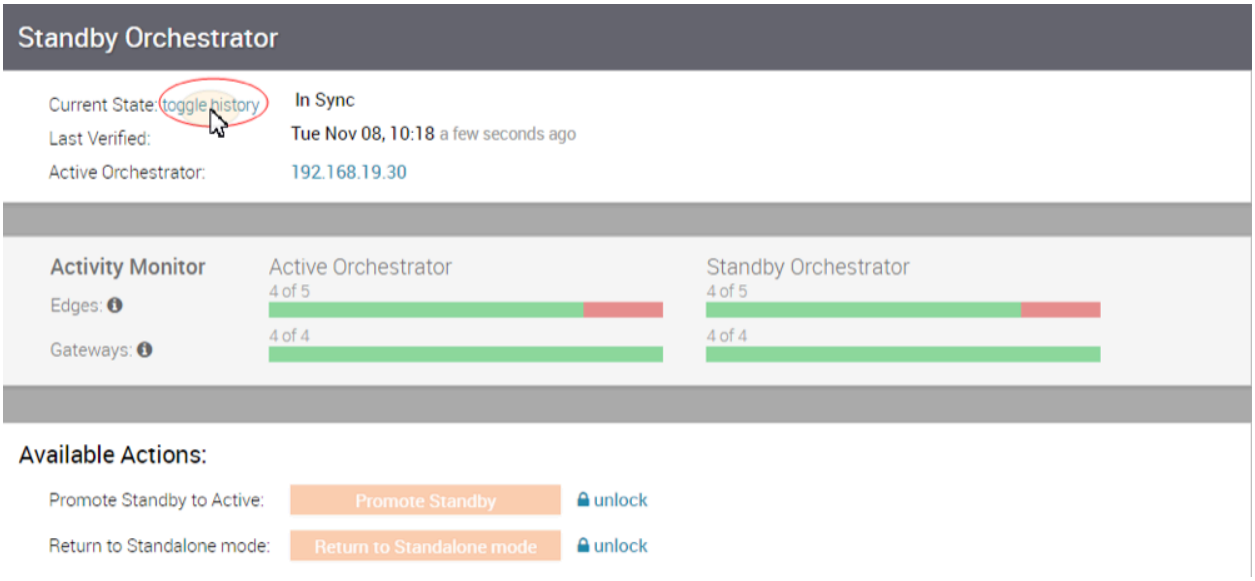
Figure 5-7: Active Orchestrator



When configuration is complete, both Orchestrators (Standby and Active) will be in sync.

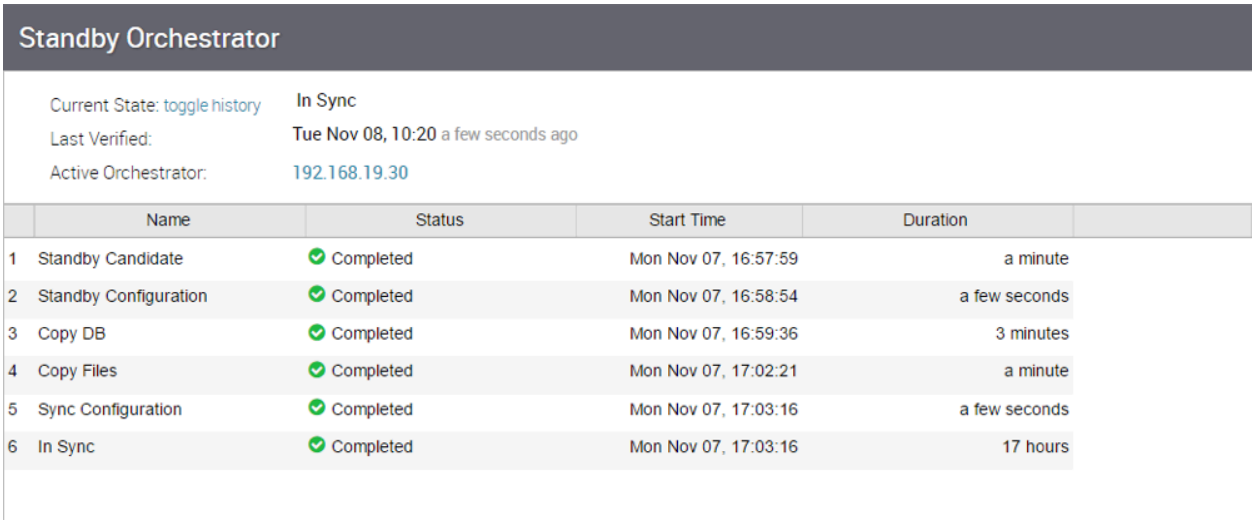
Standby Orchestrator in Sync

Figure 5-8: Standby Orchestrator in Sync



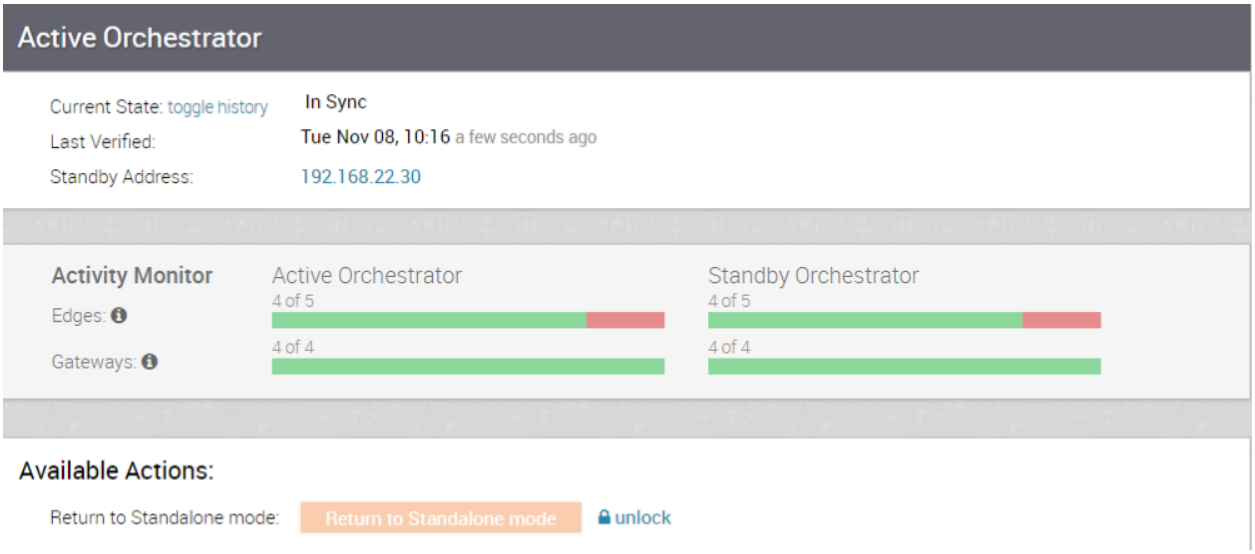
You can select the **toggle history** link to view the status of each state.

Figure 5-9: Standby Orchestrator



Active Orchestrator in Sync

Figure 5-10: Active Orchestrator in Sync



5.3 Test Failover

The following testing failover scenarios are forced failovers for example purposes. You can perform these actions in the Available Actions area of the Active and Standby screens.

5.3.1 Promote a Standby Orchestrator

This section discusses how to promote a Standby Orchestrator.

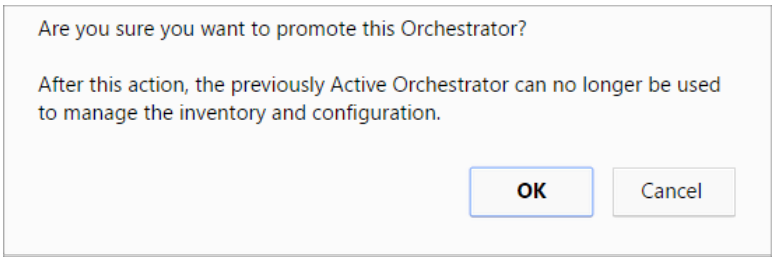
To promote a Standby Orchestrator:

1. Select the **unlock** link.
2. Select the **Promote Standby** button in the **Available Actions** area on the Standby Orchestrator screen.

   Figure 5-11: Available Actions Tab

   The following dialog box appears, indicating that when you promote your Standby Orchestrator, administrators will no longer be able to manage the SASE Orchestrator using the previously Active Orchestrator.

   Figure 5-12: Standby Orchestrator Dialog Box

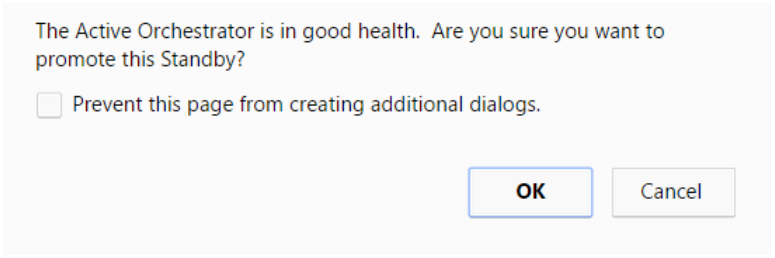


3. Select the **OK** button to promote the Standby Orchestrator.

   Another message dialog box appears to verify your request to promote the Standby Orchestrator. This message will appear only if the Standby Orchestrator perceives the Active Orchestrator to be in good health, meaning the Standby is communicating with the Active and duplicating data.

4. Select **OK** to promote the Orchestrator.

   Figure 5-13: Active Orchestrator Dialog Box



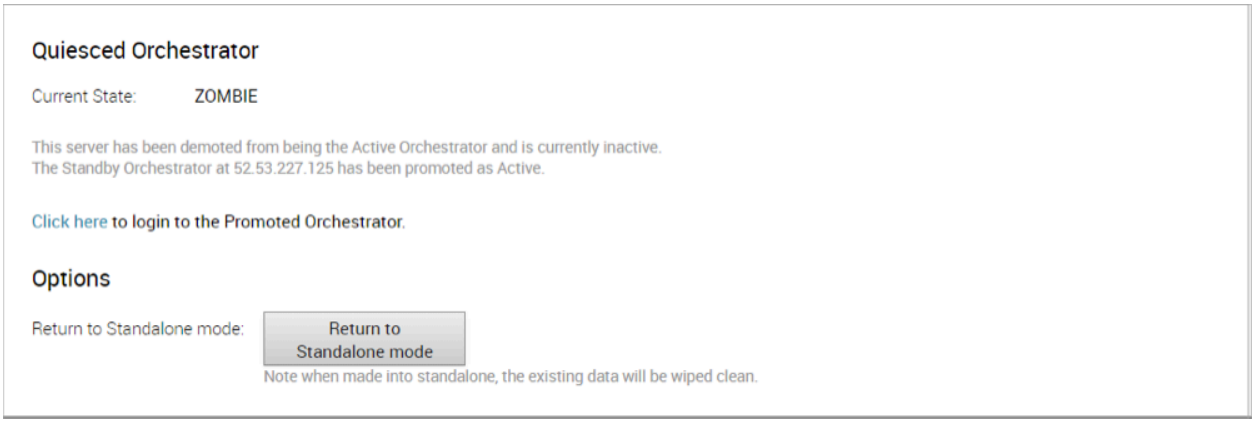
A final dialog box appears indicating that the Orchestrator is no longer a Standby and will restart in Standalone mode.

Figure 5-14: Standalone Mode Orchestrator Dialog Box

When you promote a Standby Orchestrator, it restarts in Standalone mode.

If the Standby can communicate with the formerly Active Orchestrator, it will instruct that Orchestrator to enter a Zombie state. In Zombie state, the Orchestrator communicates with its clients (edges, gateways, UI/API) that it is no longer active, and that they must communicate with the newly promoted Orchestrator. If the promoted Standby cannot communicate with the formerly Active Orchestrator, the operator should, if possible, manually demote the formerly Active Orchestrator.

Figure 5-15: Quiesced Orchestrator



5.3.2 Return to Standalone Mode

To return the Zombie to standalone mode, click the Return to Standalone Mode button in the Available Actions area on the Active Orchestrator or Standby Orchestrator screens.

Figure 5-16: Available Actions for Orchestrator

Note: The Orchestrator can be returned to the Standalone mode from the Zombie state after the time specified in the system property **`vco.disasterRecovery.zombie.expirySeconds`**, which is defaulted to 1800 seconds.

5.4 Troubleshooting SASE Orchestrator DR

This section discusses the failure states of the system. These are also listed in the UI, along with a more detailed description of the failure. Additional information is available in the log.

Recoverable Failures

The following errors are recoverable failures that can occur after SASE Orchestrator DR reaches an in sync state. If the problem causing these failures is corrected, SASE Orchestrator DR will automatically return to normal operation.

- FAILURE_SYNCING_FILES
- FAILURE_GET_STANDBY_STATUS
- FAILURE_MYSQL_ACTIVE_STATUS
- FAILURE_MYSQL_STANDBY_STATUS

Unrecoverable Failures

The following failures can occur during configuration of the SASE Orchestrator DR. SASE Orchestrator DR will not automatically recover from these failures.

- FAILURE_ACTIVE_CONFIGURING
- FAILURE_LAUNCHING_STANDBY
- FAILURE_STANDBY_CONFIGURING
- FAILURE_COPYING_DB
- FAILURE_COPYING_FILES
- FAILURE_SYNC_CONFIGURING
- FAILURE_GET_STANDBY_CONFIG
- FAILURE_STANDBY_CANDIDATE

- FAILURE_STANDBY_UNCONFIG
- FAILURE_STANDBY_PROMOTION
- FAILURE_ACTIVE_DEMOTION

5.5 Replication

The VeloCloud Orchestrator Disaster Recovery (DR) feature prevents the loss of stored data and resumes VeloCloud Orchestrator services in the event of system or network failure.

VeloCloud Orchestrator DR involves setting up an active/standby VeloCloud Orchestrator pair with data replication and a manually-triggered failover mechanism.

- The Recovery Time Objective (RTO), therefore, is dependent on explicit action by the operator to trigger promotion of the standby.
- The Recovery Point Objective (RPO), however, is essentially zero, regardless of the recovery time, because all configuration is instantaneously replicated. Monitoring data that would have been collected during the outage is cached on the Edges and Gateways pending promotion of the standby.

Note: DR is mandatory. For licensing and pricing, contact the Arista sales team for support.

Active/Standby Pair

In a VeloCloud Orchestrator DR deployment, two identical VeloCloud Orchestrator systems are configured as an active / standby pair. The operator can view the state of DR readiness through the web UI on either of the servers. Edges and gateways are aware of both VeloCloud Orchestrators, and while they receive configuration changes only from the active VeloCloud Orchestrator, they periodically send DR heartbeats to both systems to report their view of both servers and to query the DR system status. When the operator triggers a failover, the Edges and Gateways are informed of the change in their next DR heartbeat.

DR States

From the view of an operator, and the Edges and Gateways, a VeloCloud Orchestrator has one of the following four DR states:

Table 34: DR States

DR State	Description
Standalone	No DR configured.
Active	DR configured, acting as the primary VeloCloud Orchestrator server.
Standby	DR configured, acting as an inactive replica VeloCloud Orchestrator server.
Zombie	DR formerly configured and active but no longer acting as the active or standby.

Run-time Operation

When DR is configured, the standby server runs in a limited mode, blocking all API calls except those related to the DR status and the DR heartbeats. When the operator invokes a failover, the standby is promoted to become fully operational as a Standalone server. The server that was formerly active is automatically

transitioned to a Zombie state if it is responsive and visible from the promoted standby. In the Zombie state, management configuration services are blocked and any contact from edges and gateways that have not transitioned to the new active VeloCloud Orchestrator are redirected to the promoted server.

Figure 5-17: Run-time Operation

Phases	VCO A Role	VCO B Role
Initial	Standalone	Standalone
Pairing	Active	Standby
Failover	Zombie	Standalone

Set Up VeloCloud Orchestrator Replication

Two installed VeloCloud Orchestrator instances are required to initiate replication.

- The selected standby is put into a STANDBY_CANDIDATE state, enabling it to be configured by the active server.
- The active server is then given the address and credentials of the standby and it enters the ACTIVE_CONFIGURING state.

When a STANDBY_CONFIG_RQST is made from active to standby, the two servers synchronize through the state transitions.

The two Orchestrators on which Disaster Recovery (DR) need to be established must have same time. Before you initiate VeloCloud Orchestrator replication, ensure you check the following NTP configurations:

- The Gateway time zone must be set to **Etc/UTC**. Use the following command to view the NTP time zone.

```
vcadmin@vcg1-example:~$ cat /etc/timezone Etc/UTC vcadmin@vcg1-example:~$
```

  If the time zone is incorrect, use the following commands to update the time zone.

```
echo "Etc/UTC" | sudo tee /etc/timezone sudo dpkg-reconfigure --frontend noninteractive tzdata
```

- The NTP offset must be less than or equal to 15 milliseconds. Use the following command to view the NTP offset.

```
sudo ntpqvcadmin@vcg1-example:~$ sudo ntpq -p remote refid st t when poll reach delay offset
 jitter ============================================================================== *ntp1-
us1.prod.v 74.120.81.219 3 u 474 1024 377 10.171 -1.183 1.033 ntp1-eu1-old.pr .INIT. 16 u -
 1024 0 0.000 0.000 0.000 vcadmin@vcg1-example:~
```

  If the offset is incorrect, use the following commands to update the NTP offset.

```
sudo systemctl stop ntp sudo ntpdate <server> sudo systemctl start ntp
```

- By default, a list of NTP Servers are configured in the `/etc/ntpd.conf` file. The Orchestrators on which DR need to be established must have Internet to access the default NTP Servers and ensure the

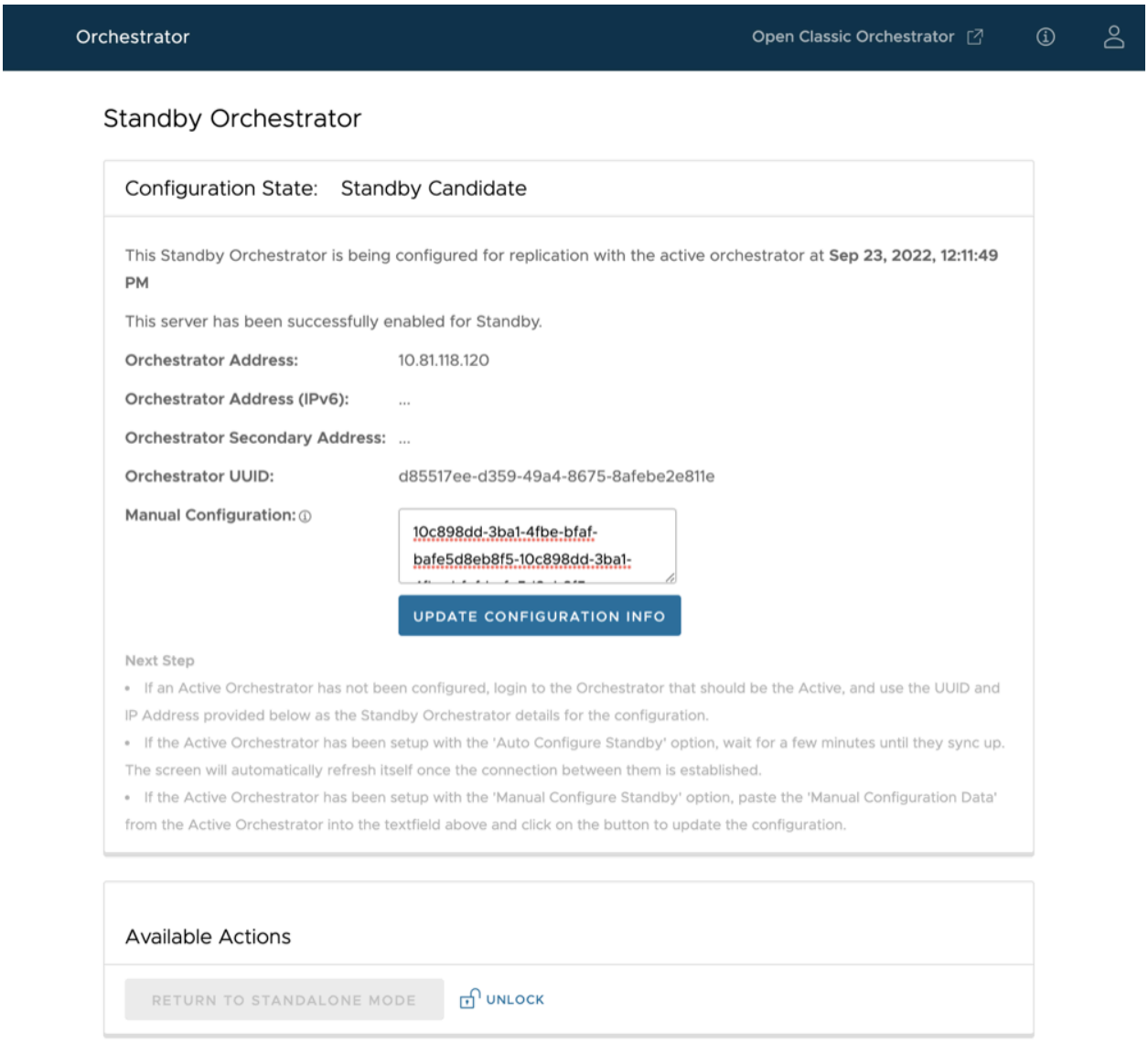
time is in sync on both the Orchestrators. Customers can also use their local NTP server running in their environment to sync time.

Set Up the Standby Orchestrator

To set up the Standby Orchestrator, perform the following steps:

1. In the **SD-WAN** service of the **Enterprise** Portal, select **Orchestrator** tab and then from the left pane select **Replication** button to display the **Orchestrator Replication** screen.
2. Activate the Standby Orchestrator by selecting the **Standby** (Replication Role) radio button.
3. Select **Enable for Standby** button.

   Figure 5-18: Standby Orchestrator



The Standby Orchestrator page appears.

4. Enter the **manual configuration** parameters and select **Update configuration info** button.

After the Standby Orchestrator has been configured for replication, configure the Active Orchestrator according to the instructions below.

Set Up the Active Orchestrator

To set up the Active Orchestrator, select the Replication Role as Active and configure the following:

Figure 5-19: Orchestrator Replication

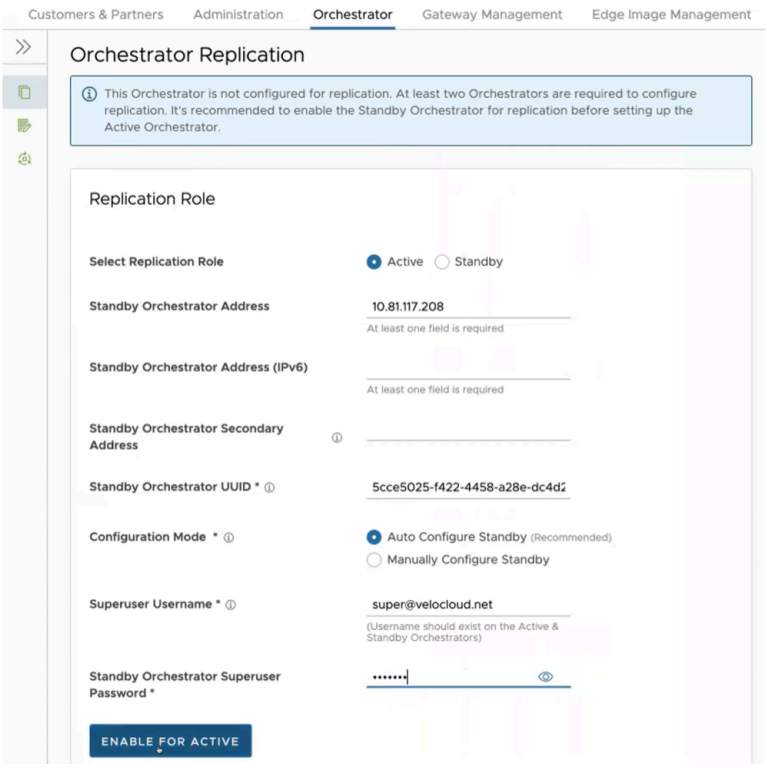


Table 35: Orchestrator Replication Fields

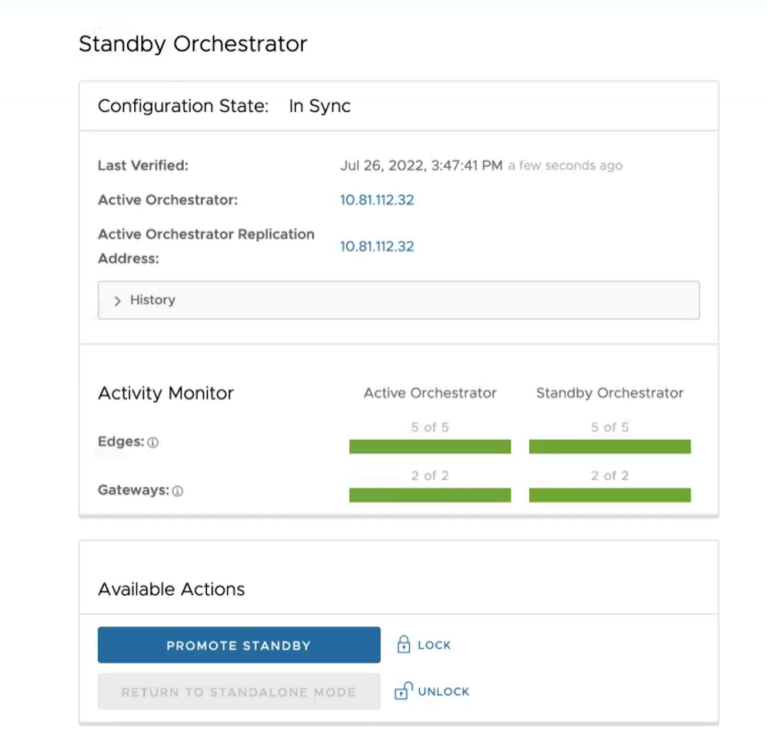
Option	Description
Select Replication Role	Select the **Active** radio button for the replication role.
Standby Orchestrator Address	Enter the primary Standby Orchestrator IP Address.
Standby Orchestrator Address (IPv6)	Enter the Standby Orchestrator IPv6 Address.
Standby Orchestrator Secondary Address	Enter the address of the standby Orchestrator's secondary interface. This address is used for replication if the standby is promoted to active. Users can add Ipv4/Ipv6 or FQDN address here.
Standby Orchestrator UUID	Enter the UUID of the standby Orchestrator.
Configuration Mode	Select the **Auto Configure Standby** or **Manually Configure Standby** radio button based on the requirement. When configured manually, paste a string value from ACTIVE VCO to STANDBY_WAIT .
Superuser Username	Enter the display name for the Orchestrator Superuser.
Standby Orchestrator Superuser Password	Enter the password for the Orchestrator Superuser. **Note:** Starting from the 4.5 release, the use of the special character "<" in the password is no longer supported. In cases where users have already used "<" in their passwords in previous releases, they must remove it to save any changes on the page.

Select **Enable for Active** button to activate replication role.

When configuration is complete, both Orchestrators (Standby and Active) are in sync.

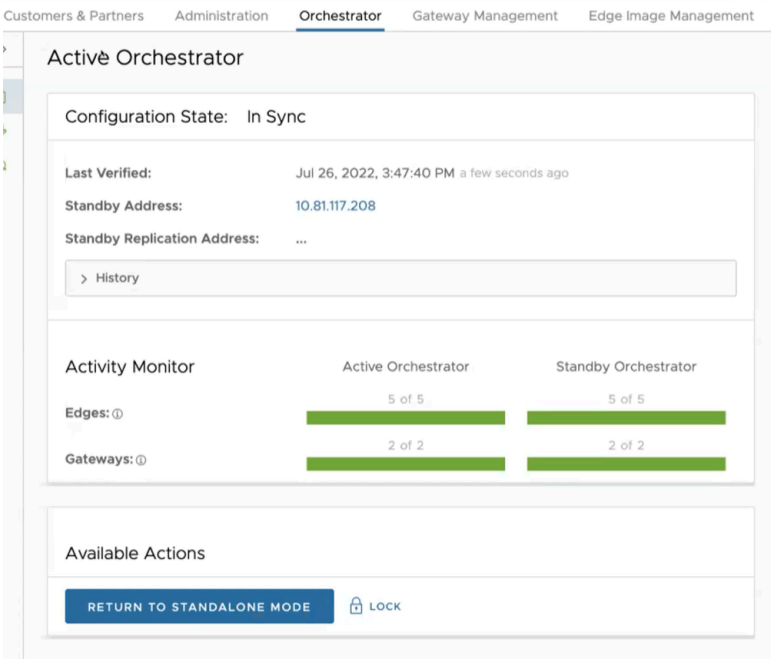
Standby Orchestrator in Sync

Figure 5-20: Configuration Status of Orchestrator



Active Orchestrator in Sync

Figure 5-21: Active Orchestrator Status



Test Failover

The following testing failover scenarios are forced failovers for example purposes. You can perform these actions in the **Available Actions** area of the **Active** and **Standby**screens.

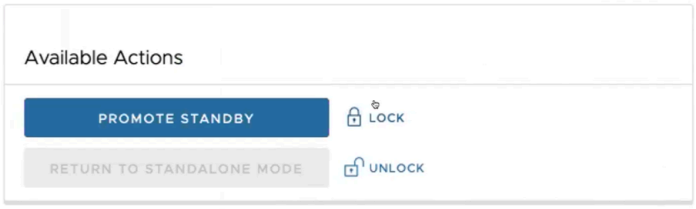
Promote a Standby Orchestrator

This section discusses how to promote a Standby Orchestrator.

To promote a Standby Orchestrator, perform the following steps:

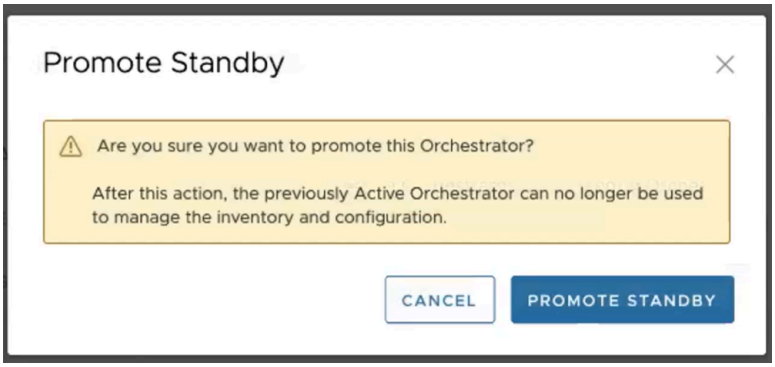
1. Select the **unlock** link.
2. Select the **Promote Standby** button in the **Available Actions** area on the Standby Orchestrator screen.

   Figure 5-22: Available Actions



   The following dialog box appears, indicating that when you promote your Standby Orchestrator, administrators can no longer be able to manage the VeloCloud Orchestrator using the previously Active Orchestrator.

   Figure 5-23: Promote Standby Orchestrator



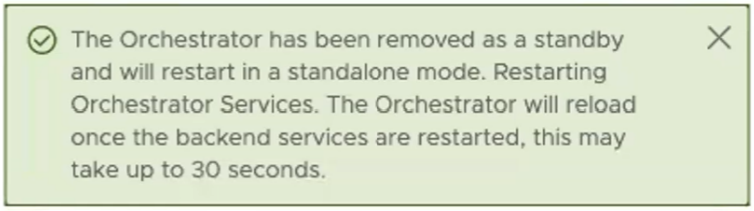
3. Select the **Promote Standby** button to promote the Standby Orchestrator.

4. Select **Force Promote Standby** to promote the Orchestrator.

Figure 5-24: Force Promote Standby Orchestrator

A final dialog box appears indicating that the Orchestrator is no longer a Standby and restarts in Standalone mode.

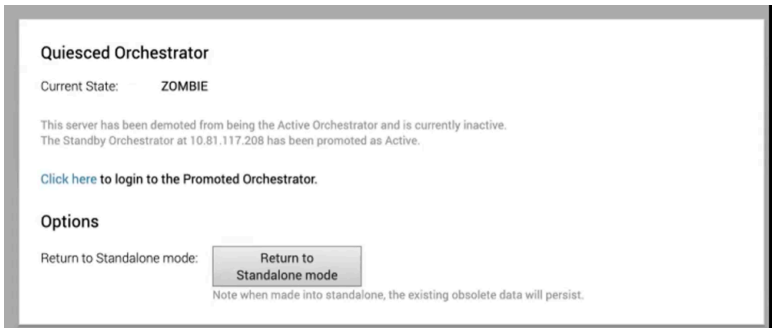
Figure 5-25: Orchestrator Removed Status



When you promote a Standby Orchestrator, it restarts in Standalone mode.

If the Standby can communicate with the formerly Active Orchestrator, it instructs that Orchestrator to enter a Zombie state. In Zombie state, the Orchestrator communicates with its clients (edges, gateways, UI/API) that it is no longer active, and that they must communicate with the newly promoted Orchestrator. If the promoted Standby cannot communicate with the formerly Active Orchestrator, the operator should, if possible, manually demote the formerly Active Orchestrator.

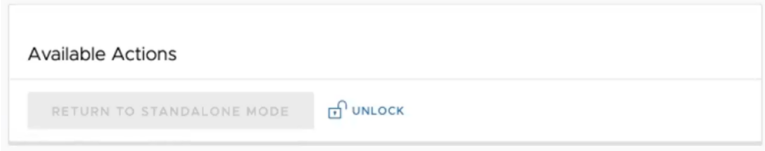
Figure 5-26: Quiesced Orchestrator



Return to Standalone Mode

To return the Zombie to standalone mode, select the **Return to Standalone Mode** button in the **Available Actions** area on the **Active Orchestrator** or **Standby Orchestrator** screens.

Figure 5-27: Return to Standalone Mode



Note: The Orchestrator can be returned to the Standalone mode from the Zombie state after the time specified in the system property `vco.disasterRecovery.zombie.expirySeconds`, which is defaulted to 1800 seconds.

Troubleshooting VeloCloud Orchestrator DR

This section describes the failure states of the system. These are also listed in the UI, along with a more detailed description of the failure. Additional information is available in the Arista log.

Recoverable Failures

The following errors are recoverable failures that can occur after VeloCloud Orchestrator DR reaches an in sync state. If the problem causing these failures is corrected, VeloCloud Orchestrator DR automatically returns to normal operation.

- FAILURE_SYNCING_FILES
- FAILURE_GET_STANDBY_STATUS
- FAILURE_MYSQL_ACTIVE_STATUS
- FAILURE_MYSQL_STANDBY_STATUS

Unrecoverable Failures

The following failures can occur during configuration of the VeloCloud Orchestrator DR. VeloCloud Orchestrator DR does not automatically recover from these failures.

- FAILURE_ACTIVE_CONFIGURING
- FAILURE_LAUNCHING_STANDBY
- FAILURE_STANDBY_CONFIGURING
- FAILURE_COPYING_DB
- FAILURE_COPYING_FILES
- FAILURE_SYNC_CONFIGURING
- FAILURE_GET_STANDBY_CONFIG
- FAILURE_STANDBY_CANDIDATE
- FAILURE_STANDBY_UNCONFIG

- FAILURE_STANDBY_PROMOTION
- FAILURE_ACTIVE_DEMOTION

Chapter 6

Upgrade Orchestrator

This section discusses how to upgrade the VeloCloud Orchestrator.

6.1 Orchestrator Upgrade Overview

The following steps are required to upgrade the VeloCloud Orchestrator.

1. Prepare for the Orchestrator Upgrade.
2. Send Upgrade Announcement.
3. Proceed with the Orchestrator upgrade.
4. Complete the Orchestrator Upgrade.

6.2 Upgrade an Orchestrator

This section discusses how to upgrade an Orchestrator.

6.2.1 Step 1: Prepare for the Orchestrator Upgrade

Contact Arista Support team to prepare for the Orchestrator upgrade.

To upgrade Orchestrator:

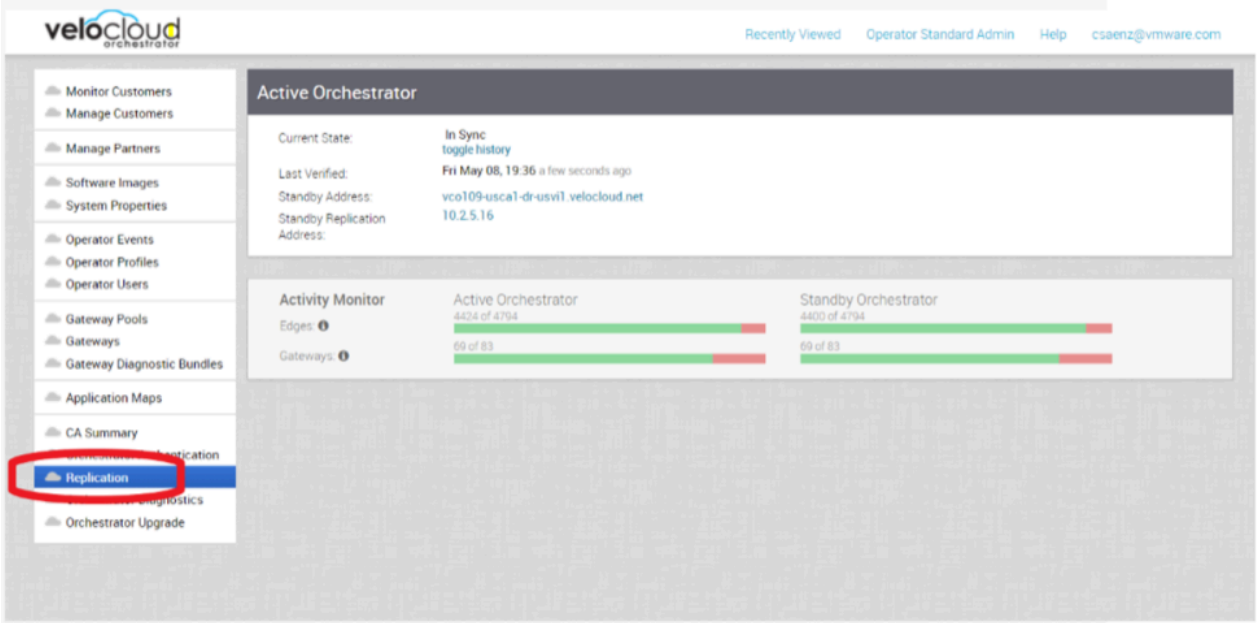
Arista Support assists you with your upgrade. Collect the following information prior to contacting Support.

- Provide the current and target Orchestrator versions, for example: current version (i.e. 2.5.2 GA-20180430), target version (3.3.2 p2).

Note: For the current version, this information can be found on the top, right corner of the Orchestrator by selecting the **Help** link and choosing **About**.

- Provide a screenshot of the replication dashboard of the Orchestrator.

Figure 6-1: Orchestrator Dashboard



- Hypervisor Type and version (i.e. vSphere 6.7)
- Commands from the Orchestrator:

> **Note:** Commands must be run as root (e.g. **`sudo <command>`** or **`sudo-i`**).

  - Run the script `/opt/vc/scripts/vco_upgrade_check.sh` to check:
    - LVM layout
    - Memory Information
    - CPU Information
    - Kernel Parameters
    - Some system properties
    - ssh configurations
    - Mysql schema and database sizes
    - File_store locations and sizes
  - Copy of `/var/log`
    - `tar -czf /store/log-`date +%Y%M%S`.tar.gz --newer-mtime="36 hours ago" /var/log`
  - From the Standby Orchestrator:
    - `sudo mysql --defaults-extra-file=/etc/mysql/velocloud.cnf velocloud -e 'SHOW SLAVE STATUS \G'`

- From the Active Orchestrator:
  - `sudo mysql --defaults-extra-file=/etc/mysql/velocloud.cnf velocloud -e 'SHOW MASTER STATUS \G'`

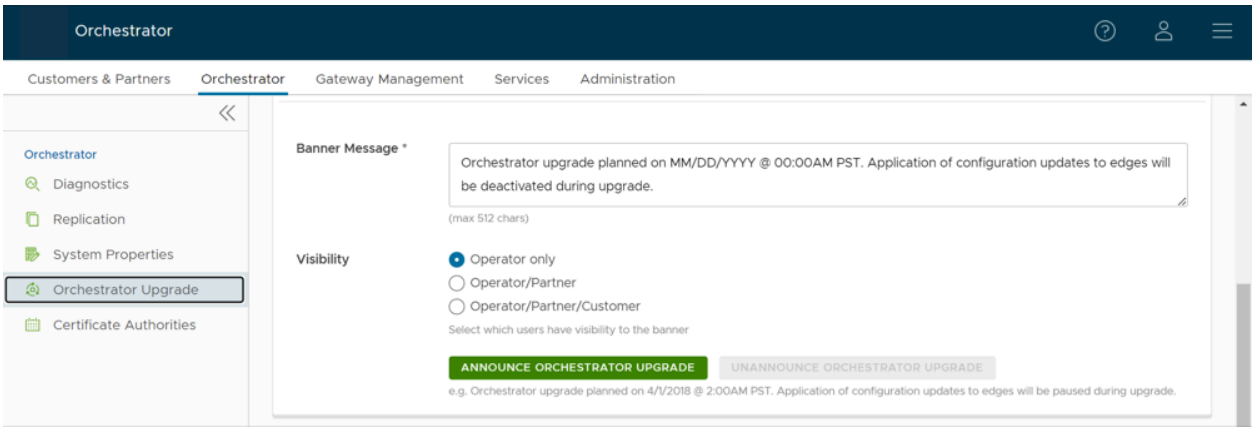
6.2.2 Step 2: Send Upgrade Announcement

The Upgrade Announcement area enables you to configure and send a message about an upcoming upgrade. This message will be displayed to all users the next time they login to the Orchestrator.

To send an upgrade announcement:

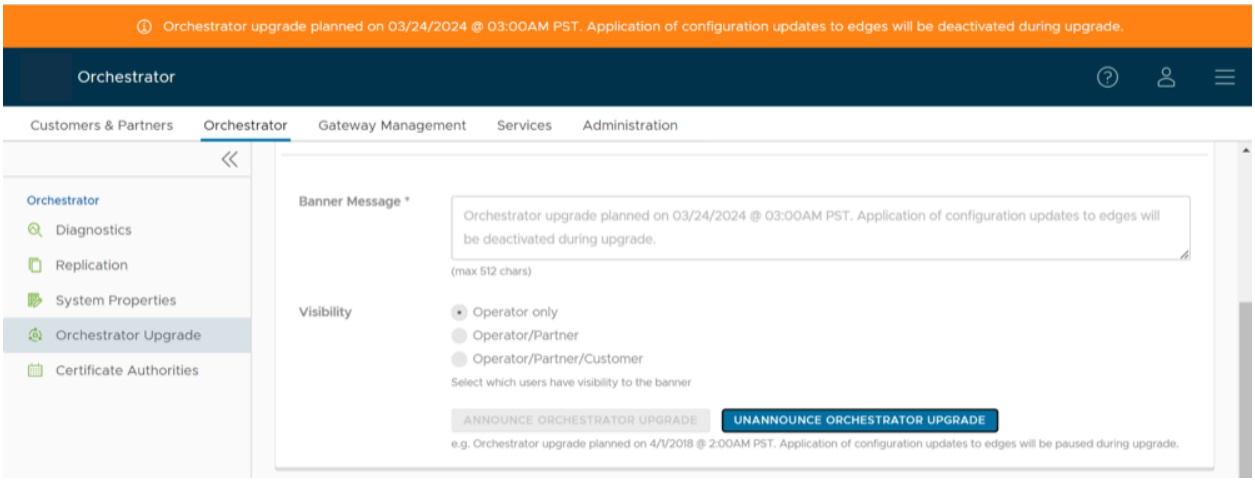
1. From the **Orchestrator**, select **Orchestrator Upgrade** from the navigation panel.
2. In the **Upgrade Announcement** area, type in your message in the **Banner Message** text box.

   Figure 6-2: Configuring the Upgrade Announcement



3. Select the **Announce Orchestrator Upgrade** button. A popup message appears indicating successful creation of your announcement and your banner message displays at the top of the Orchestrator.

   Figure 6-3: Removing the Upgrade Announcement



4. (Optional) You can remove the announcement from the Orchestrator by selecting **Unannounce Orchestrator Upgrade**. A message appears indicating you successfully unannounced the Orchestrator upgrade. The announcement displayed at the top of the Orchestrator disappears.

6.2.3 Step 3: Before Proceeding with the Orchestrator Upgrade

This section provides important information and best practices to consider prior to upgrading an Orchestrator. Remember to contact Arista Support to assist you with Orchestrator upgrades.

Consider the following when upgrading the Orchestrator:

- This upgrade work does not modify any existing APIs.
- Just like other releases, there are schema changes when upgrading to a newer release. However, these changes will not impact the upgrade process.

  The OS for the SD-WAN Orchestrator virtual appliance and the underlying data stores that store the configuration and statistics data are being upgraded. The specific upgrades include the following:

  - The OS version is changing from Ubuntu 14.04 to 18.04.
  - The Config store is moving to MySQL 8.0.
  - The Stats store is moving to ClickhouseDB.

Note: The Orchestrator OS, database, and several other dependent components currently in use have reached their end of life, and will no longer be supported.

Best Practices/Recommendations:

The following are some upgrade best practices:

- From the **System Properties** page in the Orchestrator, make a note of the value of the `edge.heartbeat.spread.factor` system property. Then, change the heartbeat spread factor to a relatively high value for a large Orchestrator (e.g. 20, 40, 60). This will help reduce the sudden spike of the resource utilization (CPU, IO) on the system. Make sure to verify that all Gateways and Edges are in a connected state before restoring the previous `edge.heartbeat.spread.factor` value from the **System Properties** page in the Orchestrator.
- Leave the demoted SD-WAN Orchestrator up for a few hours before complete shutdown or decommission.
- Freeze configuration modifications to avoid any additional configuration changes until the upgrade process is completed.

6.2.4 Step 4: Proceed with the Orchestrator Upgrade

Contact Arista Support at for assistance with the Orchestrator upgrade.

6.2.5 Step 5: Complete the Orchestrator Upgrade

After you have completed the Orchestrator upgrade, select **Complete Orchestrator Upgrade**. This re-enables the application of the configuration updates of Edges at the global level.

To verify that the status of the upgrade is complete, run the following command to display the correct version number for all the packages:

```
dpkg -l|grep vco
```

When you are logged in as an Operator, the same version number should display at the bottom right corner of the Orchestrator.

6.3 Orchestrator Disaster Recovery

This section discusses how to set up and upgrade disaster recovery in the VeloCloud Orchestrator.

6.3.1 Set Up Disaster Recovery

To set up disaster recovery in the Orchestrator:

1. Install a new Orchestrator whose version matches the Product version that is currently the Active Orchestrator.
2. Set the following properties on the Active and Standby Orchestrator, if necessary:
   a. Set `vco.disasterRecovery.transientErrorToleranceSecs` to a non-zero value (it defaults to 900 seconds in version 3.3 and later, but to zero in earlier versions). This prevents any transient errors from resulting in an Edge/Gateway management plane update.
   b. Set `vco.disasterRecovery.mysqlExpireLogsDays` (defaults is 1 day). This is the amount of time the Active Orchestrator keeps the `mysql binlog` data.
3. Set up the `network.public.address` property on the Active and Standby Orchestrators to the address contacted by the Edges (Heartbeats).
4. Set up DR by following the usual DR Setup procedure that is described in *Orchestrator Disaster Recovery*.

6.3.2 Upgrade the DR Setup

To upgrade a DR-enabled Orchestrator pair, follow the steps below:

Note: If the Orchestrator upgrade is from 2.X to 3.2.X, run `dr-standby-schema.sh` on the Standby before starting the upgrade.

1. Prepare for the Upgrade. For instructions, go to Step 1: Prepare for the Orchestrator Upgrade of the section titled, *Upgrade an Orchestrator with DR Deployment*.
2. Proceed with the Orchestrator upgrade. For instructions, go to Step 4: Proceed with the Orchestrator Upgrade of the section titled, *Upgrade an Orchestrator with DR Deployment*.

Chapter 7

Troubleshooting Orchestrator

This section discusses Orchestrator troubleshooting.

7.1 Orchestrator Diagnostics Overview

The Orchestrator Diagnostics bundle is a collection of diagnostic information that is required for Support and Engineering to troubleshoot the Orchestrator. For Orchestrator on-premises installation, Operators can collect the Orchestrator Diagnostic bundle from the Orchestrator UI and provide it to the Arista Support team for offline analysis and troubleshooting.

SD-WAN Orchestrator Diagnostics includes the following two diagnostic bundles:

- **Diagnostic Bundles Tab**: Request and download a diagnostic bundle. This information can be found in the *Arista SD-WAN Orchestrator Deployment and Monitoring Guide*. See the section titled, "*Diagnostic Bundle Tab*."
- **Database Statistics Tab**: Provides a read-only access view of some of the information from a diagnostic bundle. This information can be found in the *Arista SD-WAN Orchestrator Deployment and Monitoring Guide*. See the section titled, *Database Statistics Tab*.

7.1.1 Diagnostics Bundle Tab

Users can request and download a diagnostic bundle in the **Diagnostics Bundle** tab.

Columns in the Diagnostics Bundle Tab

The Orchestrator Diagnostics table grid includes the following columns:

Table 36: Orchestrator Diagnostics Table Description

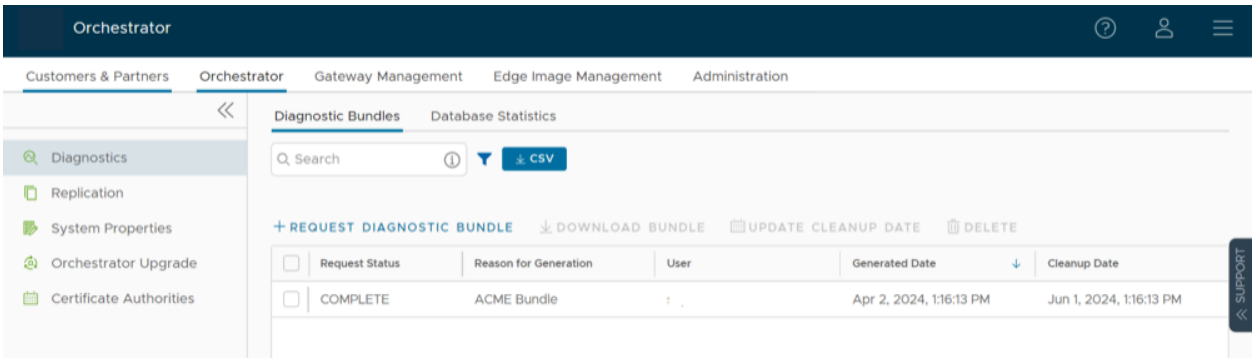
Column Name	Description
Request Status	There are two types of status requests: • Complete • In Progress If a bundle has not completed the download, the `In Progress` status appears.
Reason for Generation	The specific reason given for generating a diagnostic bundle. Select the **Request Diagnostic Bundle** button to include a description of the bundle.
User	The individual logged into the Orchestrator.
Generated	The date and time when the diagnostic bundle request was sent.
Cleanup Date	The default **Cleanup Date** is three months after the generated date, when the bundle will be automatically deleted. If you need to extend the Cleanup date period, select the **Cleanup Date** link located under the **Cleanup Date** column. For additional information, see *Updating Cleanup Date*.

Request a Diagnostic Bundle

To request a diagnostic bundle:

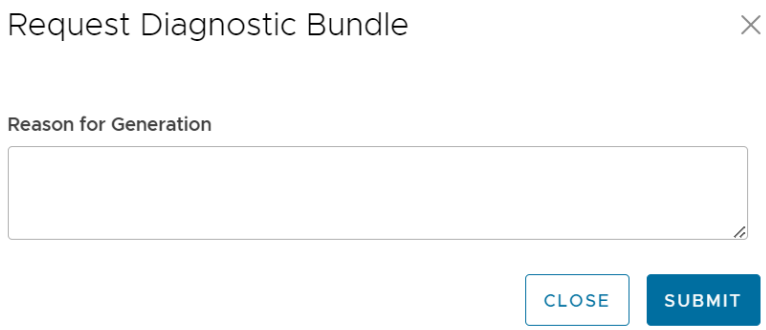
1. From the Orchestrator navigation panel, select **Diagnostics**.

   Figure 7-1: Diagnostics Screen



2. From the **Request Diagnostic Bundle** tab, select the **Request Diagnostic Bundle** button.
3. In the **Request Diagnostic Bundle** dialog, enter the reason for the request in the appropriate area.

   Figure 7-2: Request Diagnostic Bundle



4. Select **Submit**. The bundle request you created displays in the grid area of the **Diagnostic Bundle** screen with an `In Progress` status.
5. Refresh your screen to check the status of diagnostic bundle request. When the bundle is ready for download, a `Complete` status appears.

Download a Diagnostic Bundle

To download a diagnostic bundle:

1. Select a diagnostic bundle you want to download.
2. Select the **Actions** button, and choose **Download Diagnostic Bundle**. You can also select the **Complete** link to download the diagnostics bundle.

The diagnostics bundle downloads.

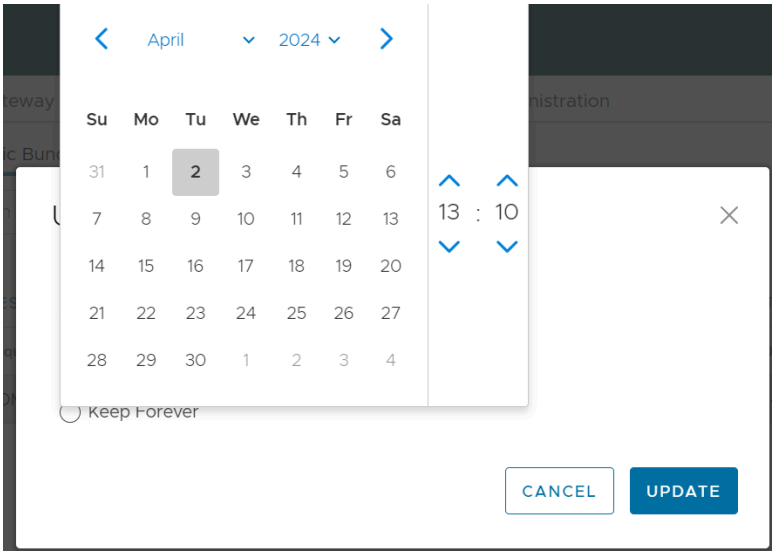
Update the Cleanup Date

The Cleanup date represents the date when the generated bundle will be automatically deleted, which by default is three months after the Generated date. You can change the Cleanup date or choose to keep the bundle indefinitely.

To update the Cleanup date:

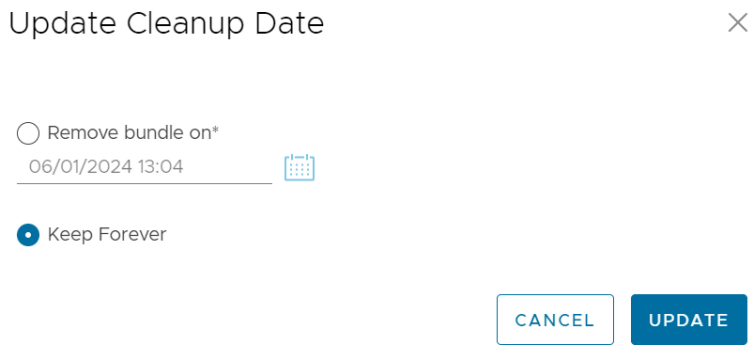
1. From the **Cleanup Date** column, select the **Cleanup Date** link of your chosen Diagnostic Bundle.
2. From the **Update Cleanup Date** dialog, select the **Calendar** icon to change the date.

Figure 7-3: Calendar Settings



3. You can also choose to keep the bundle indefinitely by checking the **Keep Forever** check box.

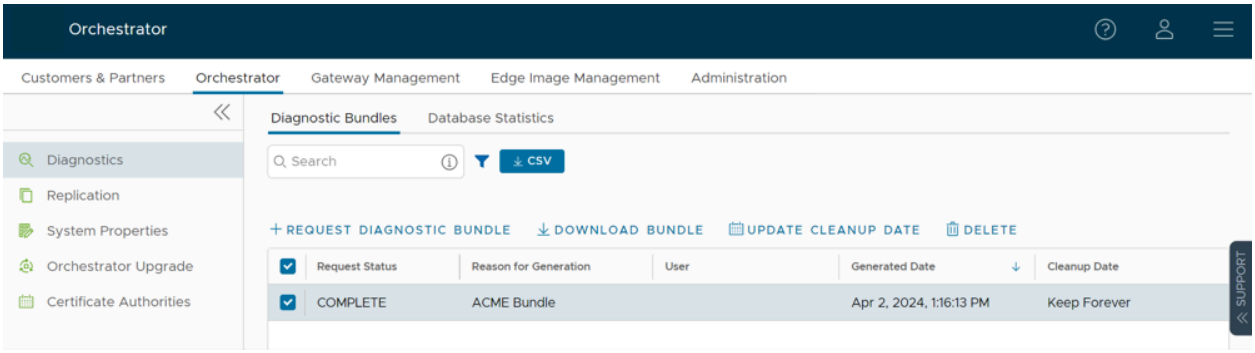
Figure 7-4: Update Cleanup Date



4. Select **OK**.

The Orchestrator Diagnostics table grid updates to reflect the changes to the Cleanup Date.

Figure 7-5: Table Grid Updates



7.1.2 Database Statistics Tab

The Database Statistics tab provides a read-only access view of some of the information from a diagnostic bundle.

If you require additional information, go to the **Diagnostic Bundles** tab, request a diagnostic bundle, and download it locally. For additional information, see *Request Diagnostic Bundle*.

The **Database Statistics** tab displays the following sections: Database Sizes, Database Table Statistics, Database Storage Info, Database Process List, Database Status Variable, Database System Variable, and Database Engine Status.

Figure 7-6: Orchestrator Database Statistics

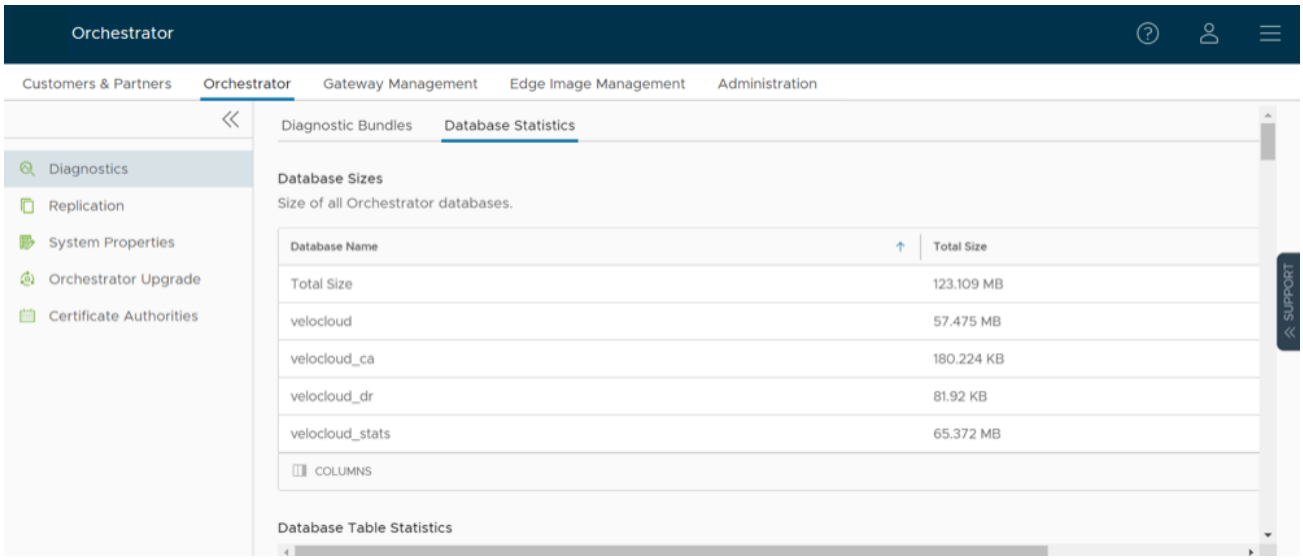


Table 37: Orchestrator Database Statistics Field Descriptions

Field	Description
Database Sizes	Sizes of the Orchestrator databases.
Database Table Statistics	Statistical details of all tables in the Orchestrator database.
Database Storage Info	Storage details of the mounted locations.
Database Process List	The top 20 records of long-running SQL queries.
Database Status Variable	The status variables of the MySQL server.
Database System Variable	System variables of the MySQL server.
Database Engine Status	The InnoDB engine status of the MySQL server.

7.2 System Metrics Monitoring

This section discusses System Metrics Monitoring on the Orchestrator.

Orchestrator System Metrics Monitoring Overview

The Orchestrator comes with a built-in system metrics monitoring stack, which includes a metrics collector and a time-series database. With the monitoring stack, you can easily check the health condition and the system load for the Orchestrator.

To enable the monitoring stack, run the following command on the orchestrator:

```
sudo /opt/vc/scripts/vco_observability_manager.sh enable
```

To check the status of the monitoring stack, run:

```
sudo /opt/vc/scripts/vco_observability_manager.sh status
```

To deactivate the monitoring stack, run:

```
sudo /opt/vc/scripts/vco_observability_manager.sh disable
```

The Metrics Collector

Telegraf is used as the Orchestrator system metrics collector, which includes plugins to collect system metrics. The following metrics are enabled by default.

Table 38: Metric Descriptions

Metric Name	Description
inputs.cpu	Metrics about CPU usage.
inputs.mem	Metrics about memory usage.
inputs.net	Metrics about network interfaces.
inputs.system	Metrics about system load and uptime.
inputs.processes	The number of processes grouped by status.
inputs.disk	Metrics about disk usage.
inputs.diskio	Metrics about disk IO by device.
inputs.procstat	CPU and memory usage for specific processes.
inputs.nginx	Nginx's basic status information (ngx_http_stub_status_module).
inputs.mysql	Statistic data from the MySQL server.
inputs.clickhouse	Metrics from one or many ClickHouse servers.
inputs.redis	Metrics from one or many redis servers.
inputs.filecount	The number and total size of files in specified directories.
inputs.ntpq	Standard NTP query metrics (requires ntpq executable).
Inputs.x509_cert	Metrics from a SSL certificate.

To activate more metrics or deactivate some enabled metrics, edit the Telegraf configuration file on the Orchestrator by the following:

- `sudo vi /etc/telegraf/telegraf.d/system_metrics_input.conf`
- `sudo systemctl restart telegraf`

The Time-series Database

Prometheus is used to store the system metrics collected by Telegraf. The metrics data will be kept in the database for three weeks at the most. By default, Prometheus listens on port 9090. If you have an external monitoring tool, provide the Prometheus database as a source, so that you can view the Orchestrator system metrics on your monitoring UI.

7.3 Rate Limiting API Requests

When there are too many API requests sent at a time, it affects the performance of the system. You can enable Rate Limiting, which enforces a limit on the number of API requests sent by each user.

The Orchestrator makes use of certain defence mechanisms that curb API abuse and provides system stability. API requests that exceed the allowed request limits are blocked and returned with HTTP 429 (Too many Requests). The system needs to go through a cool down period before making the requests again.

The following types of Rate-Limiters are deployed on Orchestrator:

- **Leaky bucket limiter** – Smooths the burst of requests and only allows a pre-defined number of requests. This limiter takes care of limiting the number of requests allowed in a given time window.
- **Concurrency limiter** – Limits the number of requests that occur in parallel which leads to concurrent requests fighting for resources and may result in long running queries.

The following are the major reasons that lead to rate limiting of the API requests:

- Large number of active or concurrent requests.
- Sudden spikes in request volume.
- Requests resulting in long running queries on the Orchestrator holding system resources for long being dropped.

Developers that rely on the API can adopt the following measures to improve the stability of their code when the VCO rate-limiting capability is enabled.

- Handle HTTP 429 response code when requests exceed rate limits.
- The penalty time duration is 5000 ms when the rate limiter reaches the maximum allowed requests in a given period. If blocked, the clients are expected to have a cool down period of 5000 ms before making requests again. The requests made during the cool down period of 5000 ms will still be rate limited.
- Use shorter time intervals for time series APIs which will not let the request to expire due to long running queries.
- Prefer batch query methods to those that query individual Customers or Edges whenever possible.

Note: Operator Super users configure Rate limits discretely based on the environment. For any queries on relevant policies, contact your Operator.

Configure Rate Limiting Policies using System Properties

You can use the following system properties to enable Rate Limiting and define the default set of policies:

- `vco.api.rateLimit.enabled`
- `vco.api.rateLimit.mode.logOnly`
- `vco.api.rateLimit.rules.global`
- `vco.api.rateLimit.rules.enterprise.default`
- `vco.api.rateLimit.rules.enterpriseProxy.default`

For additional information on the system properties, see List of System Properties.

Configure Rate Limiting Policies using APIs

It is recommended to configure the rate limiter policies as global rules using the system properties, as this approach produces the best possible API performance, facilitates troubleshooting, and ensures a consistent user experience across all Partners and Customers. In rare cases, however, Operators may determine that global policies are too lax for a particular tenant or user. For such cases, VeloCloud supports the following operator-only APIs to set policies for specific partners and enterprises.

- **enterpriseProxy/insertOrUpdateEnterpriseProxyRateLimits** – Used to configure Partner-specific policies.
- **enterprise/insertOrUpdateEnterpriseRateLimits** – Used to configure Customer-specific policies.

For additional information on the APIs, see *VeloCloud API Guide*.

Chapter 8

Enterprise Deployment and Operations for Orchestrator

This section provides information about the available options to monitor, backup, and upgrade Enterprise On-Premises deployments in a two-day operation scenario.

Even though the enterprise on-premises model has some unique advantages and features, there are considerations that the service provider or customer managing the solution must understand. Some of these considerations are as follows:

- Isolation of the solution- The Arista Cloud Operations team does not have access to apply hotfixes and upgrades.
- Restrictions on change management limit the frequency of patching and upgrades.
- Inadequate or insufficient solution monitoring- This situation may happen due to a lack of personnel capable of managing the infrastructure, resulting in functional issues, slower resolution of problems, and customer dissatisfaction.

This approach always requires a significant investment in people and time to manage, operate, and patch properly. The table below outlines some of the elements that must be considered when managing a system on-premises.

Table 39: Elements to Consider

System	Description	VeloCloud Hosted Responsibility	On-Premises Responsibility
SD-WAN Orchestration	Application QoS and link steering policy	Yes	Yes
	Security policy for apps and SD-WAN appliances	Yes	Yes
	SD-WAN appliance provisioning and troubleshooting	Yes	Yes
	Handling of SD-WAN alerting & events	Yes	Yes
	Link performance and capacity monitoring	Yes	Yes
Hypervisor	Monitoring / alerting	No	Yes
	Compute and memory resourcing	No	Yes
	Virtual networking and storage	No	Yes
	Backup	No	Yes
	Replication	No	Yes
Infrastructure	CPU, memory, compute	No	Yes
	Switching and routing	No	Yes
	Monitoring & management systems	No	Yes
	Capacity planning	No	Yes
	Software upgrades/patching	No	Yes
	Troubleshooting application/ infrastructure issues	No	Yes
Backup and Infrastructure DR	Backup infrastructure	No	Yes
	Regular testing of backup regime	No	Yes
	DR infrastructure	No	Yes
	DR testing	No	Yes

Two-day operation scenarios for Enterprise On-Premises deployments are explained in the two sections below, respectively (Day One Operations and Day Two Operations).

Day One Operations

Deactivating the Cloud-init on the Orchestrator

The data-source contains two sections: meta-data and user-data. Meta-data includes the instance ID and should not change during the lifetime of the instance, while user-data is a configuration applied on the first boot (for the instance ID in meta-data).

After the first boot up, it is recommended to deactivate the cloud-init file to speed up the Orchestrator boot sequence. To deactivate cloud-init, run the following:

```
./opt/vc/bin/cloud_init_ctl -d
```

It is not recommended to purge the cloud-init file with the command `apt purge cloud-init` (this procedure does not cause issues in the VeloCloud SD-WAN Controller). Purging the `cloud-init` file also erases some essential Orchestrator tools and scripts such as upgrade and backup scripts. If the `purge` command was used, you can restore the files using the following commands:

1. Go to the folder `/opt/vcrepo/pool/main/v/vco-tools`.
2. Install the Orchestrator tool package from the folder: `sudo dpkg -i vco-tools_3.4.1-R341-20200423-GA-69c0f688bf.deb`.

   The `vco-tools` package name may change depending on your release. Check the correct file name with the command `ls vco-tools`.

Configuring the NTP Timezone

1. The Orchestrator and Gateway timezone must be set to `Etc/UTC`.

   ```
   vcadmin@vco1-example:~$ cat /etc/timezone Etc/UTC vcadmin@vco1-example:~$
   ```

2. If the timezone is incorrect, it can be corrected by executing the following commands:

   ```
   echo "Etc/UTC" | sudo tee /etc/timezone sudo dpkg-reconfigure --frontend noninteractive tzdata
   ```

The expectation is that the NTP offset is <= 15 milliseconds.

3. Use the following command to check the NTP Offset:

   ```
   vcadmin@vco1-example:~$ sudo ntpq -p remote refid st t when poll reach delay offset jitter
    ============================================================================== *ntp1-us1.pro
   d.v 74.120.81.219 3 u 474 1024 377 10.171 -1.183 1.033 ntp1-eu1-old.pr .INIT. 16 u - 1024 0
    0.000 0.000 0.000 vcadmin@vco1-example:~$
   ```

4. If the offset is incorrect, it can be corrected by executing the following commands:

   ```
   sudo service ntp stop sudo ntpdate <server> sudo service ntp start
   ```

Orchestrator Storage

When the Orchestrator is initially deployed, three partitions are created: `/`, `/store`, `/store2`, `/store3` (version 4.0 and onwards). The partitions are created with default sizes. Follow the instructions in Expand Disk Size for guidance in modifying the default sizes to match the design.

Additional Tasks

The Orchestrator requires further configuration after implementation using the following steps:

1. Configure System Properties.
2. Set up the initial Operator Profile.
3. Set up Operator accounts.
4. Create Gateways.
5. Setup Orchestrator.
6. Create the customer account/partner account.

Detailed instructions can be found in Install VeloCloud Orchestrator.

Day Two Operations

Orchestrator Backup

This section provides the available mechanisms to periodically backup the Orchestrator database to recover from Operator errors or catastrophic failure of both the Active and Standby Orchestrator.

Remember that the Disaster Recovery feature or DR is the preferred recovery method. It provides a Recovery Point Objective of nearly zero, as all configurations on the Active Orchestrator is instantly replicated. For additional details on the Disaster recovery feature, refer to the next section.

Backup Using the Embedded Script

The Orchestrator provides an in-built configuration backup mechanism to periodically Backup the configuration to recover from Operator errors or catastrophic failure of both the Active and Standby Orchestrator. The mechanism is script-driven and is located at `/opt/vc/scripts/db_backup.sh`.

The script essentially takes a database dump of the configuration data and events, while excluding some of the large monitoring tables during the database dump process. Once the script is executed, backup files are created in the local directory path provided as input to the above script.

The Backup consists of two .gzs files, one containing the database schema definition and the other one containing the actual data without definition. The administrator should ensure that the backup directory location has enough disk space for the Backup.

Best Practices

- Mount a remote location and configure the backup script to it. The remote location should have the same storage as /store if flows are also being Backup.
- Before using the Backup Script, check the Disaster Recovery (DR) replication status from the Orchestrator replication page. They should be in sync, and no errors should be present.
- Additional to this, execute a MySQL query and check the replication lag.
  - SHOW SLAVE STATUS \G
  - In the above query, look at the field seconds_behind_master. Ideally, it should be zero, but under 10 would be sufficient as well.
  - For the large Orchestrators, it is recommended to use the Standby for the Backup script execution. There will be no difference in the Backup that is generated from both Orchestrators.

Caveats

- The Script only takes a backup of the configuration; flow stats or events are not included.
- Restoring the configuration requires assistance from the Support/Engineering team.

The Backup consists of two .gzs files, one containing the database schema definition and the other one containing the actual data without definition. The administrator should ensure that the backup directory location has enough disk space for the Backup.

Frequently Asked Questions

- *How long does the Script take to run?*

  The duration of the Backup depends on the scale of the actual customer configuration. Since the monitoring tables are excluded from the Backup operation, it is expected that the configuration Backup operation will complete quickly. For a large Orchestrator with thousands of Edge and lots of historical events, it could take up to an hour, while a smaller Orchestrator should be completed within a few minutes.

- *What is the recommended frequency to run the Backup script?*

  Depending on the size and time it takes to complete the initial backup, the Backup operation frequency can be determined. The Backup operation should be scheduled to run during off-peak hours to reduce the impact on Orchestrator resources.

- *What if the root file system doesn't have enough space for the backup?*

  It is recommended that other mounted volumes are used to store the backup. Note, it is not a best practice to use the root filesystem for the backup.

- *How does one verify if the Backup operation completed successfully?*

  The script `stdout` and `stderr` should be sufficient to determine the success or failure of the Backup operation. If the script invocation is automated, the exit code can determine the Backup operation's success or failure.

- *How is the configuration recovered?*

  Currently, Arista requires that the customer work with Arista Support to recover the configuration data. Arista Support will help to recover the customer's configuration. Customers should refrain from making any additional configuration changes until the configuration is restored.

- *What is the exact impact of executing this Script?*

  Even though a backup of the configuration should have little impact on performance, there will be an increase in resource utilization for the MySQL process. It is recommended that the Backup be run during off-peak hours.

- *Are any configuration changes allowed during the run of the Backup operation?*

  It is safe to make configuration changes while the Backup operation is running. However, to ensure up-to-date backups, it is recommended that no configuration operations are done while the Backup is running.

- *Can the configuration be restored on the original Orchestrator, or does it require a new Orchestrator?*

  Yes, the configuration can, and ideally should, be restored on the same Orchestrator if it is available. This will ensure that the monitoring data is utilized after the Restore operation is completed. If the original Orchestrator cannot be recovered and the Standby Orchestrator is down, the configuration can be restored on a new Orchestrator. In this instance, the monitoring data will be lost.

- *What actions should be taken in case the configuration needs to be restored to a new Orchestrator?*

Please contact Arista Support for the recommended set of actions on the new Orchestrator as the steps vary depending on the actual deployment.

- *Do Edges have to re-register on the newly restored Orchestrator?*

  No, Edges are not required to register on the new Orchestrator, as all needed information is preserved as part of the Backup.

Orchestrator Disaster Recovery

The Orchestrator Disaster Recovery (DR) feature prevents the loss of stored data and resumes Orchestrator services in the event of system or network failure. Orchestrator DR involves setting up an Active/Standby Orchestrator pair with data replication and a manually-triggered failover mechanism.

Note: DR is mandatory. For licensing and pricing, contact the Arista SD-WAN Sales team for support.

States

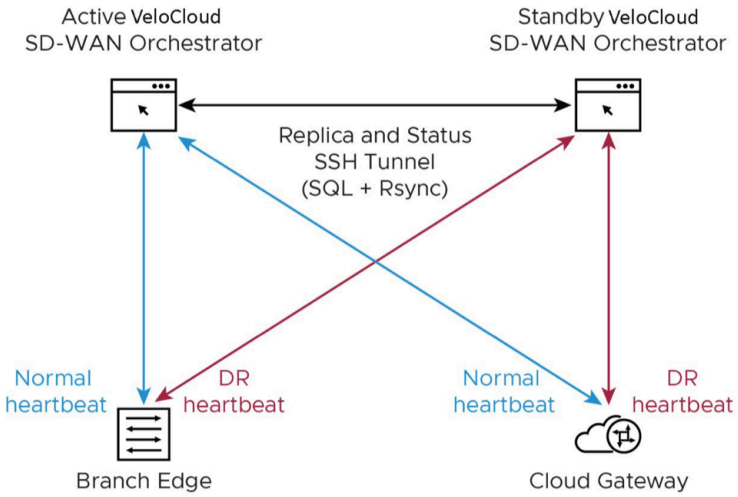
From the view of an Operator, and of the SD-WAN Edges and SD-WAN Gateways, an Orchestrator has one of four DR states:

- Standalone (no DR configured)
- Active (DR configured, acting as the primary Orchestrator server)
- Standby (DR configured, acting as an inactive replica Orchestrator server)
- Zombie (DR formerly configured and Active, but no longer working as the Active or Standby)

Table 40: DR States/Phases

Phases	VeloCloud Orchestrator A Role	VeloCloud Orchestrator B Role
Initial	Standalone	Standalone
Pairing	Active	Standby
Failover	Zombie	Standalone

Figure 8-1: Orchestrator DR Use Case Scenario

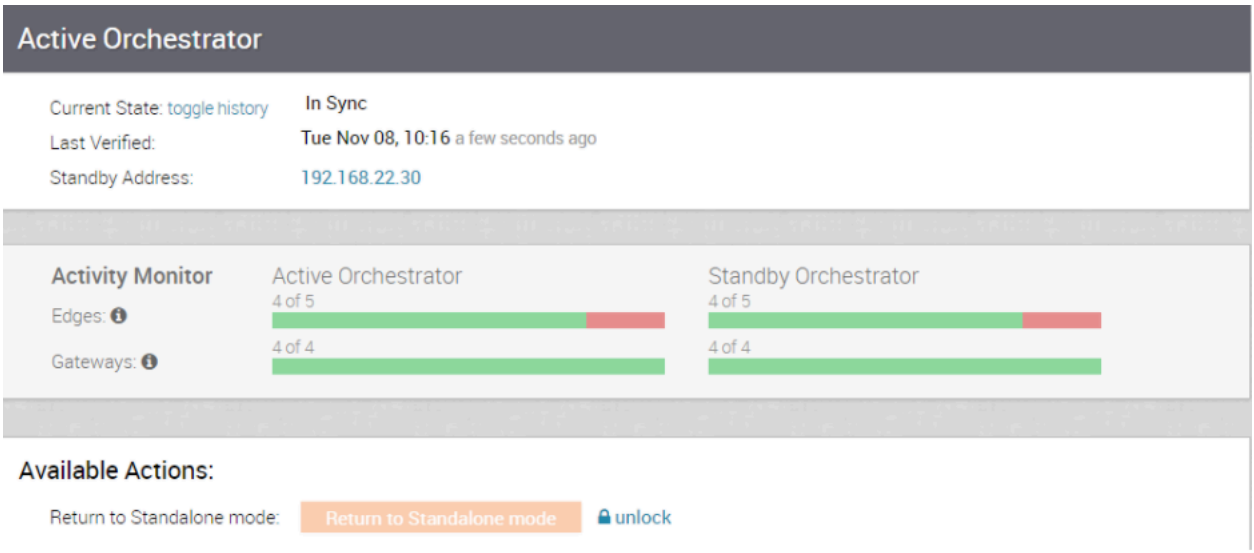


Best Practices

- Locate the Orchestrator DR in a geographically separate datacenter.

- Before promoting a Standby Orchestrator as Active, confirm that the DR replication Status is in Sync. The previously Active Orchestrator will no longer be able to manage the inventory and configuration.

Figure 8-2: Active Orchestrator



- If the Standby can communicate with the formerly Active Orchestrator, it will instruct that Orchestrator to enter a Zombie state. In the Zombie state, the Orchestrator communicates with its clients (SD-WAN Edges, SD-WAN Gateways, UI/API) that it is no longer Active, and they must communicate with the newly promoted Orchestrator.
- If the promoted Standby cannot communicate with the formerly Active Orchestrator, the Operator should, if possible, manually demote the previously Active.

See Set Up Orchestrator Replication.

Upgrade Procedure for the Orchestrator

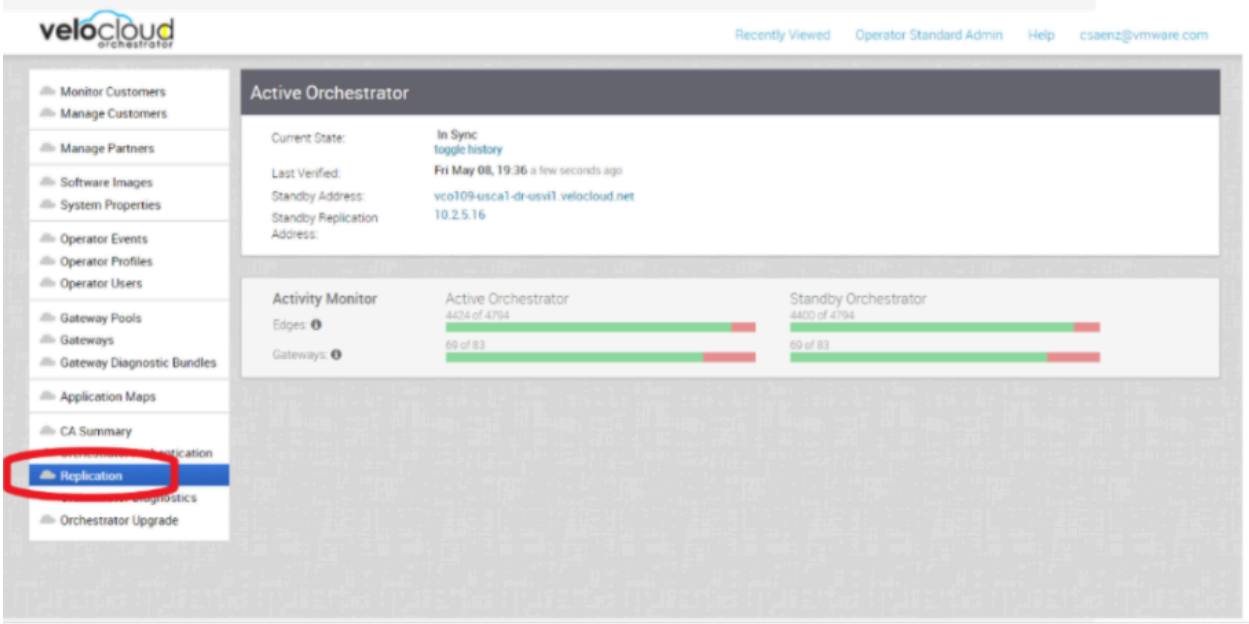
For Enterprise on-prem deployments, contact the Arista Support team to prepare for the Orchestrator upgrade

1. Arista Support assists with the upgrade. Collect the following information before contacting Arista Support.
2. Provide the current and target Orchestrator versions, for example, the current version (i.e., 3.4.2), target version (3.4.3).

> **Note:** For the current version, this information can be found on the top, right corner of the Orchestrator by selecting the Help link and choosing About.

3. Provide a screenshot of the replication dashboard of the Orchestrator.

Figure 8-3: Replication Dashboard



4. Hypervisor Type and version (i.e., vSphere 6.7)
   Commands from the Orchestrator (Commands must be run as root (e.g. **`sudo <command>`** or **`sudo -i`**).):
   - LVM layout
     - `pvdisplay-v`
     - `vgdisplay-v`
     - `lvdisplay-v`
     - `df-h`
     - `cat /etc/fstab`
   - Memory information
     - `free-m`
     - `cat /proc/meminfo`
     - `ps-ef`
     - `top-b-n 2`
   - CPU Information
     - `cat /proc/cpuinfo`
   - Copy of `/var/log`
     - `tar-czf /store/log-`date +%Y%M%S`.tar.gz--newer-mtime="36 hours ago" /var/log`
   - From the Standby Orchestrator:

- `sudo mysql--defaults-extra-file=/etc/mysql/velocloud.cnf velocloud-e 'SHOW SLAVE STATUS \G'`

- From the Active Orchestrator:
  - `sudo mysql--defaults-extra-file=/etc/mysql/velocloud.cnf velocloud-e 'SHOW MASTER STATUS \G'`

5. Contact Arista VeloCloud SD-WAN Support with the above-mentioned information for assistance with the Orchestrator upgrade.

Controller Minor Software Upgrade (Ex. from 3.3.2 P3 to 3.4.4)

The software upgrade file contains Gateway and system updates. Do NOT run `apt-get update && apt-get –y upgrade`.

Before proceeding with the SD-WAN Controller's upgrade, ensure that the Orchestrator was upgraded before to the same or a higher version.

To upgrade an SD-WAN Controller:

1. Download the SD-WAN Controller update package.
2. Upload the image to the SD-WAN Controller storage (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 SD-WAN Controller console and run:

   `sudo /opt/vc/bin/vcg_software_update`

   Example:

```
root@VCG:/var/lib/velocloud/software_update# wget -O 'vcg_update.tar' <image location>
 Resolving ftpsite.vmware.com (ftpsite.vmware.com)... Connecting to ftpsite.vmware.com
 (ftpsite.vmware.com)| <ip address>|:443... connected. HTTP request sent, awaiting response...
 200 OK Length: unspecified [application/octet-stream] Saving to: 'vcg_update.tar' [ <=> ]
 325,939,200 3.81MB/s in 82s 2020-05-23 21:59:27 (3.79 MB/s) - ‘vcg_update.tar’ saved
 [325939200] root@VCG:/var/lib/velocloud/software_update# sudo /opt/vc/bin/vcg_software_update
 ========== VCG upgrade: Sat May 23 22:08:15 UTC 2020 Upgrading gateway version 3.4.0-106-
R340-20200218-GA-c57f8316dd to 3.4.1-39-R341-20200428-GA-44354-44451-596496a88a Ign file:
 trusty InRelease Ign file: trusty Release.gpg Get: 1 file: trusty Release [2,668 B] Ign file:
 trusty/main Translation-en_US Ign file: trusty/main Translation-en (...) Writing extended
 state information... Reading package lists... Building dependency tree... Reading state
 information... Reading extended state information... Initializing package states... update-
initramfs: Generating /boot/initrd.img-3.13.0-176-generic Reboot is required. Reboot? (y/n)
 [y]:
```

Controller major software upgrade (Ex from 3.3.2 or 3.4 to 4.0)

In version 4.0, multiple changes are included:

- A new system disk layout based on LVM to allow additional flexibility in volume management
- A new kernel version
- New and upgraded base OS packages
- Improved security hardening based on the Center for Internet Security benchmarks

Due to these changes, the standard upgrade procedure which uses the upgrade script does not work. A particular upgrade procedure is required. It is in the product manual below. This procedure is to replace the 3.3.2 or 3.4 Gateway VM with the new 4.0 Gateway VM. Refer to the following document: *VeloCloud SD-WAN Partner Gateway Upgrade and Migration 3.3.2 or 3.4 to 4.0*.

This upgrade procedure requires Orchestrator system property configuration, which only Orchestrator Operator accounts can run. Create a support ticket with the Arista VeloCloud Support team to request the System Property change.

Monitoring

One of the customer's responsibilities on enterprise On-Prem deployments is to monitor the solution. Monitoring gives customer's the visibility required to be one step ahead of possible issues.

SD-WAN Controller Monitoring

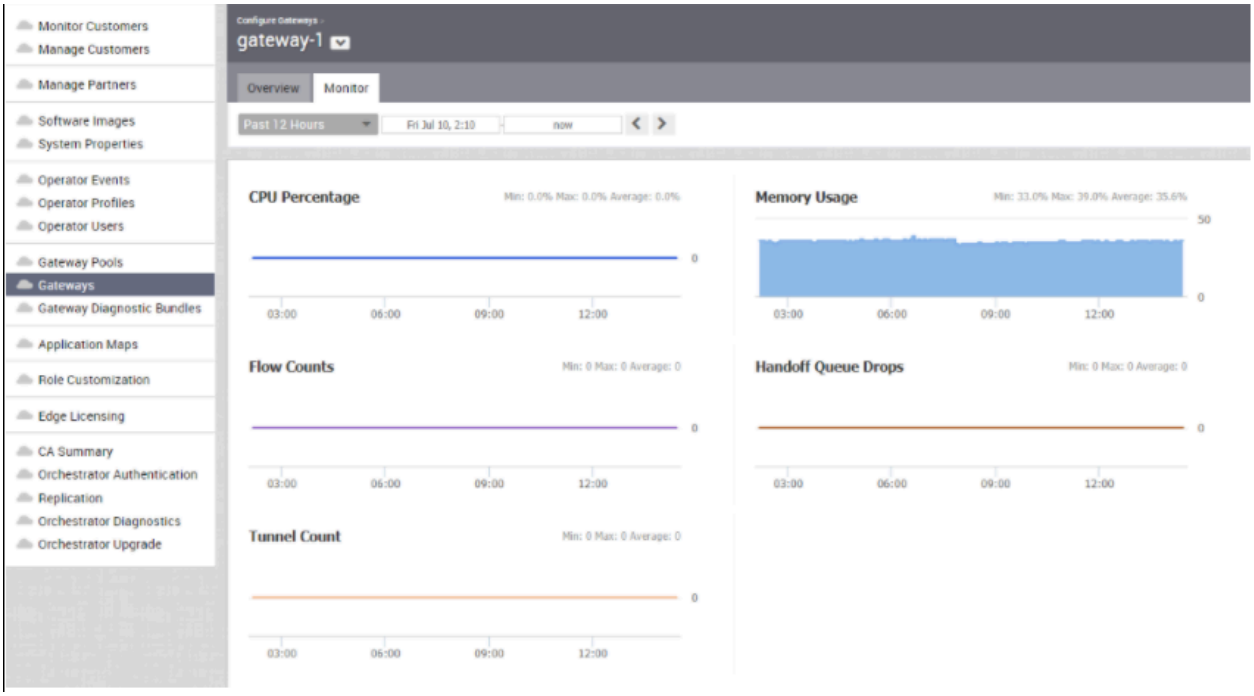
You can monitor the status and usage data of Controllers available in the Operator portal.

The procedure is as follows:

1. In the **Operator** portal, select **Gateways**.
2. The **Gateways** page displays the list of available Controllers.
3. Select the link to a Gateway. The details of the selected Controller displays.
4. Select the **Monitor** tab to view the usage data of the selected Controller.

The **Monitor** tab of the selected Controller displays the following details:

Figure 8-4: Monitor Controller



You can choose a specific period to view the Controller details for the selected duration at the top of the page.

The page displays a graphical representation of usage details of the following parameters for the period of selected time duration, along with the minimum, maximum, and average values.

Table 41: Controller Usage Details

Usage	Description
CPU Percentage	Percentage of usage of CPU
Memory Usage	Percentage of usage of memory
Flow Counts	Count of traffic flow
Handoff Queue Drops	Count of packets dropped due to queued handoff
Tunnel Count	Count of tunnel sessions

SD-WAN Gateway Controller Recommended Values to Monitor

The following list shows values that should be monitored and their thresholds. The list below is given as a start point, and it is not exhaustive. Some deployments may require assessing additional components such as flows, packet loss, etc.

Whenever a warning threshold is reached, it is recommended to review the current device scale configuration and add additional resources if required. When a critical alarm is triggered, it is crucial to contact Arista Support representatives to check the solution and provide further advice.

Table 42: Service Check Details

Service Check	Service Check Description	Warn Threshold	Critical Threshold
CPU Load	Check System Load.	60	80
Memory	Checks the memory utilization buffer, cache, and used memory.	70	80
Tunnels	Number of tunnels from connected Edges.	60% of max Scale	80% of max Scale Note: A sudden loss of all tunnels or an abnormal low quantity should also be a concern.
Handoff Drops	Due to the busy nature of traffic through a Controller, occasional drops are expected.	Consistent drops in specific queues may indicate a capacity problem.	
Disk Space	Current disk utilization	40% Free	20% Free
Controller NTP	Check for Time offset	Offset of 5 Seconds	Offset of 10 Seconds

Orchestrator Integration with Monitoring Stacks

The Orchestrator comes with a built-in system metrics monitoring stack, which can attach to an external metrics collector and a time-series database. With the monitoring stack, you can quickly check the health condition and the system load for the Orchestrator.

Before getting started, set up a time-based database and a dashboard/alerting agent. After this is complete, you can enable Telegraf in Orchestrator.

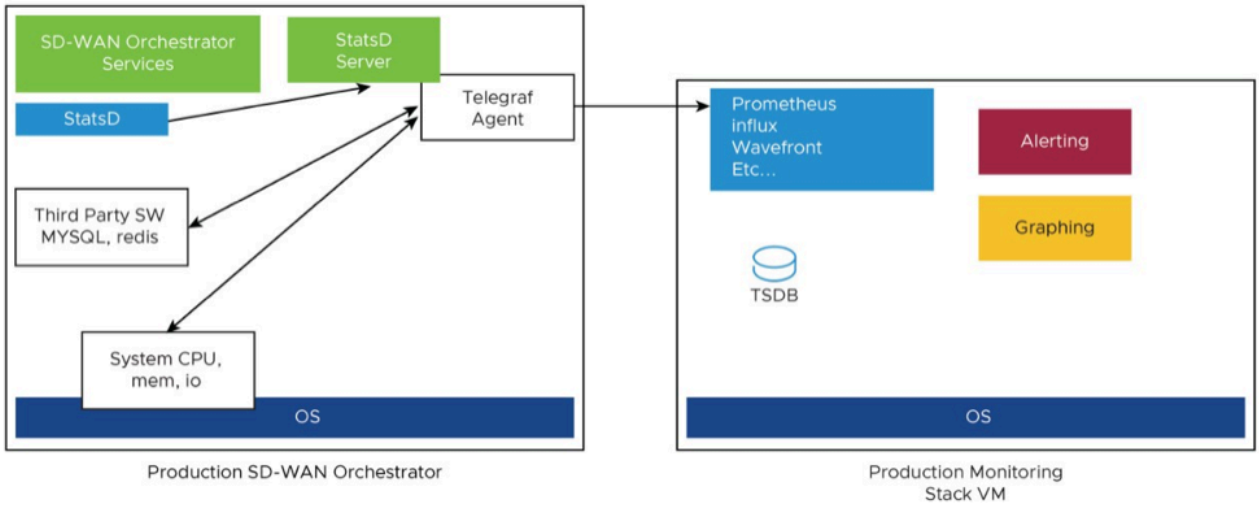
To enable the monitoring stack, run the following command on the orchestrator: `sudo /opt/vc/scripts/vco_observability_manager.sh enable`

To check the status of the monitoring stack, run: `sudo /opt/vc/scripts/vco_observability_manager.sh status`

To deactivate the monitoring stack, run:

```
sudo /opt/vc/scripts/vco_observability_manager.sh disable
```

Figure 8-5: Monitoring Stacks Topology



The Metrics Collector Telegraf is used as the Orchestrator system metrics collector with plugins to collect different system metrics. The following metrics are enabled by default.

Table 43: Metrics Collector

Metric Name	Description	Supported in Version
inputs.cpu	Metrics about CPU usage.	3.4/4.0
inputs.mem	Metrics about memory usage.	3.4/4.0
inputs.net	Metrics about network interfaces.	4.0
inputs.system	Metrics about system load and uptime.	4.0
inputs.processes	The number of processes grouped by status.	4.0
inputs.disk	Metrics about disk usage.	4.0
inputs.diskio	Metrics about disk IO by device.	4.0
inputs.procstat	CPU and memory usage for specific processes.	4.0
inputs.nginx	Nginx's basic status information (ngx_http_stub_status_module).	4.0
inputs.mysql	Statistic data from MySQL server.	3.4/4.0
inputs.redis	Metrics from one or many redis servers.	3.4/4.0
inputs.statds	API and system metrics.	3.4/4.0 (additional metrics are included in 4.0)
inputs.filecount	The number and the total size of files in specified directories.	4.0
inputs.ntpq	Standard NTP query metrics, requires ntpq executable.	4.0
Inputs.x509_cert	Metrics from a SSL certificate.	4.0

To activate additional metrics or deactivate some enabled metrics, you can edit the Telegraf configuration file on the Orchestrator using the following commands:

```
sudo vi /etc/telegraf/telegraf.d/system_metrics_input.conf
```

```
sudo systemctl restart telegraf
```

- **Time-series Database**- A time Series Database can be used to store the system metrics collected by Telegraf. A time-series database (TSDB) is a database optimized for time series data.
- **Dashboard and Alerting Agent**- allows you to query, visualize, alert, and explore the data stored in the TSDB. The following image provides an example of a dashboard using Telegraph, a TSDB and a dashboard engine, created to monitor the solution.

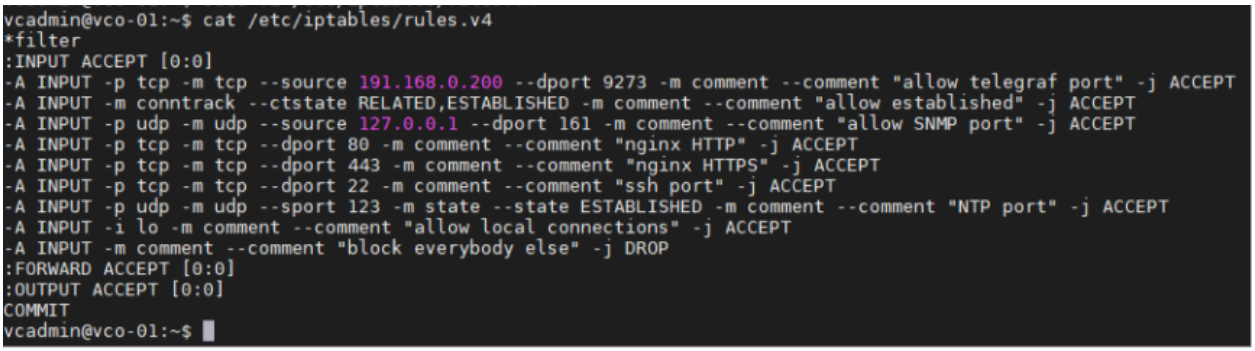
  Figure 8-6: Dashboard

Follow the instructions below to setup the time-series database.

1. Add the iptables entry to allow for external monitoring systems to access to Telegraf port. The source IP address should be specified for security reasons.

The IP address of the external monitoring system is `191.168.0.200 Add "-A INPUT-p tcp-m tcp--source 191.168.0.200--dport 9273-m comment--comment "allow telegraf port"-j ACCEPT" to /etc/iptables/rules.v4`.

Figure 8-7: Adding ports



2. Restart `iptables`.

   (Orchestrator 3.4.x)

   ```
   sudo service iptables-persistent restart
   ```

   (Orchestrator 4.x)

   ```
   sudo systemctl restart netfilter-persistent
   ```

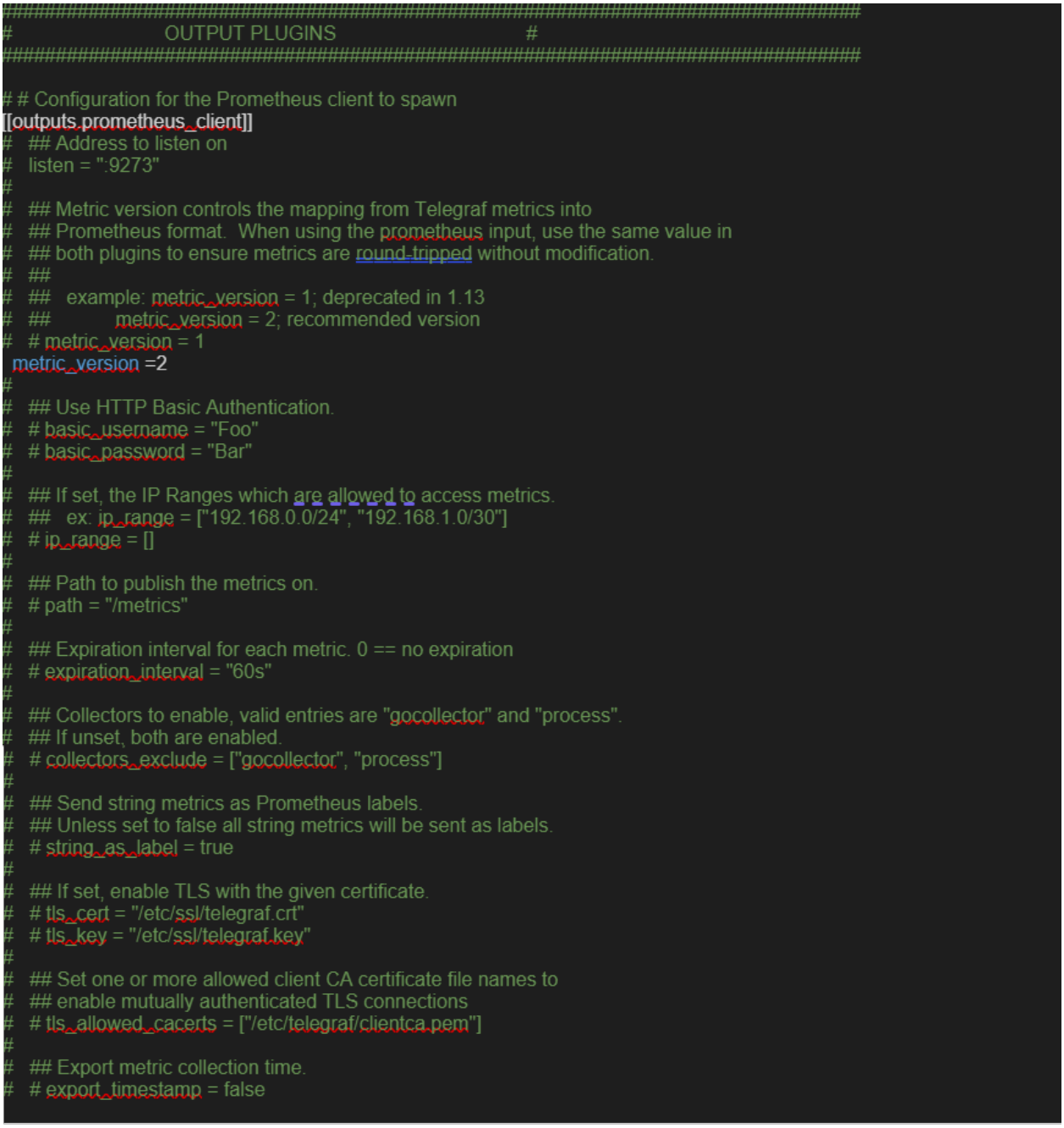
3. Ensure the `iptables` entry updated.

   Figure 8-8: IP Tables

4. Add the time-series database details in the Telegraf configuration. Create an output configuration file. For example, Prometheus uses the following:

```
/etc/telegraf/telegraf.d/prometheus_out.conf
```

Figure 8-9: Prometheus Example



Monitor Values and Thresholds

The following list shows a list of values that should be monitored and their thresholds. The list below is given as a starting point, as it is not exhaustive. Some deployments may require assessing additional components such as database transactions, automatic backups, etc.

Whenever a warning threshold is reached, it is recommended to review the current device scale configuration and add additional resources if required. When a critical alarm is triggered, it is crucial to contact the Arista Support representatives to check the solution and give further advice.

Table 44: Service Checks

Service Check	Service Check Description	Warn Threshold	Critical Threshold
CPU Load	Check System Load – Telegraf input plugin: inputs.cpu.	60	70
Memory	Checks the memory utilization buffer, cache, and used memory – Telegraf input plugin: inputs.memory.	70	80
Disk Usage	Disk Utilization in the different Orchestrator partitions, /, /store, /store2 and /store3 (version 4.0 and onwards) – Telegraf input plugin: inputs.disk (version 4.0 and onwards).	40% Free	20% Free
MySQL Server	Checks MySQL Connections-Telegraf input plugin: inputs.mysql.		Above 80% of max connection define in mysql.conf(/etc/mysql/my.cnf)
Orchestrator Time	Check for Time offset-Telegraf input plugin: inputs.ntpq (version 4.0 and onwards).	Offset of 5 Seconds	Offset of 10 Seconds
Orchestrator SSL Certificate	Checks Certificate Expiration-Telegraf input plugin: inputs.x509_cert (version 4.0 and onwards).	60 Days	30 Days
Orchestrator Internet (not applicable for MPLS only topologies)	Check for Internet access.	Response time > 5 secs	Response time > 10 secs
Orchestrator HTTP	Make sure HTTP on localhost is responding.		The localhost is not responding.
Orchestrator Total Cert Count	Check Total – Example mysql query: SELECT count(id) FROM VELOCLOUD_EDGE_CERTIFICATE WHERE validFrom <= NOW() AND validTo >=NOW()', 'SELECT count(id) FROM VELOCLOUD_GATEWAY_CERTIFICATE WHERE validFrom <= NOW() AND validTo >=NOW()	CRL	When Total Cert count exceeds 5000
DR Replication Status	Confirm the Standby Orchestrator is up-to-date.	Review that the DR Orchestrator is no more than 1000 seconds behind the Active Orchestrator. Seconds_Behind_Master: from mysql command: show slave STATUS\G;	
DR Replication Edge Gateway delta	Confirm that Edges and Gateways can talk to the DR Orchestrator. Different values between the Active and the Standby Orchestrators can be due to a difference in the timezone in Edges and Gateways.	The same amount of Edges talking with the Active Orchestrator should be able to reach the Standby Orchestrator. This value can be checked on the "replication" tab or via the API.	

API Best Practices

Orchestrator powers the management plane in the VeloCloud SD-WAN solution. It offers a broad range of configuration, monitoring, and troubleshooting functionality to service providers and enterprises. The main web service with which users interact to exercise this functionality is called the Orchestrator Portal.

Orchestrator Portal- The Orchestrator Portal allows network administrators (or scripts and applications acting on their behalf) to manage network and device configuration and query the current or historical network and device state. API clients may interact with the Portal via a JSON-RPC interface or a REST-like interface. It is possible to invoke all of the methods described in this document using either interface. There is no Portal functionality for which access is constrained exclusively to either JSON-RPC clients or REST-like ones.

Both interfaces accept exclusively HTTP POST requests. Both also expect that request bodies, contain JSON-formatted content consistent with RFC 2616. Clients are furthermore likely to formally assert where this is the case using the Content-Type request header, e.g., Content-Type: application/json.

Additional information about the VeloCloud SD-WAN API can be found here:

https://code.Arista.com/apis/1000/velocloud-sdwan-vco-api

Best Practices for enterprises and service providers using APIs- Consider the following best practices while using APIs:

- Wherever possible, aggregate API calls should be preferred to enterprise-specific ones, for example, a single call to `monitoring/getAggregateEdgeLinkMetrics` may be used to retrieve transport stats across all Edges concurrently.
- VeloCloud requests that clients limit the number of API calls in flight at any given time to no more than 2-4. If a user feels there is a compelling reason to parallelize API calls, Arista requests that they contact Arista Support to discuss alternative solutions.
- Arista doesn't recommend polling the API for stats data more frequently than every 10 min. New stats data arrives at the Orchestrator every 5 minutes. Due to jitter in reporting/processing, clients polling every 5 minutes might observe "false-positive" cases where stats aren't reflected in API calls' results. You might get the best result using request intervals of 10 minutes or greater in duration.
- Avoid querying the same information twice.
- Use sleep between APIs.
- For complex software automations, run your scripts and evaluate the CPU/Memory impact. Then adjust as required.

Orchestrator Syslog Configuration

The VeloCloud Orchestrator Syslog capability can be configured independently for the following Orchestrator processes:

- **Portal:** The Portal process runs as an internal HTTP server downstream from NGINX. The Portal service handles incoming API requests, either from the Orchestrator web interface or from an HTTP/SDK client, primarily in a synchronous fashion. These requests allow authenticated users to configure, monitor, and manage the various services provided by the Orchestrator.

  This log is very useful for AAA activities as it has all actions taken by users in the Orchestrator.

Log files: `/var/log/portal/velocloud.log` (Logs all info, warn, and error logs)

- **Upload:** The Upload process runs as an internal HTTP server downstream from NGINX. The Upload service handles incoming requests from Edges and Gateways, either synchronously or asynchronously. These requests primarily consist of activations, heartbeats, flow statistics, link statistics, and routing information sent by Edges and Gateways.

  Log files: `/var/log/upload/velocloud.log` (Logs all info, warn, and error logs)

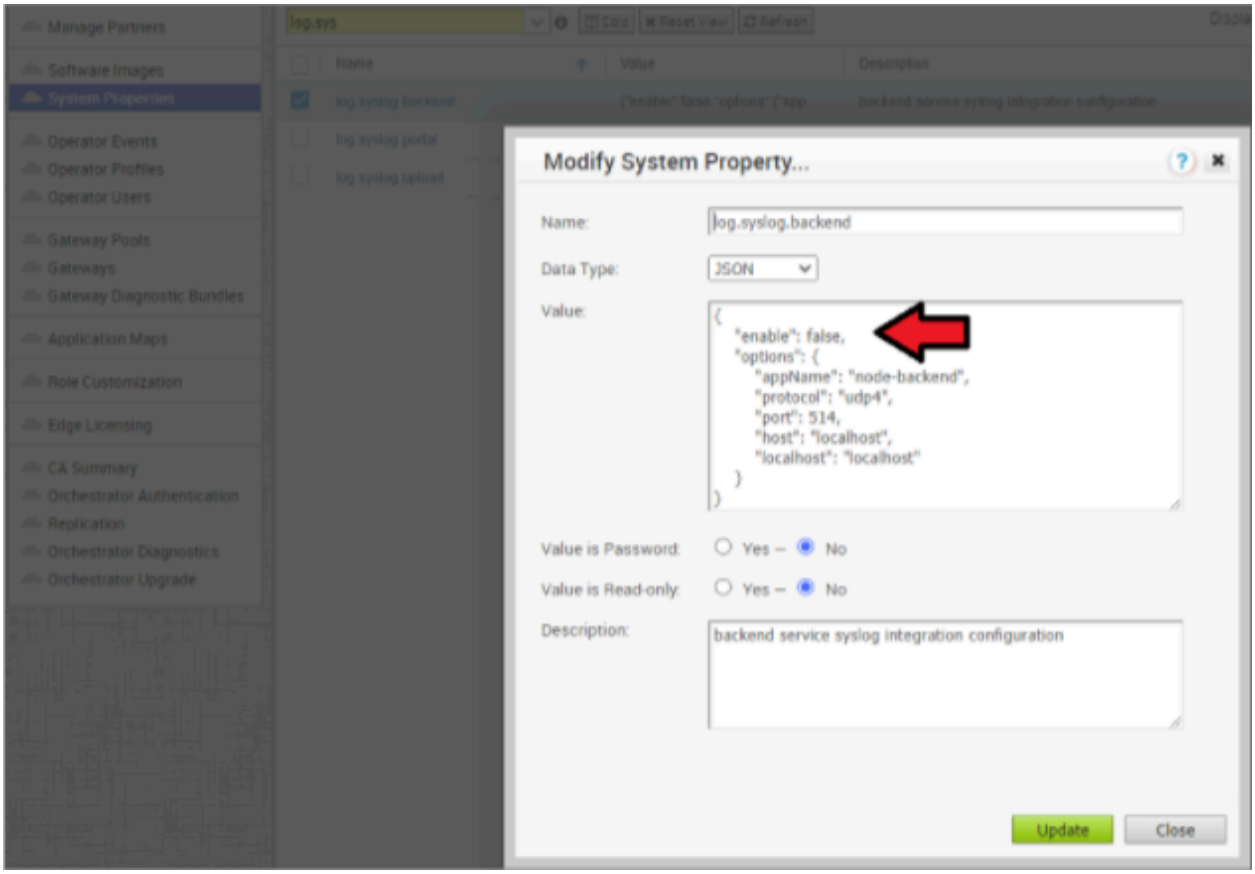
- **Backend:** Job runner that primarily runs scheduled or queued jobs. Scheduled jobs consist of cleanup, rollup, or status update activities. Queued jobs consist of processing link and flow statistics.

  Log files: `/var/log/backend/velocloud.log` (Logs all info, warn, and error logs)

Use the following steps to configure Orchestrator Syslog:

1. Navigate to **System Properties** in **Orchestrator** > **System Properties** and enter log.syslog in the search bar.
2. Change the `enable:false` value to true for one or more of the servers. Change the Host IP and port accordingly to your implementation.

   Figure 8-10: Modify System Property



Increasing Storage in the Orchestrator

For detailed instructions to increase the Storage in the Orchestrator, see the topics *Install SD-WAN Orchestrator* and *Expand Disk Size (Arista)*.

Best Practices

- Ensure the same LVM distribution applies to the Standby Orchestrator.
- It is not recommended to reduce the size of the volumes once increased. Use thin provisioning instead.
- In 3.4, when increasing the disk size, the following percentage/value distribution may be used:
  - `/` Volume: This volume is used for the operative system. Production Orchestrators are usually set to 140GBs and have from 40% to 60% usage.
  - `/store` and `/Store2`: The proportion applied in production Orchestrators is close to 85% for /Store and 15% for /Store2.

The following guidelines in the table below should be used in the 4.x release and onwards.

Table 45: Storage

Instance Size	/store	/store2	/store3	/var/log
Small (5000 Edges)	2 TB	500 GB	8 TB	100 GB
Medium (10000 Edges)	2 TB	500 GB	12 TB	125 GB
Large (15000 Edges)	2 TB	500 GB	16 TB	`150 GB

Managing Certificates in the Orchestrator

Orchestrator uses a built-in certificate server to manage the overall PKI lifecycle of all Edges and SD-WAN Controllers. X.509 certificates are issued to the devices in the network.

Detailed instructions to configure the CA can be found in the official VeloCloud SD-WAN Operator documentation under "Install Orchestrator" and "Install an SSL Certificate.".

Certificates issued by the CA are used only for the authentication of the following:

- Management plane TLS 1.2 tunnels between the Orchestrator and Edge SD-WAN Controller.
- Control and Data plane IKEv2/IPsec tunnels between SD-WAN Edges and between Edge and SD-WAN Controller.

Certificate Revocation List

On Controllers with PKI enabled, revoked certificates are stored in a Certificate Revocation List (CRL). If this list grows too long, generally due to an issue with the Orchestrator Certificate Authority, the Controller's performance becomes impacted. The CRL should be less than 4,000 entries long.

```
vcadmin@vcg1-example:~$ openssl crl -in /etc/vc-public/vco-ca-crl.pem -text | grep 'Serial Number'
 | wc -l 14 vcadmin@vcg1-example:~
```

Support Interaction

Our Customer Support organization provides 24x7x365 world-class technical assistance and personalized guidance to VeloCloud SD-WAN customers.

This section provides some guidelines to interact with the Arista Support team.

- Diagnostic Bundles

  While investigating an incident, a diagnostic bundle of the Orchestrator and SD-WAN Controller can be created. The resulting file will assist the Arista Support team to further analyze the events around an issue.

Figure 8-11: Gateway Diagnostic Bundles

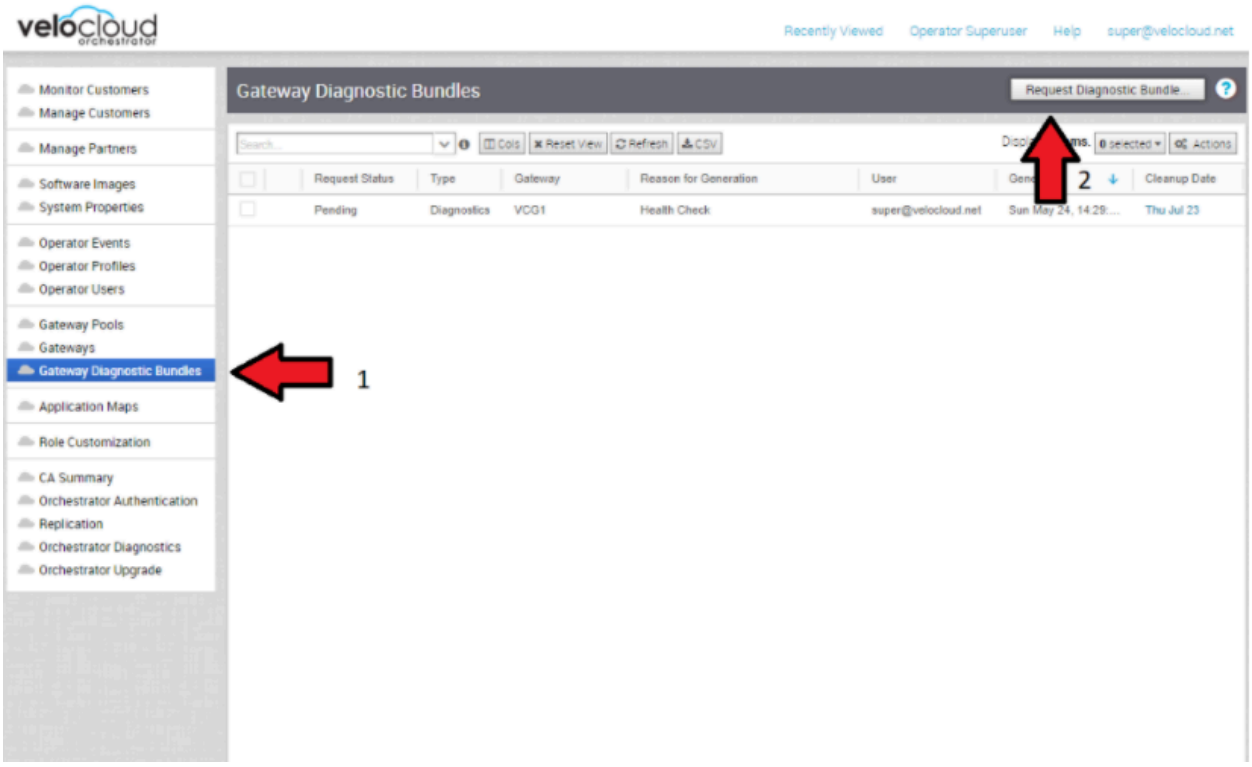
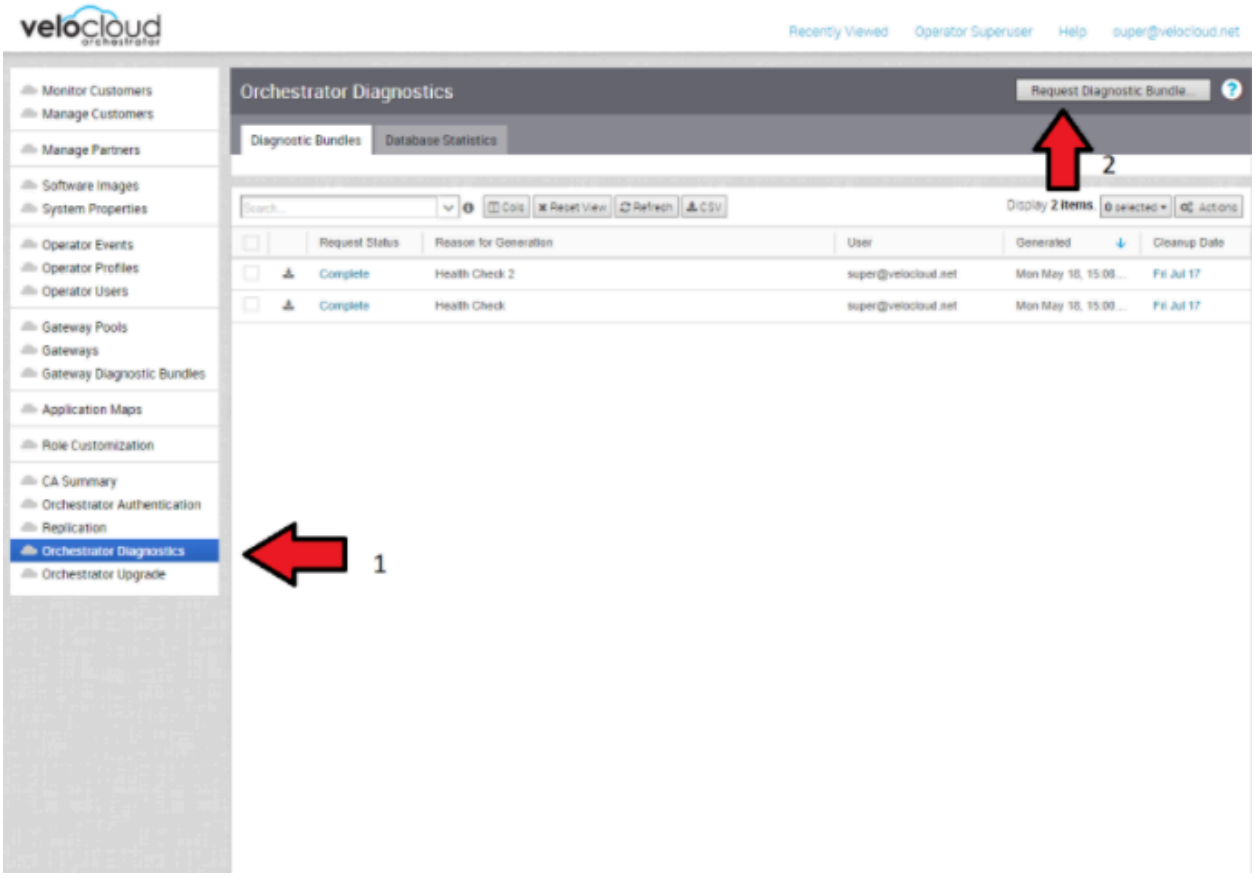


Figure 8-12: Request Diagnostic Bundles



- Share Access with Support

  On occasion assistance from Arista Support representatives for the Orchestrator and SD-WAN Controllers may be required.

  Some common ways to grant access are:

  - Remote sessions with Support: The customer would either grant remote control to the SSH jump server or follow the Support representative's instructions.
  - Creating an account for the Support team in the Orchestrator. This helps the Support team gather logs without customer interaction.
  - Through the Bastion Host: SSH permissions and keys can be configured to allow the Support engineers to access the on-premises Orchestrator and SD-WAN Controller using a Bastion Host.

  When contacting Arista SD-WAN Support to assist triaging an issue, include the data described in the table below.

Table 46: Arista SD-WAN Support Required Information

Required	Suggested
Partner Case Number	Issue Start/Stop
Partner Return Email/Phone	Impacted Flow SRC/DST IP
Orchestrator URL	Impacted Flow SRC/DST Port
Customer Name in Orchestrator	Flow Path (E2E, E2GW, Direct)
Customer Impact (High/Med/Low)	SD-WAN Gateway Name(s)
Edge Name(s)	Link to PCAP in the Orchestrator
Link to Diagnostic Bundle in Orchestrator	
Short Problem Statement	
Analysis & Requested Assistance	

Chapter 9

On Premises SD-WAN Deployment Design and Configuration Guide

This section explains how to deploy and operate an on-premises Arista VeloCloud SD-WAN solution, including the Orchestrator.

Overview

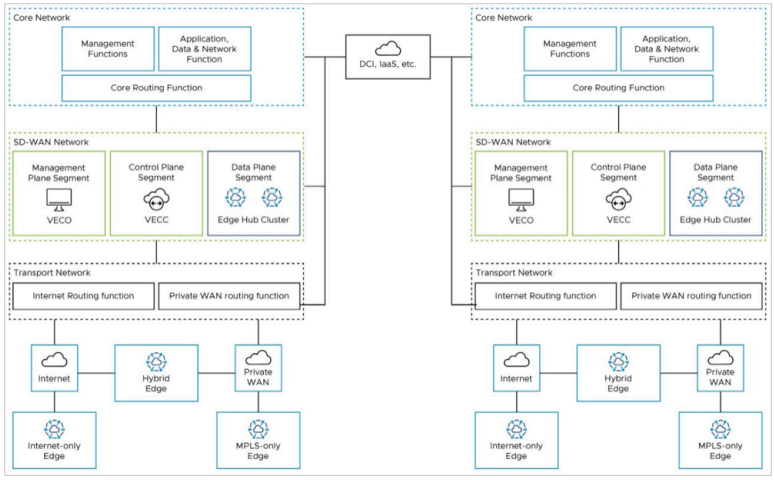
This section explains how to deploy and operate an on-premises Arista VeloCloud SD-WAN solution, which includes the VeloCloud Orchestrator, the VeloCloud Controller, and a co-located Edge Hub Cluster. It includes a reference architecture and the requirements and caveats for an on-premises SD-WAN deployment. It also covers the optional use of an External Certificate Authority and FIPS mode.

Reference Architecture

The reference architecture shows the logical grouping of the different VeloCloud SD-WAN and network functions. It also shows how the different nodes connect and communicate with each other. We expect that most real-world deployments would adapt the reference architecture in some way to accommodate their existing customer network.

The Reference Architecture is a conceptual diagram representing the solution's different functional blocks. These functions may be combined, split from one another, separated by firewalls and/or network segmentation, and so on, as long they maintain the fundamental ability to communicate with one another.

Figure 9-1: Reference Architecture Diagram



Note: In the above diagram, VECO stands for the VeloCloud Edge Cloud Orchestrator, and VECC stands for the VeloCloud Edge Cloud Controller.

Core Network

The Core Network has applications and resources that users need to access to achieve their business goals. It may also have management functions, such as network monitoring and operations. If you use an External Certificate Authority (ECA), it may reside here. Data Center Interconnect (DCI, if you have it) also ends here from a routing perspective. You may separate these functions into different logical network segments.

SD-WAN Network

The SD-WAN Network is between the Transport Network and the Core Network. It has the Orchestrator (management plane), the Controller (control & routing plane), and the on-premises Hub Cluster (data plane), if you have one. These are all in the SD-WAN Network, but you may separate the Hub Cluster from the Orchestrator and Controller logically. This way, you can keep the SD-WAN management and control-plane traffic away from the branch data-plane traffic that uses the Hub Cluster to reach the core network resources.

Transport Network

The Transport Network has the WAN transport functions in the network, such as Public WAN (internet) and Private WAN (MPLS).

The Transport Network also has the Wide Area Network routing functions. This is where you would find Public WAN (internet) routers and Private WAN (MPLS CE) routers in an on-prem customer network. You may do NAT between public and private IP addresses here, or on an Edge firewall, depending on your network setup. You can also use wireless WAN (5G, for example) in the Public WAN case.

Firewalls

While not explicitly shown in the reference architecture, firewalls may be present. They may be deployed between the different functional blocks to provide security and traffic inspection or may be used to create different segments or zones within a functional block.

Packet Flows

This section illustrates the packet flow path in the reference architecture for various operations.

Edge Activation

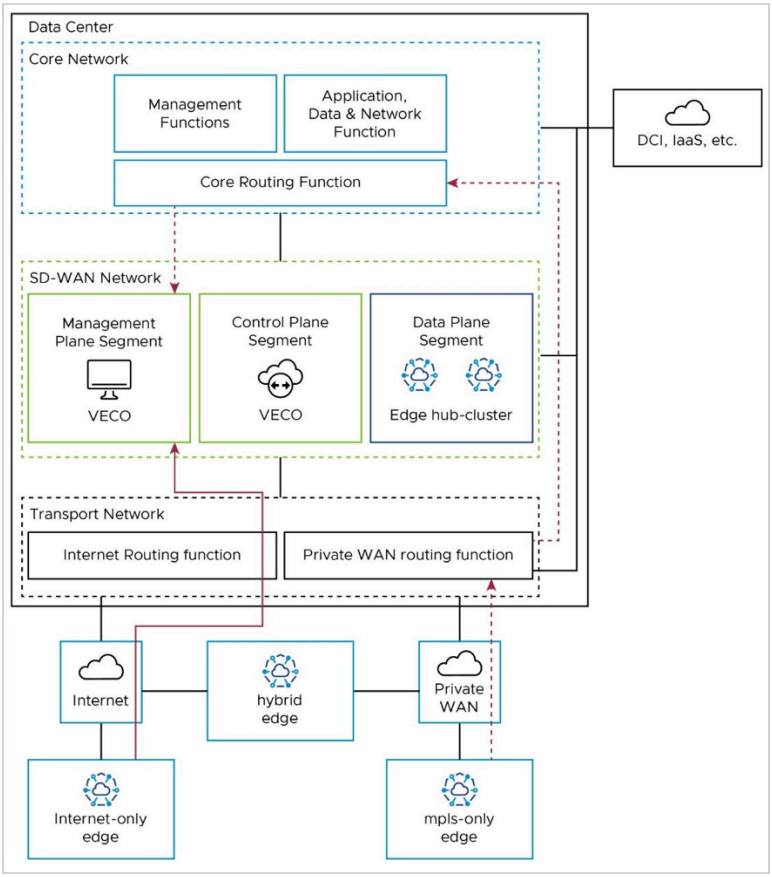
An internet-only branch location activates via a public WAN through the Internet Routing Function in the Transport Network to the public IP address of the Orchestrator (solid red line).

An MPLS-only branch location activates to the Orchestrator through the Private WAN Routing Function in the Transport Network, then through the Core Routing function in the Core Network, to the Orchestrator’s private IP address (dashed red line).

A hybrid Edge location may use either.

Packet flow for Edge Activation

Figure 9-2: Packet Flow in Data Center



Management Plane: Edge to Orchestrator (VECO)

After activation, Edges use their Loopback IP addresses to connect to the Orchestrator. The Edge puts this connection inside the tunnel to the Controller, using any transport tunnel available. This connection then goes out of the Controller through its eth1 connection to the core network and reaches the Orchestrator. The Controller keeps the Loopback IP address as the source (no SNAT as with 1-arm Controller). The Orchestrator then replies to the Edge with its loopback address, which is dynamically routable via the Controller eth1 interface for symmetric routing.

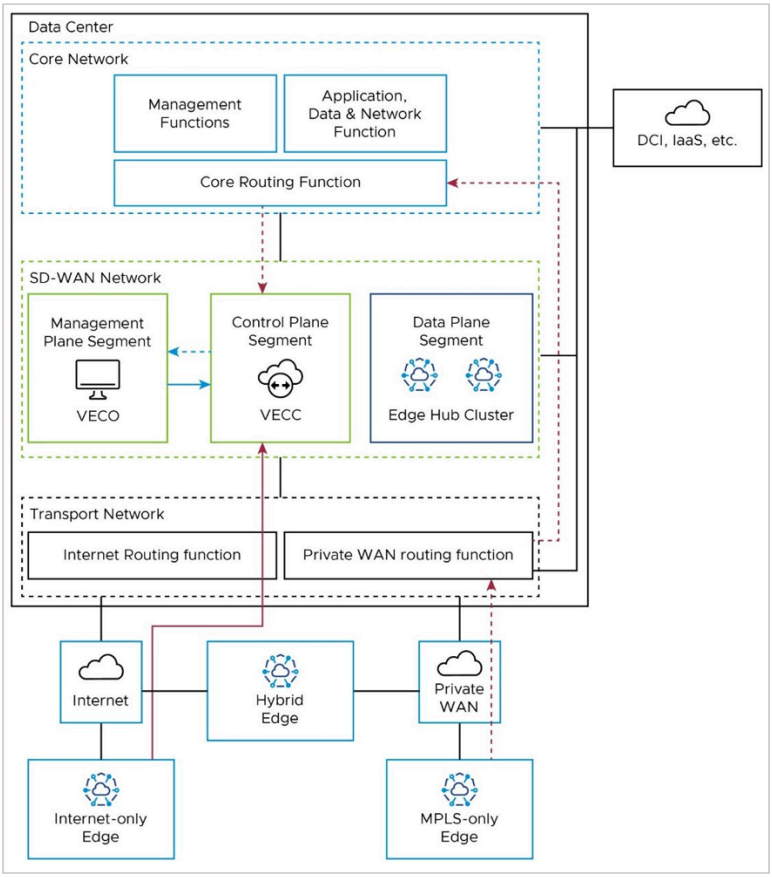
Activated Edges may also communicate directly with the Orchestrator via the underlay, in which case they would use the same packet flow paths as in the Activation section.

If Edges use the SD-WAN overlay via the Controller (also called the VECC), the path depends on the Edge type. Internet-only Edges use the SD-WAN overlay via Public WAN to the public IP of the Controller (solid red line in the diagram), where they leave the tunnel, and go from the Controller's public IP address to the public IP address of the Orchestrator (solid blue line in the diagram).

MPLS-only Edges take the SD-WAN overlay tunnel to the private IP of the Controller (dashed red line in the diagram). From there, they leave the overlay tunnel, and go to the private IP of the Orchestrator (dashed blue line in the diagram).

Note: You do not need NAT in this case because you have end-to-end private IP routing.

Figure 9-3: Packet Flow for Management Plane – Edge to Orchestrator (VECO)



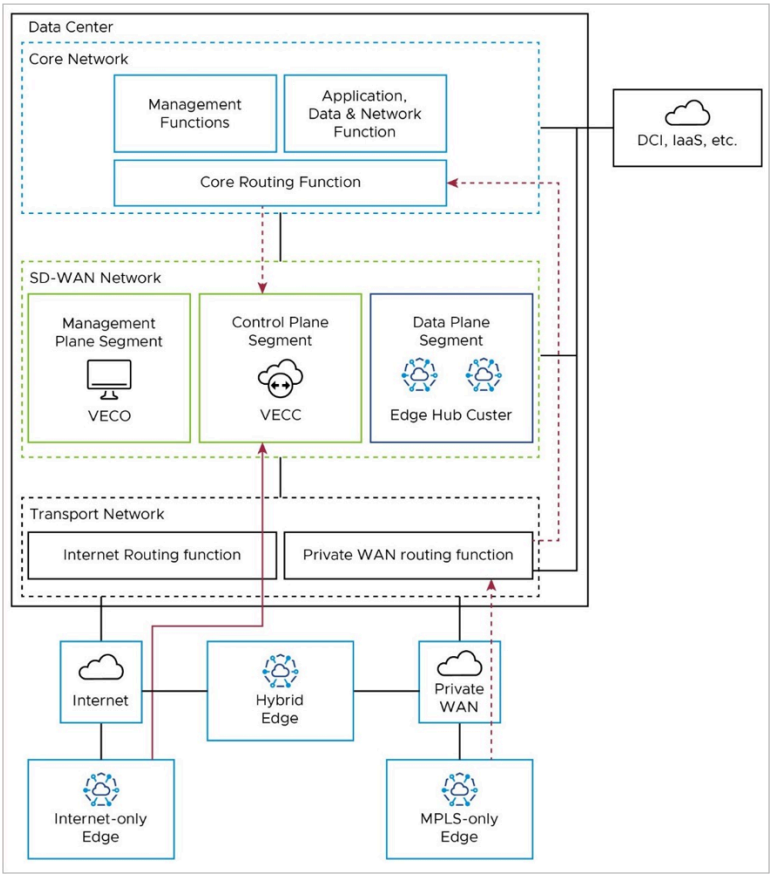
Control Plane: Edge to Controller (VECC)

All Edges connect to the Controller (also referred to as the VeloCloud Edge Cloud Controller, or VECC) with SD-WAN overlay tunnels. Usually, Edges have a Primary and Secondary Controller for backup. But for simplicity, we only show one Controller in the diagram.

Internet-only Edges connect to the Controller's public IP address with an SD-WAN overlay tunnel over the public internet. They use the Internet routing function in the Transport Network (solid red line in the diagram). MPLS-only Edges also connect to the Controller with an SD-WAN overlay tunnel, but they use the Controller's

private IP address. They use the Private WAN routing function in the Transport Network and the Core Routing function in the Core Network (solid red line in the diagram).

Figure 9-4: Packet Flow for Control Plane: Edge to Controller (VECC)

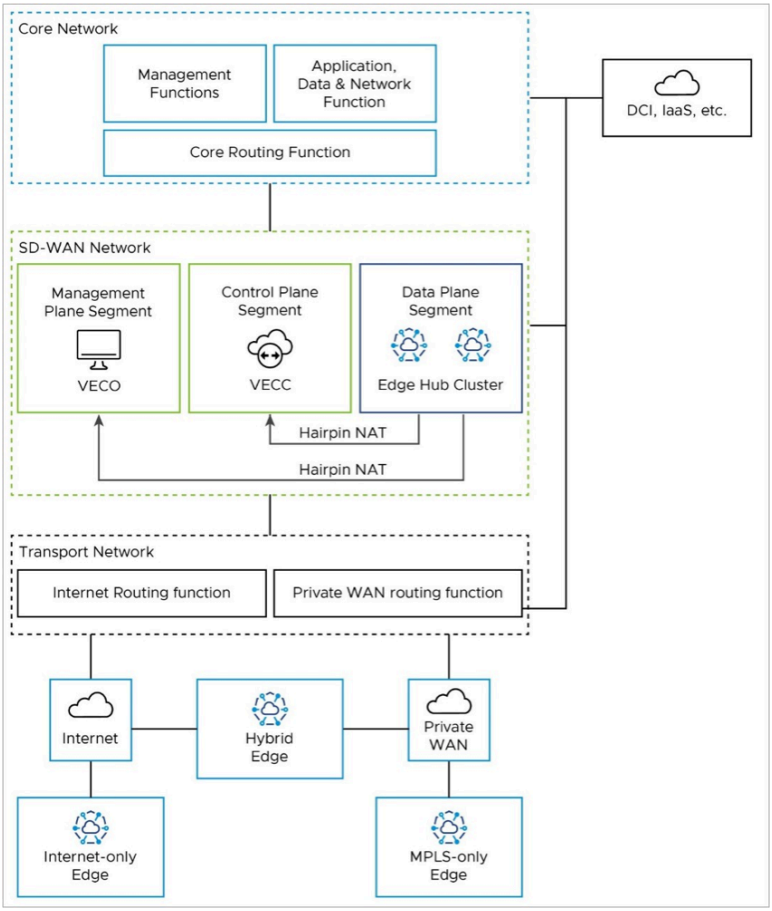


Hub Cluster to Controller (VECC)/Orchestrator (VECO): Special Case

The on-premises Hub Cluster provides a data path from Edge locations to the Core network, where the applications and resources are hosted. Usually, the Hub connects to a data center and accesses the Orchestrator and Controller remotely. So, the Hub Cluster needs to use public IP addresses to communicate with the Orchestrator/Controller; it cannot use the LAN-side / private IP network. To make the Hub Cluster look like it is at a remote location and access the Orchestrator/Controller over the public internet, you may

need to put the Hub Cluster in a separate network segment (for example, VRF). You may also need hairpin NAT functionality on a network device on the WAN side of the Hub Cluster, Controller, and Orchestrator.

Figure 9-5: Packet Flow- Hub Cluster to Controller (VECC)/Orchestrator



Edge to Core Network Functions – Data Path

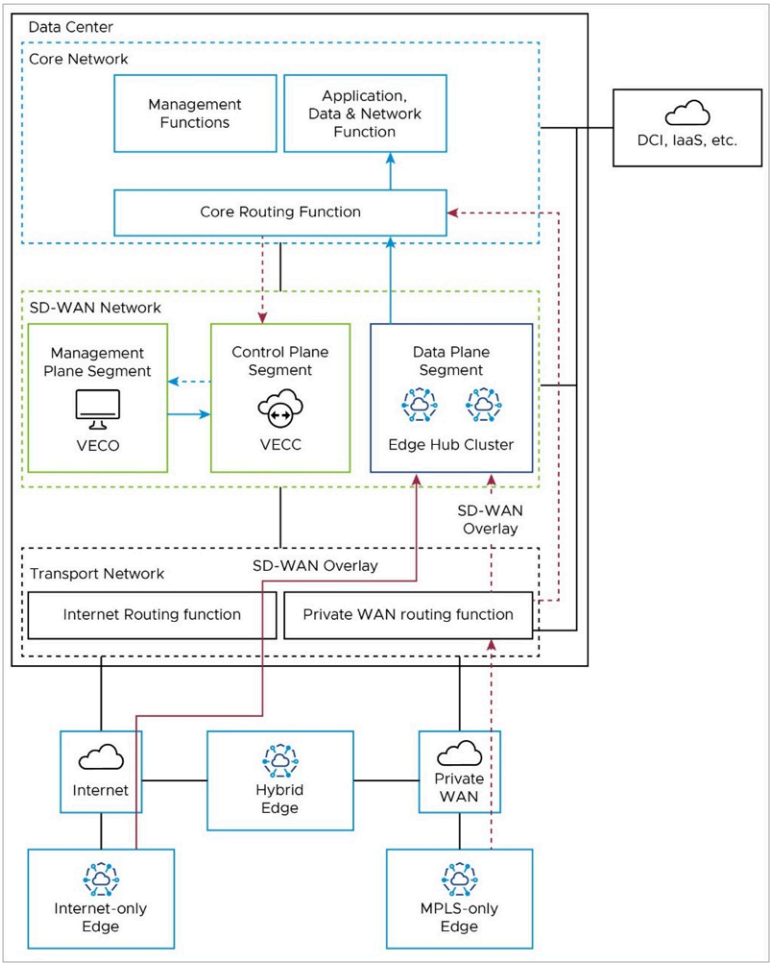
Branch Edges access applications and resources in the datacenter via the on-premises Edge Hub Cluster. Internet-only Edges establish an SD-WAN overlay tunnel to the public WAN IP address of the Hub Cluster via the Internet routing function in the Transport Network (solid red line). Similarly, MPLS-only Edges establish an SD-WAN overlay tunnel to the private WAN IP address of the Hub Cluster via the private routing function in the Transport Network (dashed red line).

Once the data path reaches the Hub Cluster, it is removed from the SD-WAN tunnel and natively routed to applications in the core network via the Core Routing Function (solid blue lines).

This path is shared by all Edge types. Hybrid locations may use a combination of public and private SD-WAN overlays to the Edge Hub Cluster.

Packet Flow for Edge to Core Network Functions- Data Path

Figure 9-6: Packet Flow- Hub Cluster to Controller (VECC)/Orchestrator

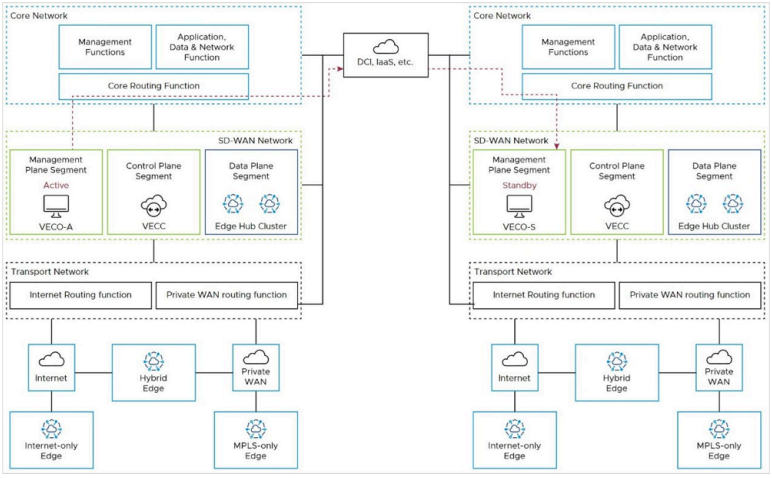


Orchestrator Disaster Recovery and Synchronization

When two VeloCloud Edge Cloud Orchestrators (VECOs, or Orchestrators) are deployed as an Active-Standby Disaster Recovery (DR) pair in geographically diverse data center locations, there is both a real-time synchronization function (for configurations, alarms, metrics, etc.) between the active and standby, as well as a keep-alive function (to detect the failure of the active Orchestrator). This data path uses the private IP address of the Orchestrator, and uses a Data Center Interconnect to reach the remote DC via Core Routing function (red line).

Additional details on Orchestrator Disaster Recovery may be found in the *Configure Orchestrator Disaster Recovery* section.

Figure 9-7: Disaster Recovery Topology and Packet Flow



Controller to Orchestrator

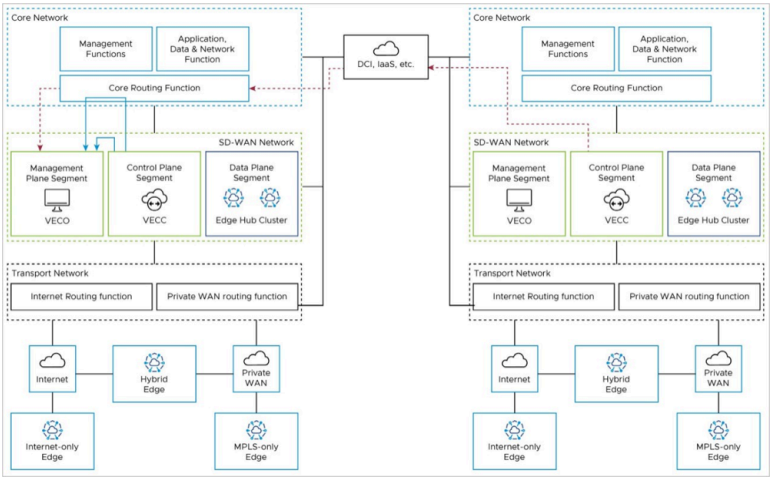
The Orchestrator (management plane node) must communicate with the Controller (control plane node). In an on-premises deployment, this is done using the private IP addresses associated with the core network facing interfaces.

Depending upon the DR configuration and the location of the active Orchestrator (in the local or remote DC), this connectivity may take different paths.

For a co-located Orchestrator and Controller, the nodes may communicate directly within the SD-WAN Network block, or pass to the Core Routing Function (solid blue lines).

For a remote Orchestrator and Controller, the Controller must use the Core Routing function and DCI to reach the DR-active Orchestrator (dashed red lines).

Figure 9-8: Orchestrator to Controller Packet Flow



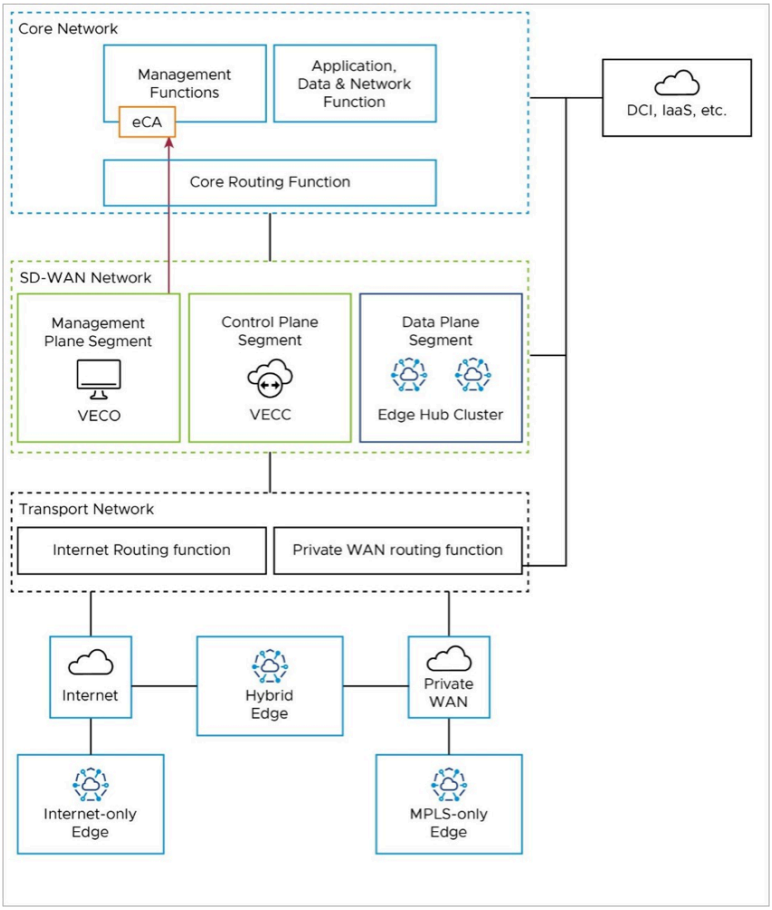
Controller to External Certificate Authority – Optional

IF an External CA (eCA) is used in the deployment, it would most likely reside in the Management Functions of the Core Network. The Orchestrator would use the private IP address of the core network facing interface to reach the eCA via the Core Routing Function.

If the Orchestrator is deployed as an active-standby DR pair, the Orchestrator fail-over may require the Orchestrator to reach the eCA via DCI.

Additional details on External CA may be found here:

Figure 9-9: Controller to External CA Packet Flow



Design Requirements & Assumptions for a Federal Deployment

This section includes requirements, caveats, unique aspects, and best practices specific to an on-premises Federal deployment.

- The solution spans two or more Data Centers.
- For redundancy, the VeloCloud Edge Cloud Orchestrator (VECO, or just Orchestrator) is deployed in Disaster Recovery mode. This means that two Orchestrators are deployed in an active-standby pair, one in each Data Center.
- The paired Orchestrators must have L3 connectivity to one another (for example, via DCI) to maintain data synchronization in the event of a failure of the active Orchestrator.

- Controllers are also present in both data centers. Since either one of the VECOs in the DR pair may be active, there must be L3 reachability between the Controllers and VECOs at both data centers.
- Internet (public) transport only Sites: Branch Edge locations that have access to only Public WAN (internet) transport networks.
- Private transport only Sites: Branch Edge locations that have access to only Private WAN (MPLS) transport networks.
- Hybrid (public/private) transport mix Sites: Branch Edge locations that have access to both Public and Private WAN transport networks.
- In some networks, no reachability to public prefixes from private transport is allowed. For example, the Orchestrator and Controller are `1.1.1.1` and `2.2.2.2`, there is no reachability to those addresses from the private transport. The private transports do not have a route to these addresses, nor can they be advertised, also there is no default route.

Derived Requirements

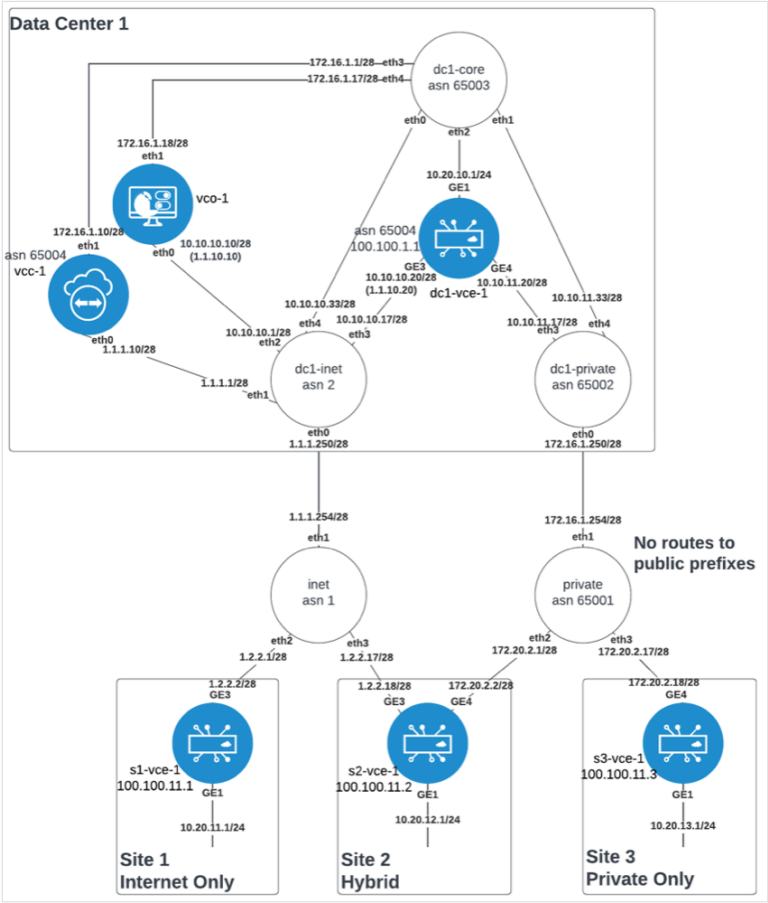
The following requirements follow from the above design requirements and assumptions:

- The VeloCloud Edge Cloud Orchestrator (VECO, or just Orchestrator) must be directly reachable on the public internet.
  - This is achieved through the use of network address translation (NAT).
- The Controller(s) must be directly reachable on the public internet.
- The Controller(s) are deployed in 2-Arm Mode (Partner Controller) to accommodate private transport reachability.
  - The Controller(s) attract Orchestrator communication from the SD-WAN Edge by advertising both the public (NAT) and private IP address of the Orchestrator to the overlay.
  - A Hub Edge uses BGP filters to block these Orchestrator advertisements from being advertised back to the Data Center.
- Edges use loopback interfaces and IP addresses for Orchestrator communication, and place the Orchestrator connection within the Partner Controller tunnel post-activation.
- A Controller in Partner Controller mode establishes BGP peering with the data center to advertise Edge loopback IP addresses being used for Edge-to-Orchestrator traffic to ensure Orchestrator return traffic remains symmetrical within the Partner Controller tunnel to the Edge.
  - BGP filters are used to ensure only the Edge loopback IP addresses are advertised through this peering – block site user prefixes.
- The Orchestrator has 2 interfaces:
  - Eth0 for Edge connections via Controller – routed via the Controller in Partner Controller mode.
  - Eth1 for HTTPS access – routed via the Hub Edge.
- Two Operator Profiles are used:
  - A Public Operator Profile assigns Public IP addresses of the Orchestrator to the Edge.
    - Used for public transport-only Edges.
  - A Private Operator Profile assigns Private IP address of Orchestrator to the Edge
    - Used for private transport-only Edges, and Hybrid Edges

A Sample Minimal Topology

The Sample Minimal Topology adds detail to the Reference Architecture, providing an example of how the solution may be integrated into an existing network.

Figure 9-10: Minimal Topology for Integration



Note: In the above diagram, VCO stands for the Orchestrator, VCC stands for the Controller, and VCE stands for the Edge.

While most real networks would have a more complex design, the expectation is that this design can be extrapolated to a real network with which it is being integrated.

Note: The interfaces and IP addressing in the diagram are for example purposes only, and will be referenced throughout the remainder of this guide.

Network Address Translation (NAT)

Network Address Translation (NAT) is used at the Hub Edge. It can be avoided if all components (Orchestrator, Controller, and Edges) are provided public IP addresses directly on their interface, but this is uncommon due to security requirements and limited IPv4 addresses in some networks.

In the minimal topology, the Orchestrator and the Edge(s) are placed behind a NAT boundary. The Controller is placed in front of the NAT boundary. In the example, the dc1-inet underlay router eth0 (internet connection) and eth1 (Controller) interfaces are in front of the NAT boundary, eth2 (Orchestrator), eth3 (Edge) and eth4

(data center) are behind the NAT boundary. This placement of components requires the minimal amount of NAT configuration required. The NATs needed are:

1. Orchestrator 1:1 NAT: public IP to private IP translation on all eth0 inbound traffic. The VCOs publicly reachable IP address is 1.1.10.10, but its real IP address behind the NAT boundary is 10.10.10.10. All traffic initiating from and ingressing from the internet with destination IP address of 1.1.10.10 needs to have the destination address translated to 10.10.10.10. This translation must be bidirectional, in that the return traffic with a source IP address of 10.10.10.10 needs to be translated to a source IP address of 1.1.10.10.

   The same requirement is necessary for inbound traffic on eth4, as public (internet only) Edges attempting to communicate to Orchestrator using VCOs public IP address (1.1.10.10) will egress the vcc on eth1 and route through the data center core to the internet edge, inbound on eth4.

2. Edge 1:1 NAT: public IP to private IP translation on all eth0 inbound traffic. The Edges publicly reachable IP address is 1.1.10.20, but its real IP address behind the NAT boundary is 10.10.10.20. All traffic initiating from and ingressing from the internet with a destination IP address of 1.1.10.20 needs to have the destination address translated to 10.10.10.20. This translation must be bidirectional, in that the return traffic with a source IP address of 10.10.10.20 needs to be translated to a source IP address of 1.1.10.20.

3. Edge SNAT: Source Network Address Translation on all eth1 outbound traffic. All traffic initiating from the Edge destined to the Controller (egress eth1) with a source IP address of 10.10.10.20 needs to have the source IP address translated to 1.1.10.20. This translation must be bidirectional, in that the return traffic with a destination address to 1.1.10.20 needs to be translated to a destination IP address of 10.10.10.20.

An example configuration of these NAT rules in vyos:

```
nat { destination { rule 10 { destination { address 1.1.10.10 } inbound-interface eth0 translation
 { address 10.10.10.10 } } rule 15 { destination { address 1.1.10.10 } inbound-interface eth4
 translation { address 10.10.10.10 } } rule 20 { destination { address 1.1.10.20 } inbound-inter
face eth0 translation { address 10.10.10.20 } } } source { rule 20 { outbound-interface eth1
 source { address 10.10.10.20 } translation { address 1.1.10.20 } } rule 999 { outbound-interface
 eth0 translation { address masquerade } } } }
```

Routing

You need to designate 4 routing summary blocks:

1. All Hub Edge WAN IP addresses – the sample minimal topology uses `10.10.0.0/16`.
2. All Spoke Edge Private WAN IP addresses – the sample minimal topology uses `172.20.0.0/16`.
3. All Edge Loopbacks – the sample minimal topology uses `100.100.0.0/16`.
4. All Spoke Edge client user subnets – the sample minimal topology uses `10.20.0.0/16`.

Orchestrator Routing

The VeloCloud Edge Cloud Orchestrator (VECO, or just Orchestrator) operating system needs to have the following routes:

1. Hub Edge WAN IP summary – via eth0 next hop with metric 0.
2. Spoke Edge Private WAN IP summary – via eth0 next hop with metric 0.
3. Spoke Edge Public WAN IP summary (default route) – via eth0 next hop with metric 0.
4. All Edge Loopbacks – via eth0 next hop with metric 0.

5. All Spoke Edge client user subnets – via eth1 next hop with metric 0.
6. Controller eth1 subnet – via eth1 next hop with metric 0.

Controller Routing

The Controller operating system needs to have the following routes:

1. Default route – via eth0 next hop with metric 0
2. Default route – via eth1 next hop with metric 5
3. Orchestrator eth1 subnet- via eth1 next hop with metric 0

Controller SD-WAN Control Plane Routing

- The Controller control plane injects Orchestrator routes to the overlay for both Orchestrator IPs: 1.1.10.10/32 and 10.10.10.10/32. This draws connections to Orchestrator from the Edge to use the Controller VCMP tunnels.
- The Controller control plane peers BGP on eth1 with its next-hop to dynamically advertise Loopback IP addresses of Edges with VCMP tunnels established. This ensures return traffic from the Orchestrator to the Edges will come back to the Controller to be placed back into the VCMP tunnel.

Connections

Orchestrator Connections

- The Orchestrator uses eth0 for all HTTPS management connections to Edges via the underlay and via the overlay (from Controller tunnels).
- The Orchestrator uses eth1 for all HTTPS user/administrator GUI.
- The Orchestrator-to-Controller communication uses Controller eth1-to-Orchestrator eth1 path.

Controller Connections

- The Controller uses eth0 for all public transport VCMP tunnels, and eth1 for all private transport VCMP tunnels.

Edge Connections

- Edges use their WAN IP address for all activations to the Orchestrator. Private transport Edges use 10.10.10.10 to activate, public transport Edges use 1.1.10.10 to activate. Hybrid Edges can use whichever path to the Orchestrator is available.
- Edges then build VCMP tunnels (UDP 2426) using their WAN IP addresses to the Controller. Public transports build to Controller eth0, and private transports build to Controller eth1.
- Post activation, Edges use their Loopback IP addresses to source all connections to the Orchestrator. The Edge places this connection inside the tunnel to the Controller, using either transport tunnel available. This connection then egresses the Controller through its eth1 connection to the core network destined for the Orchestrator. The Controller leaves the Loopback IP address as the source in tact (no SNAT as with 1-arm Controller). The Orchestrator then replies to the Edge destined to its loopback which is dynamically routable via the Controller eth1 interface to ensure symmetric routing.

Prerequisites

The following are the prerequisites needed to deploy an on-premises SD-WAN solution:

1. ESXi 6.5.0 or higher.
2. Intel Xeon.
3. Hyperthreading disabled (for Edge).
4. SSD storage supporting 10k or higher IOPS
5. Intel NIC with DPDK and SR-IOV support
6. VDS, vswitches, and port groups pre-configured
7. Encrypted disks are optional
8. Plan for 5GB/year/edge
9. Linux (such as Ubuntu VM) with genisoimage and tree installed

Cloud-init

The Orchestrator, Controller (here referred to as the Gateway), and Edge all boot from a mounted ISO file for their initial bootstrap configuration. This ISO file uses the Linux cloud-init method. Cloud-init consists of YAML files. YAML files are created, and genisoimage is used to create an ISO for each VM.

Note: For the Cloud-init section the Controller will be referred to as the Gateway as this is the equivalent name for this component and what a user would see on the Orchestrator User Interface.

1. **Create YAML files**

   On the Linux machine, create 3 YAML files each for Orchestrator and Gateway: meta-data, network-config, and user-data. Create 2 files for Edge: user-data and meta-data. Arrange in directory structure to match the following:

   On the Linux machine, create 3 yaml files each for Orchestrator and Gateway: meta-data, network-config, and user-data. Create 2 files for Edge: user-data and meta-data. Arrange in directory structure to match the following:

   Note: The acronym VCG matches to the Gateway (Controller) component; VCE matches to the Edge; and VCO matches to the Orchestrator.

   ```
   isos/ | |--VCG | |--meta-data | |--network-config | |--user-data | |--VCE | |--meta-data | |--
   user-data | |--VCO | |--meta-data | |--network-config | |--user-data
   ```

   a. **Orchestrator (VCO) Example:**

      1. **user-data**

         ```
         #cloud-config hostname: vco-1 password: Velocloud123 chpasswd: {expire: False}
          ssh_pwauth: True velocloud: fips_mode: compliant vco: super_users: list: | admin@lab.loc
         al:Velocloud123 remove_default_users: False
         ```

      2. **network-config**

         ```
         version: 2 ethernets: eth0: addresses: - 10.10.10.10/28 routes: - to: 0.0.0.0/0
          via: 10.10.10.1 metric: 0 nameservers: addresses: [10.10.10.1] eth1: addresses: -
          172.16.1.18/28 routes: - to: 10.20.0.0/16 via: 172.16.1.17 metric: 0 - to: 172.16.0.0/12
          via: 172.16.1.17 metric: 0 - to: 192.168.0.0/16 via: 172.16.1.17 metric: 0
         ```

3. **meta-data**

```
instance-id: vco-1 local-hostname: vco-1
```

b. **Gateway (VCG) Example:**

1. **user-data**

```
#cloud-config hostname: vcc-1 password: Velocloud123 chpasswd: {expire: False}
 ssh_pwauth: True velocloud: fips_mode: compliant
```

2. **network-config**

```
version: 2 ethernets: eth0: addresses: - 1.1.1.10/28 routes: - to: 0.0.0.0/0 via: 1.1.1.1
 metric: 0 nameservers: addresses: [1.1.1.1] eth1: addresses: - 172.16.1.10/28 routes: -
 to: 0.0.0.0/0 via: 172.16.1.1 metric: 5 - to: 172.16.1.16/28 via: 172.16.1.1 metric: 0
```

3. **meta-data**

```
instance-id: vcc-1 local-hostname: vcc-1
```

c. **Edge (VCE) Example**

1. **user-data**

```
#cloud-config hostname: vce password: Velocloud123 chpasswd: {expire: False} ssh_pwauth:
 True
```

2. **meta-data**

```
instance-id: vce
```

2. **Generate ISO**

Example:

```
cd iso cd vco genisoimage -output cdrom.iso -volid cidata -joliet -rock user-data meta-data
 network-config cd .. cd vcc genisoimage -output cdrom.iso -volid cidata -joliet -rock user-
data meta-data network-config cd .. cd vce genisoimage -output cdrom.iso -volid cidata -joliet
 -rock user-data meta-data cd ..
```

The directory structure should look as follows with cdrom.iso added to each directory:

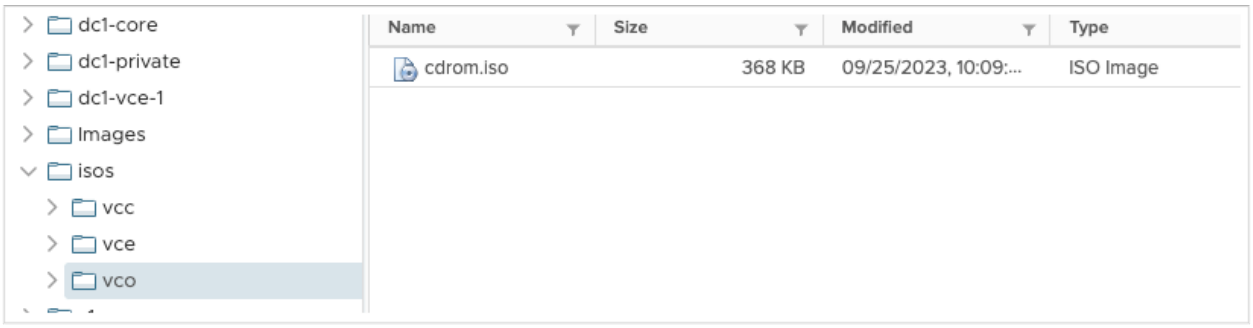
Note: The acronym VCG matches to the Gateway component; VCE matches to the Edge; and VCO matches to the Orchestrator.

```
isos/ | |--VCG | |--cdrom.iso | |--meta-data | |--network-config | |--user-data | |--VCE | |--
cdrom.iso | |--meta-data | |--user-data | |--VCO | |--cdrom.iso | |--meta-data | |--network-co
nfig | |--user-data
```

3. **Upload the 3 ISO files to the ESXi datastore**

Upload location for ISO files

Figure 9-11: ISO Files Upload Location



4. **Orchestrator Deployment- ESXi**

> **Note:** Skip to the next section if using vCenter.

a. Create/Register VM

b. Deploy a virtual machine from an OVF or OVA file, **Next**

c. Provide appropriate name, browse or drag OVA file, **Next**

d. Select storage, **Next**

e. Select appropriate port group for the Orchestrator eth0 interface (VCO also refers to the Orchestrator)

f. Select **thick provision**

g. Uncheck **power on automatically**

h. **Next**

i. **Finish**

j. Wait for the OVF import to complete

k. Select the newly deployed VM

l. **Edit**

m. **Add Network Adapter**

n. Select appropriate port group for Orchestrator eth1 interface

o. Expand CD/DVD Drive 1

p. Change to **Datastore ISO file**

q. Browse to location of uploaded ISO files, select Orchestrator (or VCO) ISO, **Select**

r. Check **Connect at Power On**, and **Connect**

s. **Save**

t. Power On VM

u. Wait for FIPS mode, the VM will automatically reboot after 5-10 minutes

v. Login with vcadmin/Velocloud123

w. Configure NTP at `/etc/ntp.conf` (only if private NTP needed, default uses 0.ubuntu.pool.ntp.org)

1. `sudo vi /etc/ntp.conf`
2. `pool 10.10.10.1 iburst`
3. `escape`
4. `:wq!`
5. `Sudo service ntp restart`
6. `Sudo ntpq -c peers`

5. **Orchestrator Deployment – vCenter**

a. Actions, Deploy OVF Template.

b. Choose VCO OVA file from local, Next

c. Provide appropriate name, select appropriate location

d. Select appropriate compute resource, Next

e. **Review**

f. **Next**

g. Select storage, **thin provision**

h. Select appropriate port group for the Orchestrator (or, VCO) eth0 interface, **Next**

i. Template values, do not use, leave all empty, **Next**

j. **Finish**

k. Wait for OVF Deploy/Import tasks to complete

l. Select VM

m. **Actions** > **Edit Settings**

n. **ADD NEW DEVICE**

o. Network Adapter

p. **New Network**: select appropriate port group for the Orchestrator (or, VCO) eth1 interface

q. Expand CD/DVD Drive

r. Change to **Datastore ISO file**

s. Browse to ISO, select it, **OK**

t. Check box for **Connect at Power on**

u. **OK**

v. Power On VM

w. Wait for FIPS mode, VM will automatically reboot after 5-10 minutes

x. Login with *vcadmin*/*Velocloud123*

y. Configure NTP at `/etc/ntp.conf` (only if private NTP needed, default uses `0.ubuntu.pool.ntp.org`)

- `sudo vi /etc/ntp.conf`
- `pool 10.10.10.1 iburst`
- `escape`
- `:wq!`
- `Sudo service ntp restart`
- `Sudo ntpq -c peers`

6. **Orchestrator Initial Configuration**

a. Login to the Orchestrator at https://1.1.10.10/ui/operator with credentials `super@velocloud.net/vcadm!n`; or `admin@lab.local/Velocloud123`; or customized credentials per ISO file are also available if ISO boot was used.

b. From the top navigation **Select Orchestrator**.

c. From the left navigation select **System Properties**.

d. Select `network.public.address`.

e. Replace localhost in value field with `1.1.10.10`, **Save Changes**.

f. Search the top search bar for websocket.

g. Select `network.portal.websocket.address`.

h. Value: `172.16.1.18`, **System Properties**.

i. Search top search bar for source.

j. Select `gateway.activation.validate.source`.

k. Change value to **False**, **Save Changes**.

Note: To configure the Orchestrator in a Disaster Recovery (DR) topology, see the topic *Configure Orchestrator Disaster Recovery*.

7. **Gateway Staging**

a. From the top navigation select **Gateway Management**

b. From the left navigation select **Gateway Pools**

c. Select **Default Pool**

d. Change Partner Gateway Handoff to: **Allow**

e. **SAVE CHANGES**

f. From the left navigation select **Gateways**

g. **New Gateway**

1. Name: *vcc-1*
2. IPv4 address: `1.1.1.10`
3. Service State: **In Service**
4. Gateway Pool: **Default Pool**
5. **Create**
6. Select the newly created Gateway vcc-1
7. Gateway Roles: **Partner Gateway**
8. Partner Gateway (Advanced Hand Off) Details, delete all default Static Routes
9. Static Routes **+ADD**
10. Subnet: `10.10.10.10/32`, Cost: **1**, Encrypt: **Check the box**, Handoff: **VLAN**
11. Subnet: `1.1.10.10/32`, Cost: **1**, Encrypt: **Check the box**, Handoff: **VLAN**
12. **SAVE CHANGES**
13. Save the activation key from the yellow bar at the top

8. **Gateway Deployment- vCenter**

Note: For a Gateway ESXi install, follow a similar process as outlined in Step 4 for the ESXi Orchestrator.

a. Deploy OVF Template.

b. Choose VCG OVA file from local. (VCG is the equivalent for Gateway)

c. Provide appropriate name, select appropriate location.

d. Select appropriate compute resource.

e. **Review**

f. Select storage, thick provision.

g. Select appropriate port group for Outside, Inside.

h. Template values: do not use, leave all empty.

i. **Next**

j. **Finish**

k. Wait for OVF Deploy/Import tasks to complete.

l. Select VM.

m. Actions, **Edit Settings**

n. Expand CD/DVD Drive 1.

o. Change to **DataStore ISO file**.

p. Browse to ISO, select, **OK**.

q. Check Connect at Power On, **OK**.

r. Power On VM.

s. Wait for FIPS mode to enable, the instance will reboot automatically.

t. FIPS is complete when login says `Linux 4.15.0-1113-fips x86_64`.

```
vcg-1 login: vcadmin Password: Welcome to Velocloud OS (GNU/Linux 4.15.0-1113-fips x86_64)
```

u. Login with *vcadmin*/*Velocloud123*

v. Configure NTP at `/etc/ntp.conf` (only if private NTP needed, default uses 0.ubuntu.pool.ntp.org).

```
pool 10.10.10.1 iburst sudo service ntp restart sudo ntpq -c peers
```

w. Ensure offset is less than or equal to 15ms using:

```
sudo ntpq-p
```

x. Configure `vc_blocked_subnets`

1. `cd /opt/vc/etc`
2. `sudo vi vc_blocked_subnets.json`
3. delete all lines
4. insert: {}
5. press the **escape** key
6. `:wq!`

y. Configure `gatewayd`

1. `cd /etc/config`
2. `sudo vi gatewayd`
3. in the wan field (4th line) and geneve field (5th line) substitute eth1 in place of eth0

```
"wan": ["eth1"], "geneve": ["eth1"],
```

4. press the **escape** key
5. `:wq!`

z.

```
sudo reboot
```

aa. Manually activate

1. login with *vcadmin*/*Velocloud123*
2. ```
   sudo su
   ```
3. ```
   cd /opt/vc/bin
   ```
4. ```
   activate.py -I -s 172.16.1.18 <activation key>
   ```
5. a successful activation would output the following:

   ```
   Activation successful, VCO overridden back to 172.16.1.18
   ```

9. Orchestrator Operator Profile

> **Note:** For an ESXi installation, follow a process similar to Step 4a for the Orchestrator.

a. Login to the Orchestrator User Interface as an Operator.

> **Note:** Make sure the URL includes `/operator` at the end.

b. Top navigation: **Edge Image Management**

c. Left navigation: **Software**

d. **Upload Image**

e. Browse to or drop an appropriate Edge image, for example:

`edge-imageupdate-VC_VMDK-x86_64-5.2.0.2-83770177-R5202-20230725-GA-6969b39047`

f. **Done**

g. Left navigation: **Application Maps**

h. **Upload**

i. Browse to or drop appropriate application map, for example: `r5200_app_map.json`

j. **Edit** (do not save)

k. Rename to version, for example: **R5200**

l. **Save Changes**

m. Top navigation: **Administration**

n. Left navigation: **Operator Profiles**

o. **NEW**

1. Name: **R5200-PUBLIC**
   ```

2. **Create**
3. Select newly created **R5200**
4. Ensure Orchestrator Address: **IP address**
5. Ensure Orchestrator IPv4 Address: `1.1.10.10`
6. Application Map Assignment, JSON File: **R5200**
7. Software Version: **Toggled On**
8. Version: **5.2.0.2**
9. **Save Changes**

p. Left Navigation: **Operator Profiles**

q. Check the box next to **R5200**

r. **DUPLICATE**

s. Name: R5200-PRIVATE, **CREATE**

t. Select R5200-PRIVATE

u. Change Orchestrator IPv4 address to: `10.10.10.10`

v. **SAVE CHANGES**

10. **Customer Configuration**

a. Login to the Orchestrator User Interface as an Operator.

Note: Make sure the URL includes **/operator** at the end.

b. Top navigation: **Customers & Partners**

c. Left navigation: **Manage Customers**

d. **+NEW CUSTOMER**

1. Company Name: *cust1*
2. Check the **SASE Support Access** box
3. Check the **SASE User Management Access** box
4. **Next**
5. **Administrative Account**
6. Username: *cust1@lab.local*
7. Password: *Velocloud123*
8. **Next**

9. **Services**
10. Check the **SD-WAN** box
11. Gateway Pool: Default Pool
12. Check the **Allow Customer to Manage Software** box
13. Software Image, **+ADD**
14. Select **R5200-PRIVATE** and **R5200-PUBLIC**, select the right pointing arrow to move these to the **Selected Images**
15. Select the radio button to the right of **R5200-PRIVATE**
16. **Done**
17. **SD-WAN, Edge Licensing**, **+ADD**
18. Search for the term 'POC'
19. Select **POC | 10Gbps |**, select the right pointing arrow to move this license to **Selected Edge Licenses**
20. **Save**
21. Check the box for Feature Access, Stateful Firewall
22. **ADD CUSTOMER**

e. Select the newly created '*cust1*' check box to the left

f. At top, select **EDIT CUSTOMER SYSTEM SETTINGS**

g. Left navigation, select **Customer Configuration**

h. Scroll to the bottom and then expand **SD-WAN Settings**

i. Check the box for **Distributed Cost Calculation**

j. Check the box for **Use NSD Policy**

k. **Save Changes**

11. **Partner Gateway Configuration**

a. Top navigation: customer selected, **Global Settings**

b. Left navigation: **Customer Configuration**

c. Expand Gateway Pool

d. Toggle Partner Hand Off to **ON**

e. Configure Hand Off radio button, change to **Per Gateway**

f. Confirm with **OK**

g. Under vcc-1 – Global Segment, select **Configure BFD & BGP**

h. BGP toggle to **ON**

i. Customer ASN: **65004**

j. Hand Off Interface, Local IP address: `172.16.1.10/28`

k. Use for Private Tunnels: **Check the box**

l. Advertise Local IP address via BGP: **Check the box**

m. BGP, Neighbor IP: `172.16.1.1`

n. Neighbor ASN: **65003**

o. Secure BGP Routes: **Check the box**

p. BGP Inbound Filters, **+ADD**

q. Match type: prefix for IPv4, match value: `0.0.0.0/0`, Exact Match: **No**, Action Type: **Deny**

r. BGP Outbound Filters, **+ADD**

s. Match Type: prefix for IPv4, match value `100.100.0.0/16`, Exact Match = No, Action Type: **Permit**, Action Set: **Community: 777:777**, Community Additive: **Activated**

t. **UPDATE**

u. **SAVE CHANGES**

12. **Quick Start Profile Configuration**

a. Dark blue navigation bar, select **Global Settings** Dropdown, select **SD-WAN**

b. Top navigation bar, select **Configure**

c. Left Navigation bar, select **Profiles**

d. Select **Quick Start Profile**

e. Select **Firewall** tab

f. Expand **Edge Access**

g. Console Access: **Allow**

h. **SAVE CHANGES**

i. Select **Device** Tab

j. Expand **Connectivity** > **Interfaces**, X out all Edge models not in use

k. Select **GE1**

l. Change Capability to **Routed**

m. Disable **Underlay Accounting**

n. Disable **Enable WAN Link**

o. **IPv4 Settings > Addressing Type**, change to **Static**

p. Check the **Advertise** box

q. Uncheck the **NAT Direct Traffic** box

r. **SAVE**

s. Repeat steps k-through-r for **GE2**

t. **Select GE2**, uncheck the **Interface Enabled** box

u. **SAVE**

v. **SAVE CHANGES**

w. Expand **VLAN**, select 1-Corporate, **DELETE**

x. **Confirm DELETE**

y. Under **Connectivity** > **Interfaces**, select **GE3**

z. **IPv4 Settings** > **Addressing Type: Static**, then **SAVE**

aa. **Select GE4**

ab. **IPv4 Settings** > **Addressing Type: Static**

ac. WAN Link: **User Defined**

ad. **SAVE**

ae. Under VPN Services, Expand Gateway Handoff Assignment, **+SELECT GATEWAYS**

af. Check vcc-1, **UPDATE**

ag. Under **VPN Services**, toggle **Cloud VPN** to **ON**

ah. **SAVE CHANGES**

13. **Hub Edge Profile**

a. **Configure** > **Profiles**

b. Check the box next to **Quick Start Profile**

c. **DUPLICATE**

d. Name: Hub, **CREATE**

14. **Hub Edge Staging**

a. Top navigation: **Configure**

b. Left navigation: **Edges**

c. **+ADD EDGE**

d. Name: *dc1-vce-1*

e. Model: select as appropriate

f. Profile: **Hub**

g. Edge License: **POC**

h. **NEXT**

i. **ADD EDGE**

j. Expand Loopback Interfaces, **+ADD**

k. Interface ID: **1**

l. IPv4 Address: `100.100.1.1`, **ADD**

m. Expand Management Traffic

n. Source Interface: **LO1**

o. Expand Interfaces section

p. Select **GE1**

q. IPv4 settings, enter IP address: `10.10.11.1`

r. Set Cidr prefix: **24**

s. **SAVE**

t. Select **GE3**

u. IPv4 settings, enter IP address: `10.10.10.20`

v. CIDR Prefix: **28**

w. Gateway: `10.10.10.17`

x. **SAVE**

y. Select **GE4**

z. IPv4 settings, enter IP address: `10.10.11.20`

aa. CIDR Prefix: **28**

ab. Gateway: `10.10.11.17`

ac. **SAVE**

ad. WAN Link Configuration section, **+ADD USER DEFINED WAN LINK**

ae. Link Type: **Private**

af. Name: **Private**

ag. SD-WAN Service Reachable: **Checked**

ah. SD-WAN Service Reachable Backup: **Uncheck the box**

ai. Interfaces: **GE4**

aj. **ADD LINK**

ak. **SAVE CHANGES**

al. Copy the activation key from the yellow bar at the top

15. **Hub Edge BGP Configuration**

CAUTION: You must apply proper route maps to prevents loops to the Orchestrator.

Important: The following steps are mandatory and must be done prior to activating the Hub Edge.

a. **Configure** > **Edges**

b. Select *dc1-vce-1*

c. BGP: Check Override, **Toggle ON**, Expand

d. Local ASN: **65004**

e. Filter List: **+ADD**

f. Filter Name: **outbound**

g. Filter Rules: 1 Rule, select the link

h. Match type: **Prefix for IPv4**

i. Match value: `10.10.10.10/32`

j. Exact Match: **check the box**

k. Action Type: **Deny**

l. Check first rule, **CLONE**

m. Match Type: **Prefix for IPv4**

n. Match Value: `1.1.10.10/32`

o. Exact Match: **check the box**

p. Action Type: **Deny**

q. **+ADD**

r. Match Type: **Prefix for IPv4**

s. Match Value: `100.100.0.0/16`

t. Exact Match: **do not check the box**

u. Action Type: **Deny**

v. **+ADD**

w. Match Type: **Prefix for IPv4**

x. Match Value: `0.0.0.0/0`

y. Exact Match: **do not check the box**

z. Action Type: **Permit**

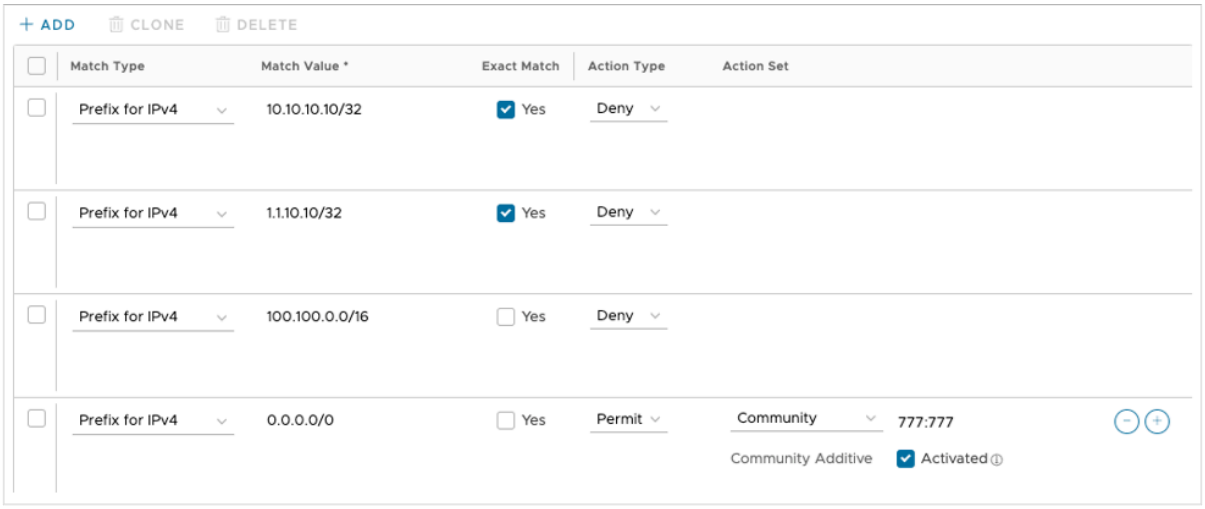
aa. Action Set: **Community 777:777**

ab. Community Additive: **Activated checked**

ac. **SUBMIT**

ad. RESULT: Outbound Filter Rules

Figure 9-12: Outbound Filter



ae. Filter List, **+ADD**

af. Filter Name: **inbound**

ag. Filter Rules: 1 Rule, select the link

ah. Match type: **Community**

ai. Match Value: **777:777**

aj. Exact Match: **No**

ak. Action Type: **Deny**

al. **+ADD**

am. Match Type: **Prefix for IPv4**

an. Match Value: `0.0.0.0/0`

ao. Exact Match: **check the box**

ap. Action Type: **permit**

aq. **+ADD**

ar. Match Type: **Prefix for IPv4**

as. Match Value: `0.0.0.0/0`

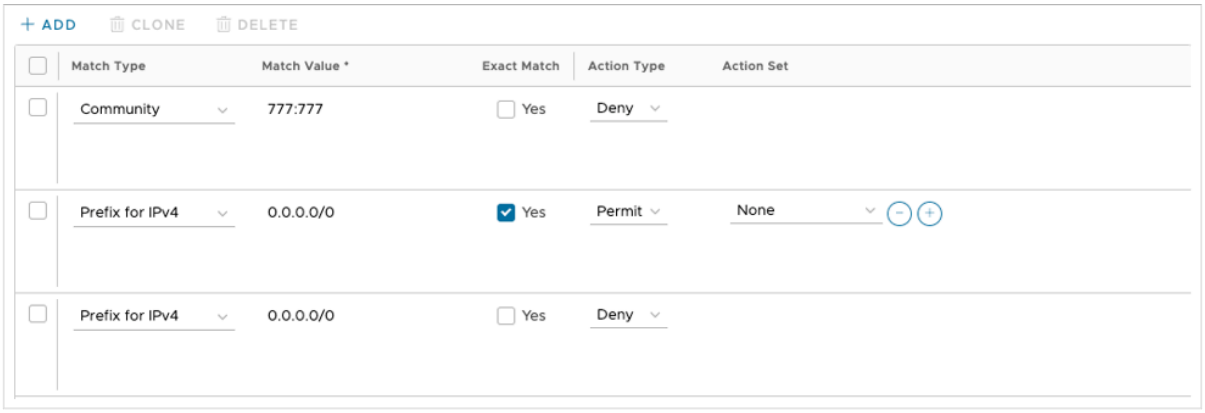
at. Exact Match: **do not check the box**

au. Action Type: **Deny**

av. **SUBMIT**

aw. RESULT: Inbound Filter Rules

Figure 9-13: Inbound Filters



ax. Neighbors, **+ADD**

ay. Neighbor IP: `10.10.11.2`

az. ASN: **65003**

ba. Inbound Filter: **inbound**

bb. Outbound Filter: **outbound**

bc. **SAVE CHANGES**

16. Hub Edge Deployment

> **Note:** For an ESXi installation, follow a process similar to Step 4a for the Orchestrator. Mount an ISO instead of using the OVA wizard template and use the set_wan_config.sh command to set WAN IP addresses.

a. Login to vCenter

b. Deploy OVF Template.

c. Choose Edge (that is, VCE) OVA file from local

d. Provide an appropriate name, select an appropriate location

e. Select an appropriate compute resource

f. **Review**

g. Select storage, **thick provision**

h. Select appropriate port groups:

1. GE1: dc1-lan-pg
2. GE2: dc1-lan-pg
3. GE3: dc1-inet-pg

i. Next

j. Template values:

1. Orchestrator address: `10.10.10.10`
2. Activation code: **(paste from previous)**
3. Check the box to ignore Orchestrator certificate validation errors
4. Default Users Password: *Velocloud123*
5. DNS1: `10.10.10.17`
6. DNS2: `10.10.11.17`
7. GE3 interface IPv4 allocation: **STATIC**
8. GE3 interface IPv4: `10.10.10.20`
9. GE3 interface IPv4 subnet mask: `255.255.255.240`
10. GE3 interface default gateway: `10.10.10.17`
11. GE4 interface IPv4 allocation: **STATIC**
12. GE4 interface IPv4: `10.10.11.20`
13. GE4 interface IPv4 subnet mask: `255.255.255.240`
14. GE4 interface default gateway: 10.10.11.17
15. **NEXT**

k. FINISH

l. Wait for OVF Deploy/Import tasks to complete

m. Power On VM

n. Login with *vcadmin*/*Velocloud123*

o. If needed (such as with an ESXi deployment), manually activate:

- `activate.py -i -s 1.1.10.10 <activation key>`

17. Spoke Edge Profile

a. Dark blue navigation bar, select **Global Settings** Dropdown, select **SD-WAN**

b. Top navigation bar, select **Configure**

c. Left Navigation bar, select **Profiles**

d. Select the checkbox next to Hub Profile

e. **Duplicate**

f. Name: **Spoke-Hybrid**

g. **Create**

h. Select **Device** tab

i. Expand **Connectivity** > **Interfaces**, select appropriate SD-WAN Edge models

j. Under **VPN Services**, check the box next to **Branch to Hub Site** (permanent VPN): Enable Branch to Hubs

k. On the right, select **Edit Hubs**

l. Select the check box next to *dc1-vce-1*, select the arrow that moves it rightward on the Hubs list

m. **UPDATE HUBS**

n. **SAVE CHANGES**

o. **Configure** > **Profiles**

p. Check the box next to **Spoke-Hybrid**

q. **DUPLICATE**

r. Name: **Spoke-Public**

s. **CREATE**

t. On the **Configure** > **Device** tab

u. Expand **Interfaces**

v. Select **GE4**

w. Change **Addressing Type** to **DHCP**

x. Change **WAN Link** to **Auto-Detect**

y. **SAVE**

z. **SAVE CHANGES**

aa. **Configure** > **Profiles**

ab. Check the box next to **Spoke-Hybrid**

ac. **DUPLICATE**

ad. Name: **Spoke-Private**

ae. **CREATE**

af. Back on the **Configure** > **Device** tab

ag. Expand the Interfaces section

ah. Select **GE3**

ai. Change Addressing Type: **DHCP**

aj. **SAVE**

ak. **SAVE CHANGES**

18. Public Spoke Staging

a. **Configure** > **Edges**

b. **+ADD EDGE**

c. Name: *s1*

d. Model: select as appropriate

e. Profile: **Spoke-Public**

f. Edge License: **POC**

g. **NEXT**

h. **ADD EDGE**

i. Expand **Loopback Interfaces**

j. **+ADD**

k. Interface ID: **1**

l. IPv4 address: 100.100.11.1

m. **ADD**

n. Expand the **Management Traffic** section

o. Change Source Interface: **Lo1**

p. Expand the **Interfaces** section

q. Select **GE1**

r. IPv4 Settings, IP Address: `10.20.11.1`

s. Cidr prefix: **24**

t. **SAVE**

u. Expand the **Interfaces** section

v. Select **GE3**

w. IPv4 settings, IP address: `1.2.2.2`

x. CIDR Prefix: **28**

y. Gateway: `1.2.2.1`

z. **SAVE**

aa. **SAVE CHANGES**

ab. Copy the activation key from the yellow bar at the top

ac. **Configure** > **Edges**

ad. Check the box next to **s1**

ae. **MORE**

af. Select **Assign Operator Profile**

ag. Change to **R5200-PUBLIC**

ah. **ASSIGN**

19. **Spoke Deployment**

Note: For an ESXi installation, follow a process similar to **Step 4a** for the Orchestrator. Mount an ISO instead of using the OVA wizard template and use the **`set_wan_config.sh`** command to set WAN IP addresses.

a. Login to vCenter

b. Deploy OVF Template

c. Choose the Edge (VCE) OVA file from local

d. Provide an appropriate name, select an appropriate location

e. Select an appropriate compute resource

f. **Review**

g. Select storage, **thick provision**

h. Select appropriate port groups

1. GE1: s1-lan-pg
2. GE2: s1-lan-pg
3. GE3: inet-s1-pg

i. Next

j. Template Values:

1. Orchestrator address: `1.1.10.10`
2. Activation code: (paste from section 14 step 21)
3. Check the box to ignore VCO (Orchestrator) certificate validation errors
4. Default Users Password: *Velocloud123*
5. DNS1: `1.2.2.1`
6. DNS2: `1.2.2.1`

7. GE3 interface IPv4 allocation: **STATIC**
8. GE3 interface IPv4: `1.2.2.2`
9. GE3 interface IPv4 subnet mask: `255.255.255.240`
10. GE3 interface default gateway: `1.2.2.1`
11. **NEXT**

k. **FINISH**

l. Wait for the OVF Deploy/Import tasks to complete

m. **Actions** > **Edit Settings**

n. Uncheck the box for **Connected** for Network Adapter 4

o. Power On VM

p. Login with vcadmin/Velocloud123

q. If needed, manually activate as follows:

- **`activate.py -i -s 1.1.10.10 <activation key>`**

Chapter 10

Federal Information Processing Standards (FIPS) Guide

This section provides an overview of the implementation of Federal Information Processing Standard (FIPS) on the Orchestrator and Gateways, encompassing the three operational modes and the procedures necessary to activate FIPS mode.

Overview

The Federal Information Processing Standards of the United States are a set of publicly announced standards that the National Institute of Standards and Technology (NIST) has developed for use in computer systems of non-military United States government agencies and contractors.

FIPS 140-2 certification for cryptographic modules enables organizations to meet compliance requirements within the public sector, healthcare, and finance industries. It defines the critical security parameters that must be used for encryption in the products sold into the U.S. public sector. FIPS 140-2 is, therefore, required under multiple compliance programs, such as Federal Risk and Authorization Management Program (FedRAMP), Federal Information Security Management Act of 2002 (FISMA) and the Health Information Technology for Economic and Clinical Health Act (HITECH).

FIPS certification ensures that software has been thoroughly reviewed and tested before being deployed and utilized within an agency or organization requiring data encryption. Industries storing and processing sensitive data that spans outside the public sector space, leading to FIPS-certified software being widely adopted within the payment card industry, healthcare, and other regulated industries.

System Components

FIPS-validated modules for the SD-WAN Orchestrator and Gateway include the following system components:

- The cryptographic library in the system kernel.
- OpenSSL general purpose cryptographic library that includes TLS implementation.
- OpenSSH client and server applications.

FIPS Cryptographic Algorithms

Ciphers:

AES128-CTR, AES192-CTR, AES256-CTR

MACS:

HMAC-SHA2-256, HMAC-SHA2-512

Key Exchange:

EDCH-SHA2-NISTP521, ECDH-SHA2-NISTP384, ECDH-SHA2-NISTP256, Diffie-Hellman-Group-Exchange-SHA256

Application Security System

FIPS-mode operation also enables an application security system on the Orchestrator and Gateway. This includes an allowlist of core applications used by the SD-WAN appliances, and a denylist of processes that should never be allowed.

The application security system also prevents unauthorized access from within existing processes, such as file access, external executable processes, and so forth. Unauthorized access attempts are recorded in an audit log, which can be viewed using the **`aa-logprof`** CLI command.

FIPS Modes of Operation

Non-FIPS Mode

This is the default mode of operation. SD-WAN appliances are capable of using non-FIPS-approved ciphers such as Triple DES (3DES), and all software packages can be updated.

FIPS Compliant Mode

SD-WAN appliances are capable of only using FIPS-approved ciphers, and the software package can be updated to include a FIPS module that is patched for security vulnerability but not tested by NIST.

FIPS Strict Mode

SD-WAN appliances are capable of only using FIPS-approved ciphers, and only non-cryptographic software packages may be updated. For systems running in “Strict FIPS mode”, FIPS system components will not receive regular security updates. These modules will be updated when new validated FIPS modules are available.

Enabling FIPS Mode

The Orchestrator and Gateways are appliances that run as virtual machines. FIPS mode may be enabled for them during instantiation using a cloud-init file, or after instantiation using a CLI command.

Important: Enabling FIPS mode has an impact on system operations. Therefore, it should be enabled before the system is put into production when there are no active Edges or Gateways. This is because the enablement process will install FIPS-compliant software modules and reboot the system to transition into FIPS mode. This requires a cryptographic rekey to be performed after reboot, which is service impacting to any attached Edges or Gateways.

However, there is no impact to an Orchestrator already in FIPS mode if a user transitions it from “Strict” to “Compliant”.

Enabling FIPS via Cloud-Init File

For both the Orchestrator and Gateways, the mode of operation (FIPS mode versus non-FIPS mode) can be specified in cloud-init during the VM spin up process. If no flags are set, the system will default to non-FIPS mode.

The line `fips_mode: strict` should be added to the user-data portion of the cloud-init file as shown in the example below:

```
#cloud-config velocloud: fips_mode: strict
```

If no flags are set, the system will instantiate in the “default” mode of “FIPS disabled”.

Enabling FIPS via the Command Line

FIPS mode can also be enabled via CLI post-deployment (not recommended). Please note that SSH host keys will change.

```
sudo /opt/vc/bin/vc_fips_enable --mode=[strict|compliant] [--no-reboot] [--noninteractive]
```

Verifying FIPS Mode of Operation on SD-WAN Orchestrator and Gateway

Three CLI commands may be used to verify which mode of operation the SD-WAN Orchestrator and Gateway appliances are operating in. These are shown in the following example:

```
$ cat /opt/vc/etc/fips/fips.json {"fips_mode": "compliant", "prev_fips_mode": "none"} $ cat /proc/
sys/crypto/fips_enabled 1 $ dmesg |grep -i "fips mode" [ 0.000000] fips mode: enabled
```

Orchestrator Upgrade

If the Orchestrator fails to upgrade due to a FIPS error, first check if the `fips.cnf` file exists at `/etc/mysql/conf.d`.

If the file is missing, do the following:

With sudo, create a file `/etc/mysql/conf.d/fips.cnf` with:

```
[mysqld]
ssl_fips_mode=1
sudo chmod 644 /etc/mysql/conf.d/fips.cnf
sudo systemctl restart mysql.service
```

Chapter 11

External Certificate Authority Design and Deployment on an On Premises Orchestrator

This guide covers the use of certificates and the expected functions of an external certificate authority (CA). We will cover what an external CA is and what it does, the different types of certificates and the requirements for each, and some tips for troubleshooting if you run into any issues. All these elements are discussed in the context of an Orchestrator deployed on-premises.

Note: This guide is intended as a supplement to the "External Certificate Authority" documentation provided in the Operator Guide, which describes how to configure External CA in the Orchestrator.

The Use of Certificates in Arista SD-WAN

Device certificates are issued to SD-WAN Edges and Gateways and bind a public encryption key to an entity. They are used to establish trust between devices by verifying the authenticity of the public key presented by each device.

In Arista SD-WAN, certificates are used for mutual authentication during the establishment of VCMP overlay tunnels between SD-WAN endpoints. For example, Edge-to-Gateway, Edge-to-Hub, and Edge-to-Edge tunnels. This protects both data plane and management plane traffic traversing the SD-WAN overlay.

To protect management plane traffic, certificates are also inspected by the Orchestrator to authenticate Edge and Gateway client API calls. In Arista's hosted deployment model, the Arista Edge Cloud Orchestrator has an integrated Certificate Authority that is used to generate and renew device certificates. However, customers may require that device certificates be generated and maintained using their own CA. An on-prem Orchestrator supports integration with an external CA to address these requirements. The Orchestrator will then act as a pass-through device between the SD-WAN devices and the External CA. Note that integration with an external CA is only supported for Orchestrators deployed on-premises, and not when using Arista's hosted Orchestrators. The private key must be under Arista control to meet security and compliance requirements.

Expected Functions of an External CA

There are three functions that an External CA is expected to provide:

1. **Provide a rooted chain of trust in the form of one or more PEM-encoded certificates.**

   This chain of certificates is distributed by the Orchestrator to Edges and Gateways to allow for mutual, certificate based authentication during the establishment of SD-WAN overlay tunnels.

   The chain of trust is installed on the Orchestrator API Gateway (NGINX) allowing the Gateway to authenticate API calls from Edges and Gateways. All certificates in the chain must have:

   a. Basic Constraint: CA: TRUE

   b. Key Usage: CRITICAL

   c. Include CRL signing

d. Certificate Signing

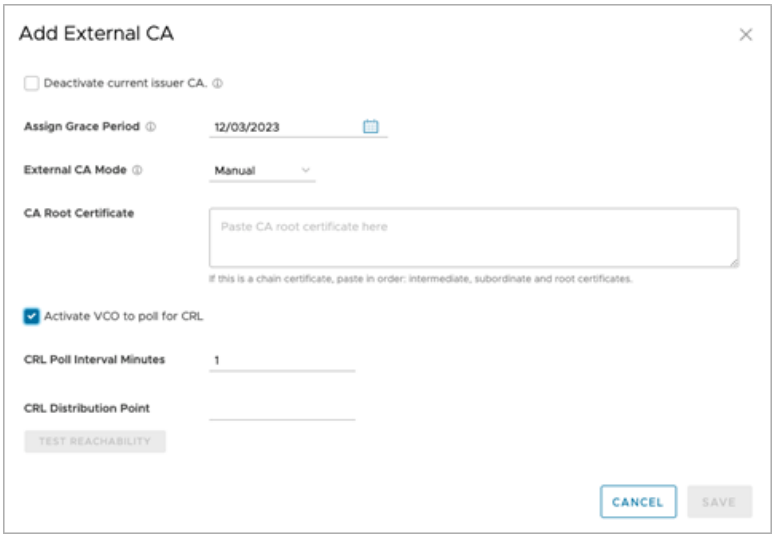
The chain must be continuous from a leaf certificate to a self-signed root certificate. The chain need not be rooted by a well-known public issuer, the root can be a private self-signed certificate.

2. **Provide signed certificate revocation lists (CRLs) to the Orchestrator.**

   CRLs generated by the customer's external CA are distributed by the Orchestrator to Edges and Gateways. Edges and Gateways do not communicate directly with the external CA. Also, the Orchestrator may be configured to automatically fetch CRLs from a CRL distribution point, or CRLs may be manually updated to the Orchestrator.

   Configure Orchestrator polling when creating an External CA

   Figure 11-1: Add External CA



3. **Generate certificates on request for SD-WAN Gateways and Edges in response to CSRs generated by those devices.**

   Edges and Gateways generate RSA key pairs and CSRs when certificate enrollment or renewal is requested. The Orchestrator receives the CSR via the device heartbeat. From the Orchestrator the CSR can be delivered by several mechanisms to the customer CA, receiving a signed certificate in return.

Certificate Types and Requirements

There are two types of certificates, one for Edges, and the second for Gateways. Only the Common Name (CN) is reserved for Arista SD-WAN use.

The Edge Certificate:

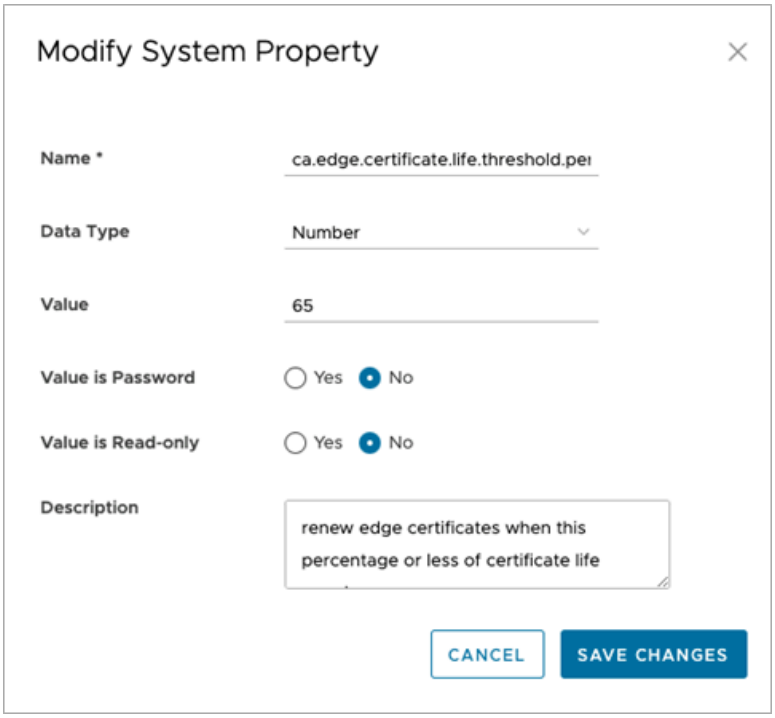
- It is composed of the literal string gateway concatenated with no spaces to the Gateway Logical-ID.
- The Gateway Logical-ID is a globally unique identifier the Orchestrator generates to identify the Gateway.
- Example: Subject: CN = gateway6f9589ca-7946-4b85-a2c2-be6e03f8a4c5

Only one CN can be defined. Other fields in the Distinguished Name (DN) are free for use by the external CA. For help in obtaining the Edge and Gateway Logical-IDs, see the section below "Locating Logical-IDs".

Gateway and Edge Certificate Requirements

- 2048-bit or 4096-bit RSA keys
- X509v3 Extended Key Usage: Web Server Authentication, TLS Web Client Authentication, Key Usage = TLS Web Server & TLS Web Client
- Certificates must be valid as of the time they are distributed to the Edge or Gateway. The validFrom time must be the current time or earlier.
- Certificate lifetime is delegated to the external CA according to established customer policy.
  - The Orchestrator automatically renews certificates when a configurable fraction of their lifetime has passed.
    - The default value is 33%, meaning that when 1/3 of the certificate's lifetime has passed, the Orchestrator will request that the Edge or Gateway generate a new CSR to trigger the generation of a new certificate.
    - The lifetime fraction for renewal is configurable on the Orchestrator.
    - Renewal can also be windowed so that certificate generation is only done within a defined maintenance window.

Figure 11-2: Configure Certificate Lifetime for all Edges



Example: Certificate and CSR Subject Names (DNs)

The Edge and Gateway, once activated, build a fixed CSR subject name, with just the CN attribute defined:

- **Edge:**
  - Subject: CN = VC11130031

- **Gateway:**
  - Subject: CN = 00:50:56:81:7e:2d

Integration with an external CA requires flexibility in both the generation of CSR subject names and in the handling of certificate subject names by both the Management Plane and the Data Plane.

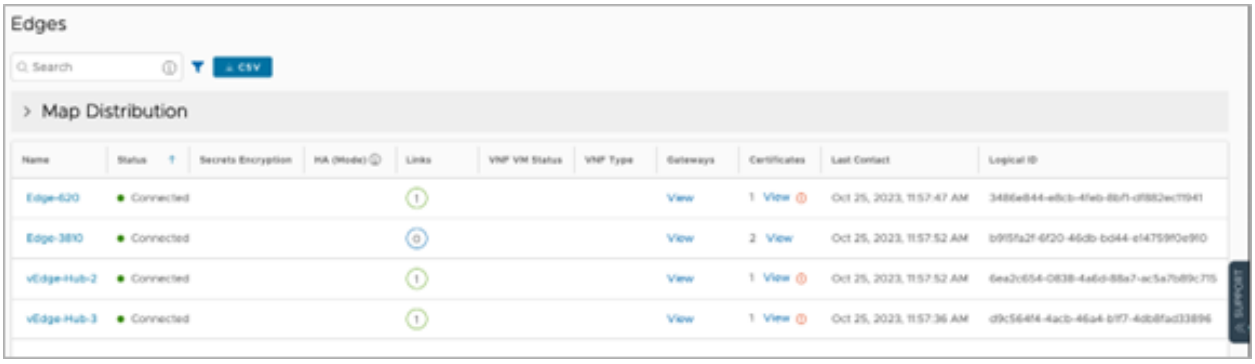
Note: In generating certificates, the external CA can appropriate the O and OU fields for their own purposes.

The logical IDs used in certificate generation from CSRs may be obtained from the following locations:

- **Edge Logical ID**

  In the Orchestrator, the Edge Logical-ID may be found in the **Logical ID** column by going to **Monitor** > **Edges**. If it is not displayed, select **Columns** at the bottom of the screen and select it.

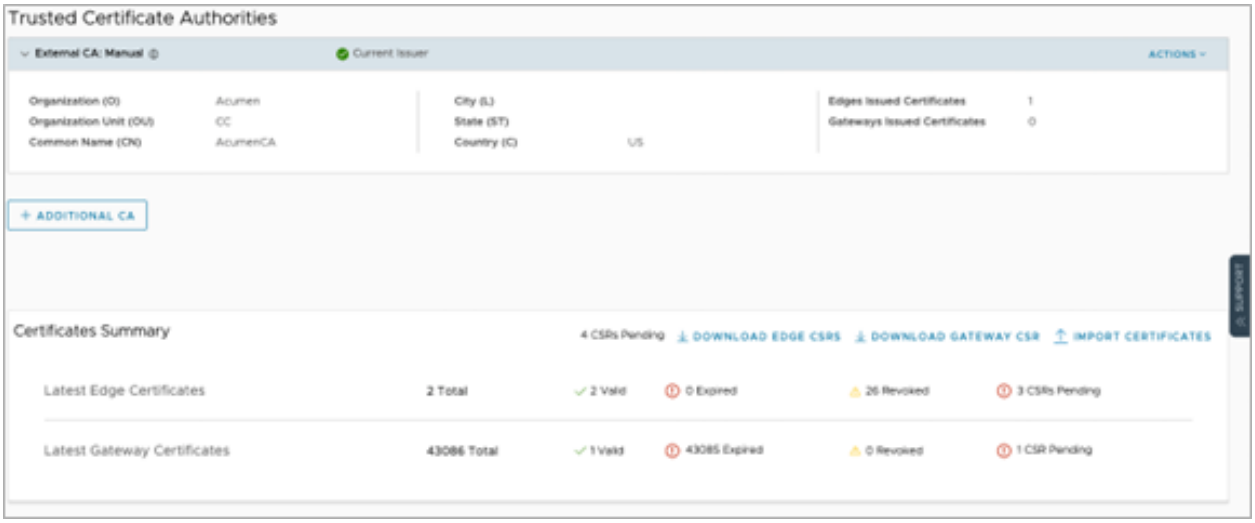
  Figure 11-3: Locating the Edge Logical ID



- **Gateway Logical ID**

The Gateway Logical ID may be found on the Orchestrator UI by logging in as an Operator level user and navigating to **Gateway Management** > **Gateways**. Select the desired Gateway, then select the drop-down next to the Gateway name to obtain details about the Gateway, including its logical ID.

Figure 11-4: Locating the Gateway Logical ID



Orchestrator to External CA Communication

Two integration mechanisms are supported on the Orchestrator:

- Synchronous API-based integration with an EJBCA CA.
- Manual Mode, where transmission of CSRs and certificates are performed by users or by middleware using Orchestrator API calls.

1. Expose a rooted chain of trust in the form of one or more PEM-encoded certificates.

The PEM-encoded chain can be manually configured on the Orchestrator at the time of external CA definition on the Orchestrator, in the Orchestrator UI. The chain can be pulled from the External CA, or from customer middleware by API at the time of External CA definition on the Orchestrator.

2. Provision of signed certificate revocation lists.

CRLs may be manually pushed to the Orchestrator via API, manually uploaded to the Orchestrator from the Orchestrator UI, or pulled by the Orchestrator from a CRL distribution point on a configured schedule. The CRL distribution point is defined by an HTTP URL visible to the Orchestrator, and is configured when the CA is defined.

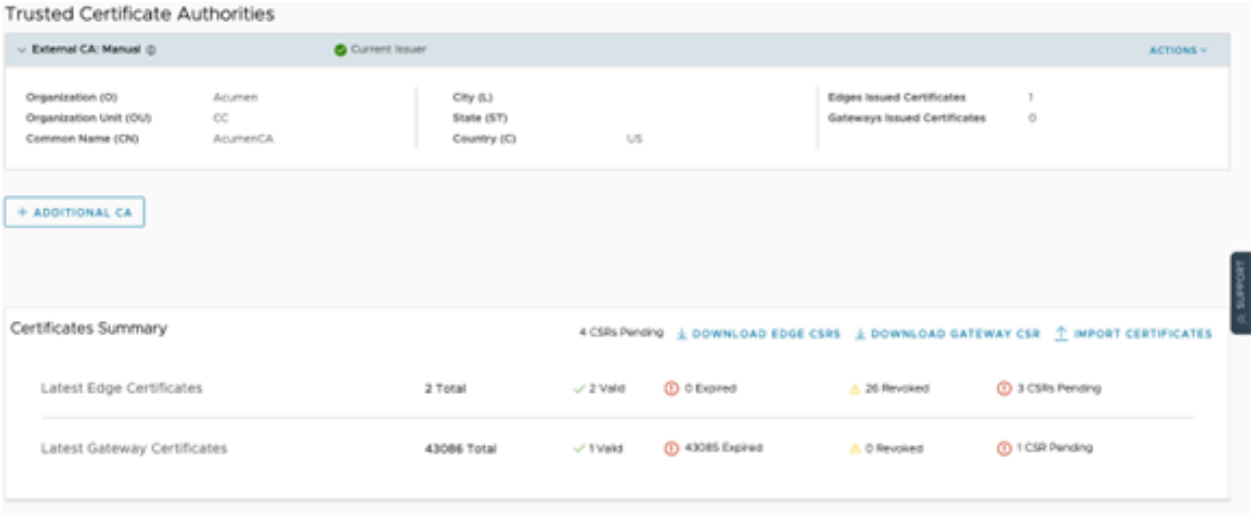
3. Certificate Enrollment

Certificate enrollment can be manual:

- The CSR is downloaded by a customer administrator from the Orchestrator UI and delivered to the external CA for certificate generation.

- While logged in as an Operator-level user, in the Orchestrator UI, navigate to **Certificate Authorities** > **Certificates Summary**. Here, you can select **DOWNLOAD EDGE CSRs** or **DOWNLOAD GATEWAY CSRs**.

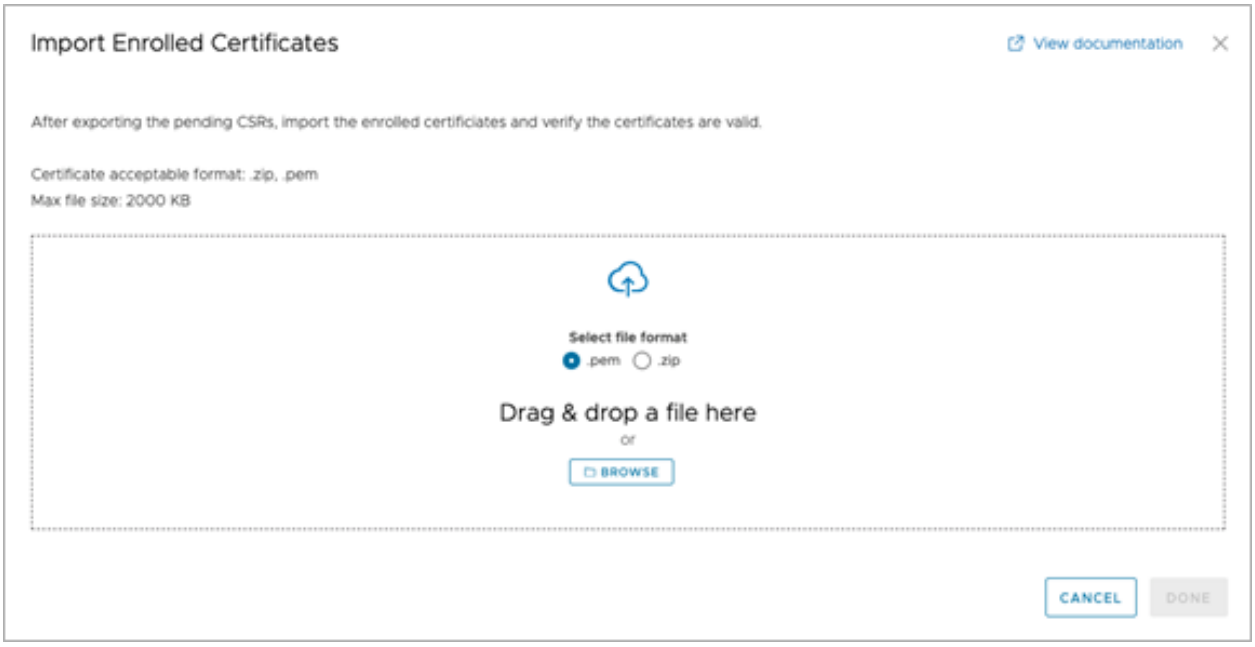
Figure 11-5: Manual CSR Download for Edge or Gateway



- These CSRs may then be used to manually create a certificate in the external CA using the criteria described above.

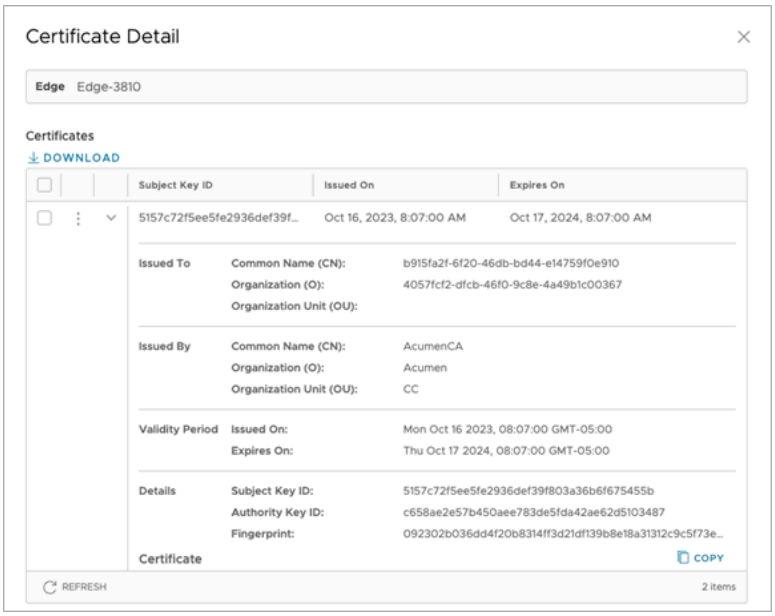
- The certificate generated by the eCA from the CSR is uploaded to the Orchestrator. It is then automatically distributed by the Orchestrator to the Edge or Gateway.
- This is done in the UI by navigating to **Certificate Authorities** > **Certificates Summary** and selecting **IMPORT CERTIFICATES**.

Figure 11-6: Import Enrolled Certificates



- The certificates currently assigned by the Orchestrator to Edges and Gateways may be viewed and downloaded by navigating to **Certificate Authorities** > **Certificates** and selecting **View Certificate** next to the desired Edge or Gateway.
- Key details of the issued certificate are displayed on that screen.

Figure 11-7: Certificate Detail for an Edge



- Selecting **Copy** will copy the raw text of the certificate to the clipboard, if desired.

Certificate enrollment can be automated by customer middleware:

- Middleware pulls CSR(s) by API.
- Middleware then pushes certificates back by API.

Certificate enrollment can be automated by Arista custom integration with common 3rd party Certificate Authorities:

- The Orchestrator pushes the CSR to an external CA using 3rd party API integration.
- The Orchestrator receives a certificate on the return of API call.
- PrimeKey EJBCA Enterprise edition is currently the only ‘out of the box’ integration supported. Microsoft CA Server integration is currently under consideration.

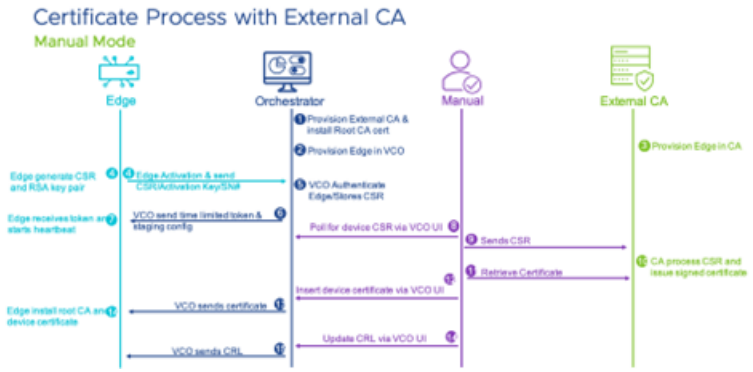
Certificate enrollment can also be automated by a custom integration developed in conjunction with Arista Professional Services.

External CA Modes

Manual Mode

The Orchestrator presents a UI allowing the user to poll/request CSR, install root CA's certificate and device certificate, and CRLs. This process is very similar to the Middleware mode using Southbound API, but instead of APIs, users interface with Orchestrator through UI.

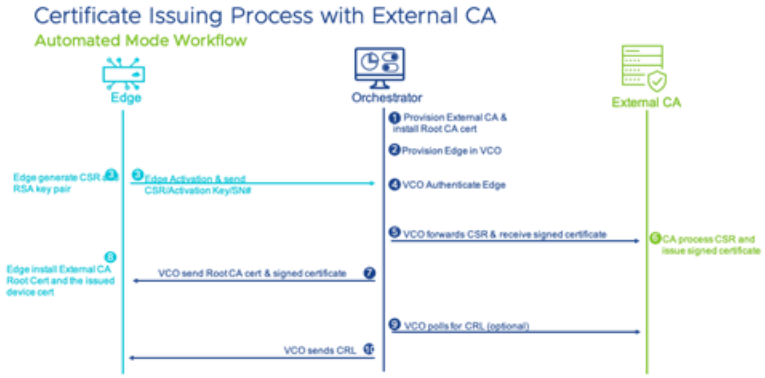
Figure 11-8: Manual Mode Certificate Process with External CA



Automated Mode

The Orchestrator integrates directly with an external CA (EJBCA, MS CA, etc) through REST APIs for certificate request, renewal, and revocation.

Figure 11-9: Automated Mode Certificate Process with External CA



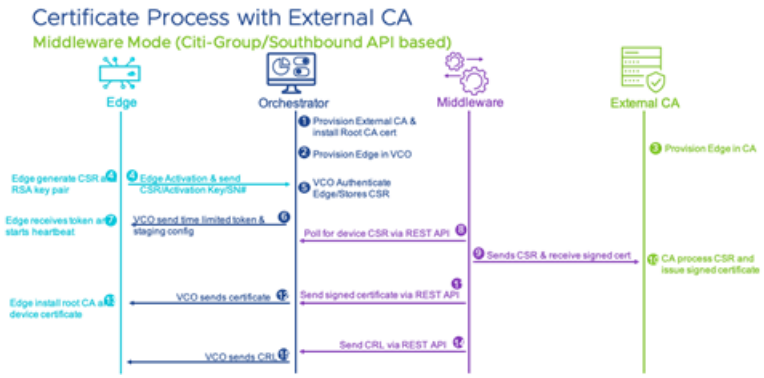
Middleware (Asynchronous) Mode

The Orchestrator integrates with an automation platform (for example, Cisco NSO), and the automation platform will interface with external CA and Orchestrator for certificate request, renewal, and revocation. There are Northbound and SouthBound APIs for integration with middleware and the function varies depending on whether northbound or southbound APIs are used.

- **Southbound API**- This is where the Orchestrator receives REST APIs from the Middleware system for the certificate operations (CSR, issue, renewal, and so forth). The Middleware system in this case will be an automation engine where the device provisioning and configuration is done, and this includes certificate operations. Citi-group is the customer that uses this process.

(Southbound API based)

Figure 11-10: Middleware Certificate Process with External CA



- **NorthBound API**- This is where the Orchestrator initiates all certificate operations and uses a Middleware system as a proxy and protocol translation to the External CA server. An example of why a customer would use this method is the External CA does not support REST API and requires the Middleware system for protocol translation from the REST API (from the Orchestrator) to something that an external CA can support.

Troubleshooting

This section offers guidance on how to resolve common issues experienced when deploying and operating an external CA.

Edges deployed in High Availability (HA)

When two Edges are deployed in HA, there will be two certificates shown under a single Edge in **Certificate Authority** > **Certificates** > **Latest Edge Certificates**. In manual mode, a separate certificate must be generated by the external CA using the CSRs for the Active and Standby Edges.

Edges starting from the 5.1.0 version can enable HA as they are compatible with External CA. Edges running version 5.0.x or earlier software are not supported with external CA.

Connection errors due to certificate mismatches

If an Edge or Gateway fails to connect, it may be the result of a certificate mismatch. The first step is to check if the Edge or Gateway has **Authentication Mode** set to **Certificate Acquire**. A certificate mismatch can be indicated by logging into the CLI of the Edge or Gateway and viewing the output of the `mgd.log` file using the **`sudo tail -f /var/log/mgd.log`** command.

The certificates that have been pushed to the Edge and Gateway can also be decoded using the **openssl** utility in Linux and compared to the certificate in the Orchestrator. These certificates, as well as PKI keys derived from them, are stored on the Edge and Gateway in `/etc/vc_private and /etc/vc_public`.

Utility: debug.py --pki

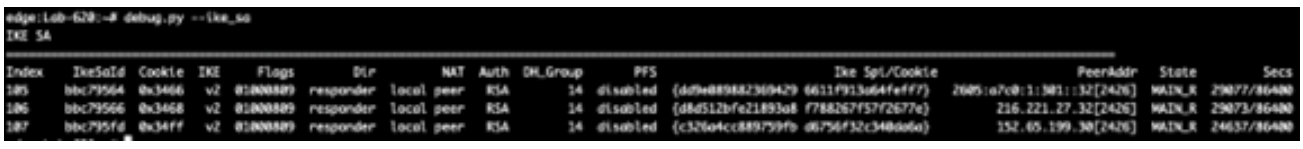
The **`debug.py –pki`** utility may be used from the Edge or Gateway command line interface to view details about PKI settings, the trusted CA list, and the certificate:

Figure 11-11: Example of the Debug Utility debug.py --pki

Utility: debug.py --ike_sa

The **`debug.py --ike_sa`** utility can be used to show the details of current security associations. In the example below, the three existing security associations correspond to the Orchestrator and Primary and Secondary Gateways, respectively:

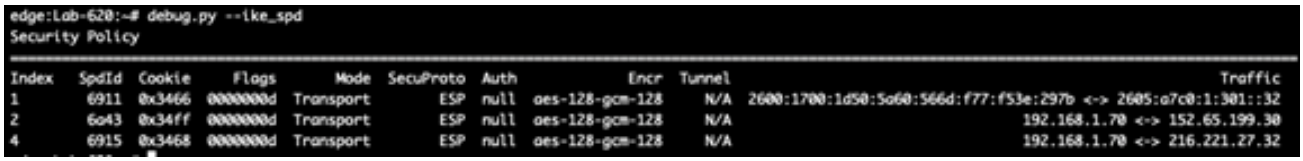
Figure 11-12: Example of the Debug Utility debug.py --ike_sa



Utility: debug.py --ike_spd

Details of security protocols and encryption methods used per peer can be found using the **`debug.py --ike_spd`** utility:

Figure 11-13: Example of the Debug Utility debug.py --ike_spd



Decoding Certificates using OpenSSL

The details and contents of certificates may be viewed by using the Linux openssl tool. It will show you the details of the certificate like Version, Serial Number, Signature Algorithm, Issuer, Validity, Subject, Public Key Algorithm and many more.

Some examples:

- The full certificate may be seen by using the following command, where `cert.pem` is the file name of the certificate to be parsed out:
  - `openssl x509 -text -noout -in cert.pem`
- Specific fields within the certificate may be verified by piping the command to grep. For example:
  - ```
    openssl x509 -text -noout -in cert.pem | grep -a1 Basic openssl x509 -text -noout -in
     ICA_from_Acumen.pem | grep -a1 "Subject Key" openssl x509 -text -noout -in ICA_from_Acum
    en.pem | grep -a1 "Authority Key" openssl x509 -text -noout -in ICA_from_Acumen.pem | grep -
    a1 X509v3
    ```

Certificate Authority Renewal

In the event of a CA compromise or the normal renewal of a CA, the Orchestrator includes the ability to renew the External CA. This is done by using the **+ Add CA** function.

When the new CA is added the user has the option to retain the existing CA for continuity purposes for a period of time. When configuring a new CA, Edges and Gateways will be provisioned certificates from the newer CA and, depending upon the renewal interval, all Edges and Gateways will have certificates from the new CA. At this point the original CA can be decommissioned from the **CA Summary** section of the Orchestrator UI.
    ```

Chapter 12

References

12.1 Related Documents

The following documentation is available for ***Arista VeloCloud SD-WAN***:

- *Arista VeloCloud SD-WAN Operator Guide*
- *Arista VeloCloud SD-WAN Administration Guide*
- *Arista VeloCloud SD-WAN Gateway Monitoring Guide*
- *Arista VeloCloud SD-WAN Bastion Orchestrator Configuration Guide*
- *Arista VeloCloud SD-WAN Partner Guide*
- *Arista VeloCloud SASE Global Settings Guide*
- *Arista VeloCloud SD-WAN Design Guide for Enhanced Firewall Services*
- *Arista VeloCloud SD-WAN API*
- *Arista VeloCloud Portal API*