

ARISTA

Quick Start Guide

Dipl.-Ing. Martin Krausch

Systems Engineer

Arista Networks Germany

Quick Start Guide version 5.3.0

June 2022

Table of Contents

Preface	8
Management Configuration	9
Default Console Username and Password:	9
Changing Username and Password	9
Configuring Enable Password	9
Useful navigation & show commands	9
Telnet Access to another device from an Arista system	11
Disabling ZTP Process	11
Configuring OOB and a Default Route within Management VRF	12
Enabling Telnet and SSH Access:	12
Changing the hostname	12
Saving config to Startup-config	13
Saving config to a file	13
Deleting the startup-config file	13
Deleting a config file	14
Replacing a config file	15
Configuration Checkpoints	18
Generating a Certificate and activating it on EOS	19
Accessing the Linux system	21
Configuration a range of parameters	22
Configuring sFlow	23
Software Upgrade	24
Using USB Stick	24
Using FTP	24
Accelerated Software Upgrade (ASU)	27
Smart System Upgrade (SSU)	27
Maintenance Mode	27
Installing Extensions	31
Upgrading TerminAttr	31
Port Configuration	33
Configuring the description and Port Speed:	33

Configuring Breakout Ports	33
Configuring an IP address on a Port	34
Configuring a Port Profile	35
Configuring a VLAN	36
Adding a VLAN:	36
Assigning an IP address to a VLAN	36
Assigning a tagged Port to a VLAN	36
Assigning an untagged Port to a VLAN:	36
Configuring a Voice VLAN	37
Configuring a Port Channel	38
IEEE 802.1Q tagged LACP BPDUs:	38
Configuring MLAG	39
Example of how to configure an MLAG	39
MLAG verification	42
Configuring MSTP	44
Default setup	44
Example of how to configure MSTP	45
Configuring Link State Tracking	49
Configuring VRRP	50
Configuring Virtual ARP (VARP)	51
Configuring OSPF	52
Checking OSPF Connectivity	53
QoS Configuration	54
Configuring Policy Maps	54
Assigning a traffic class to a tx-queue	54
Configuring a QoS profile	54
QoS Classification for Queue assignments	55
Configuring ACL based QoS	55
Configuring Interfaces for QoS	55
Multi-Domain Segmentation Services (MSS)	57
Configuring MSS-G	57
Configuring MSS-G on CloudVision with Studios “Segment Security”	62

Testing MSS-G	64
Monitoring MSS-G in CloudVision	66
MACsec Configuration	68
Configuring MACsec	68
Checking MACsec Configuration	71
Using eAPI	72
EOS setup	72
JSON output via CLI	72
eAPI Web Interface	74
Remote Python scripting	77
Ansible Automation	83
Basic Components of Ansible:	83
Installing Ansible on Ubuntu	85
Installing Python Client for eAPI (pyeapi)	87
Setting up EOS systems	88
Setting up an Inventory File	89
Checking connectivity to EOS devices via Ansible	91
Sending CLI commands to EOS devices via Ansible	93
Ansible Playbooks	96
CloudVision Management	101
CloudVision password recovery	101
EOS setup for management access	101
Adding a device to CloudVision	103
Removing a device from CloudVision	112
Snapshots:	114
Configlets	117
Importing Configlets:	118
Applying Configlets to a Container:	121
Adding parts of a configuration without reconciliation:	125
Traffic Flows	127
Integration of 3rd party devices into CloudVision	129
Configuration of an Extreme Networks XOS based 3rd party device:	129

Configuration of CloudVision	130
Configuration of the Collector Config for SNMPv2c:	136
Configuration of the Collector Config for SNMPv3	139
CloudVision ZTP based automatic onboarding	142
Enabling ZTP Process on EOS devices:	142
Setting up an DHCP Server	144
Starting & Stopping the DHCP Server	148
EOS & CloudVision ZTP Outputs	149
CloudVision Portal RESTful API Client (CVPRAC)	155
Installing CVPRAC on Ubuntu Linux	155
Example Python Script for reading information	156
Example Python Script creating a new Container within CloudVision	158
Configuring MAC based Client Authentication	160
Setting up a RADIUS Server	160
Setting up Arista EOS systems	163
IP Fabric	166
IP Fabric BGP Underlay Spine Layer	169
IP Fabric BGP Underlay Leaf-Range Peer-Filter	172
IP Fabric BGP Underlay Leaf Layer	173
IP Fabric BGP Underlay MLAG Leaf Layer	176
Checking Underlay BGP Connectivity:	179
IP Fabric EVPN Overlay Spine Layer:	183
IP Fabric BGP Overlay Leaf-Range Peer-Filter	185
IP Fabric EVPN Overlay Leaf Layer	187
Checking Overlay BGP Connectivity	191
IP Fabric L2 EVPN Overlay Leaf Layer	192
IP Fabric L2 EVPN Overlay Leaf Layer verification	195
IP Fabric L3 EVPN Overlay Leaf Layer for symmetric Routing	198
IP Fabric L3 EVPN Overlay Leaf Layer verification:	204
IPv6 BGP unnumbered for Fabric Underlay	212
Automated IP Fabric setup via Ansible AVD	214
Installing Python3-pip	215

Installing Ansible and additional Requirements	216
Cloning an Arista Demo Repository from Github	218
Installing Arista AVD	219
Installing newest Development versions	220
Example Lab Setup	221
Configuring the Inventory file	222
Adjusting the dc1-upload-configlets.yml file	224
Configuring the CVP.yml file	225
Configuring the DC1.yml file	227
Configuring the DC1_FABRIC.yml file	229
Configuring the DC1_SERVERS.yml file	233
Configuring the DC1_TENANTS_NETWORKS.yml file	235
Creating DHCP ZTP Configuration Files on CloudVision	237
Creating offline Config Files	243
Creating Config Files and Provisioning via CVP	249
Updating Config Files and Provisioning via CVP	260
Customized Playbooks with Arista AVD	269
Docker Virtualization on EOS	272
Differences between Docker Containers and VMs	272
Docker support on EOS	273
Installing Ubuntu in a Docker Container via Bash	275
Testing Docker via Bash	275
Creating a new Ubuntu Image and a Docker Container via Bash	276
Removing a Docker Container via Bash	276
Creating a new Ubuntu Container and configuring Ubuntu via Bash	276
Docker Networking Modes	278
Configuring a MACVLAN Interface on Docker and attaching it to an existing Container via Bash	279
Creating a new Ubuntu Container with the attached Macvlan Interface on Docker via Bash	283
Installing Ubuntu in a Docker Container via EOS CLI	284
Downloading an Ubuntu Image via EOS CLI	284
Creating a new Ubuntu Container via EOS CLI	284
Creating a Docker Container backup via EOS CLI	285
Deleting a backed up Docker Container via EOS CLI	285

KVM based Virtualization on EOS	286
Copying VM Image to the system	286
Creating a VM on EOS	287
Troubleshooting	288
Tech-Support Files	288
TCPDUMP Control Packet Sniffing	288
TCPDUMP Data Packet Sniffing	292
Sending TCPDUMP outputs directly to a remote Wireshark station	292

Preface

The purpose of this short manual is to help experienced Administrators or Field SEs, who basically know the different technologies from other vendors or platforms, to give a guideline how to configure these features concrete on an Arista EOS system.

Each Chapter gives a very short introduction about the feature and lots of real world CLI logs how to configure it.

If you are unfamiliar with certain features I strongly recommend reading the book "Arista Warrior" from Gary A. Donahue or having a closer look at Arista Manuals or the other tons of docs which can be found at www.arista.com or on the Internet.

I started writing this manual when I myself onboarded at Arista to document the configuration differences compared to the platforms I was used to working with and, above all, to write down possible pitfalls and useful hints when getting in touch with EOS based systems.

For most of the config examples I used GNS3[®] and an actual vEOS-lab vmdk Image but also EVE-NG which enables technicians to create and test quite complex network designs in a completely simulated test environment without the need to use dedicated Hardware.

This is a really huge advantage of Arista's EOS as this Linux based Operating System can be used completely virtualized even in Cloud Environments or White Boxes from 3rd party vendors.

Please be aware that when using newer software versions some commands may have changed over time and you will have to take care of this when using the concrete configurations used in this manual.

I do hope that this manual will also help other technicians getting up to speed when configuring EOS systems for the first time without digging through all the product manuals or tech docs in detail.

Sincerely,

Dipl.-Ing. Martin Krausch
Systems Engineer
Arista Networks Germany

Management Configuration

Default Console Username and Password:

Username: **admin**
Password: **No password (return)**

Changing Username and Password

```
localhost(config)#username admin role network-admin privilege 15 secret arista
```

Configuring Enable Password

```
localhost(config)#enable password secretpw
```

Useful navigation & show commands

It is not necessary to exit the configuration mode to run any commands or to put a “do” command in front!

```
Spine-1#conf t
Spine-1(config)#
Spine-1(config)#show run
! Command: show running-config
! device: Spine-1 (vEOS, EOS-4.22.2F)
!
! boot system flash:/vEOS-lab.swi
!
...
```

To display only the part of the “show run” which can be displayed within a terminal window output type:

```
localhost#show run | more
```

Typing “**end**” goes directly to the enable mode

```
Spine-2(config-if-Ma1)#end
Spine-2#
```

To display differences or changes within a “show” command output you may use the “watch diff” command so that changing values will be marked and the output will be updated every 2 seconds. **Control + c** will stop the output:

```
DC1-LEAF2A#watch diff show interface ethernet 5
```

```
Every 2.0s: CliShell -s ar -p 15 -c show interface ethernet 5    Fri Jun 12
08:53:38 2020
```

```
Ethernet5 is up, line protocol is up (connected)
  Hardware is Ethernet, address is 500f.0005.0005 (bia 500f.0005.0005)
  Description: MLAG_PEER_DC1-LEAF2B_Ethernet5
  Member of Port-Channel5
  Ethernet MTU 9214 bytes
  Full-duplex, Unconfigured, auto negotiation: off, uni-link: n/a
  Up 3 days, 18 hours, 38 minutes, 39 seconds
  Loopback Mode : None
  2 link status changes since last clear
  Last clearing of "show interface" counters never
  5 minutes input rate 0 bps (- with framing overhead), 0 packets/sec
  5 minutes output rate 0 bps (- with framing overhead), 0 packets/sec
    1262021 packets input, 96748755 bytes
    Received 56 broadcasts, 21768 multicast
    0 runts, 0 giants
    0 input errors, 0 CRC, 0 alignment, 0 symbol, 0 input discards
    0 PAUSE input
    620054 packets output, 52046199 bytes
    Sent 294 broadcasts, 21766 multicast
    0 output errors, 0 collisions
    0 late collision, 0 deferred, 0 output discards
    0 PAUSE output
```

To display the number of lines of an show command output you may use the “| wc” command. In this example you may want to know how many arp entries are currently active on the EOS system:

```
CORE-1#show arp
Address          Age (sec)  Hardware Addr  Interface
192.168.101.204  0:48:24   504e.007d.a3b8  Vlan121, not learned
192.168.122.204  1:26:33   504e.007d.a3b8  Vlan122, not learned
10.8.1.102       3:32:03   504e.0061.9a03  Vlan210, Port-Channel100
10.9.230.102     0:00:04   504e.0061.9a03  Vlan230, Port-Channel100
10.9.230.201     3:51:54   504e.005b.0272  Vlan230, not learned
10.9.230.202     3:27:54   504e.006f.3d87  Vlan230, not learned
10.11.247.102    3:32:03   504e.0061.9a03  Vlan247, Port-Channel100
10.12.1.204      2:51:43   504e.0007.1de6  Vlan501, not learned
10.12.101.204    2:22:34   504e.0007.1de6  Vlan601, not learned
172.16.2.1       0:00:00   504e.0061.9a03  Vlan4094, Port-Channel100
CORE-1#show arp | wc -l
11
CORE-1#
```

In sum there are 10 ARP entries. When using the command “show arp | wc -l” you get the information that the number of lines of the output from the show command would be 11 lines. As the first line is representing the header you can easily derive that **11 - 1 = 10 ARP entries** are currently active.

Telnet Access to another device from an Arista system

```
localhost#show vrf
Maximum number of vrfs allowed: 1024
  Vrf      RD          Protocols      State          Interfaces
-----
  MGMT     <not set>    ipv4,ipv6      v4:no routing, v6:no routing Management1
```

```
localhost#routing-context vrf MGMT
localhost(vrf:MGMT)#telnet 192.168.10.71
Trying 192.168.10.71...
Connected to 192.168.10.71.
Escape character is 'off'.
Username: admin
Password:
Last login: Tue Oct  8 08:45:57 on ttyS0
Spine-2>
Spine-2>exit
Connection closed by foreign host.
localhost(vrf:MGMT)#
```

Note:

The command "routing-context" is deprecated by "cli vrf [VRF_ID]" in actual EOS versions

Disabling ZTP Process

By default ZTP is enabled on all Arista systems!

When disabling ZTP the system will reboot automatically!

```
localhost>en
localhost#zerotouch disable
Oct  4 12:37:53 localhost ZeroTouch: %ZTP-6-CANCEL: Cancelling Zero Touch
Provisioning
localhost#Flushing AAA accounting queue: [ OK ]

Restarting system [12:37:57] watchdog punch .
```

Configuring OOB and a Default Route within Management VRF

Static IP address:

```
vrf instance MGMT
!
interface Management1
  vrf MGMT
  ip address 192.168.10.70/24
!
ip route vrf MGMT 0.0.0.0/0 192.168.10.1
!
ip routing vrf MGMT
```

Verifying that the OOB Management Port can reach other networks:

```
localhost#ping vrf MGMT 192.168.30.33
PING 192.168.30.33 (192.168.30.33) 72(100) bytes of data.
 80 bytes from 192.168.30.33: icmp_seq=1 ttl=63 time=1.45 ms
 80 bytes from 192.168.30.33: icmp_seq=2 ttl=63 time=0.409 ms
 80 bytes from 192.168.30.33: icmp_seq=3 ttl=63 time=0.698 ms
 80 bytes from 192.168.30.33: icmp_seq=4 ttl=63 time=0.686 ms
 80 bytes from 192.168.30.33: icmp_seq=5 ttl=63 time=0.635 ms

--- 192.168.30.33 ping statistics ---
 5 packets transmitted, 5 received, 0% packet loss, time 4ms
 rtt min/avg/max/mdev = 0.409/0.776/1.456/0.357 ms, ipg/ewma 1.162/1.109 ms
localhost#
```

Enabling Telnet and SSH Access:

SSH is **enabled** by default but you have to configure a username/password to login:

```
localhost(config)#username admin role network-admin privilege 15 secret arista
```

Telnet is **disabled** by default but you have also to configure a username/password to login:

```
localhost(config)#username admin role network-admin privilege 15 secret arista
localhost(config)#management telnet
localhost(config-mgmt-telnet)#no shutdown
```

Changing the hostname

```
localhost#conf t
localhost(config)#hostname Spine-1
Spine-1(config)#
```

Saving config to Startup-config

```
test#write memory
Copy completed successfully.
```

Or:

```
localhost#copy running-config startup-config
Copy completed successfully.
```

Saving config to a file

```
localhost#copy running-config flash://baseline-config
Copy completed successfully.
localhost #dir
Directory of flash:/
```

-rw-	924	Oct 4 13:03	AsuFastPktTransmit.log
drwx	4096	Oct 4 10:56	Fossil
-rw-	568	Oct 4 13:03	SsuRestore.log
-rw-	568	Oct 4 13:03	SsuRestoreLegacy.log
-rw-	626	Oct 4 13:14	baseline-config
-rw-	24	Sep 26 22:56	boot-config
drwx	4096	Oct 4 13:04	debug
drwx	4096	Oct 4 10:56	fastpkttx.backup
drwx	16384	Sep 26 22:56	lost+found
drwx	4096	Oct 4 13:14	persist
drwx	4096	Oct 4 11:01	schedule
-rw-	678	Oct 4 13:11	startup-config
-rw-	418919380	Sep 26 22:56	vEOS-lab.swi
-rw-	13	Oct 4 12:37	zerotouch-config

4093313024 bytes total (3229323264 bytes free)

Deleting the startup-config file

```
localhost #write erase
Proceed with erasing startup configuration? [confirm]y
cleanup None None
localhost #dir
Directory of flash:/
```

-rw-	2079	Oct 8 08:24	AsuFastPktTransmit.log
drwx	4096	Oct 4 10:56	Fossil
-rw-	1278	Oct 8 08:23	SsuRestore.log
-rw-	1278	Oct 8 08:23	SsuRestoreLegacy.log
-rw-	24	Sep 26 22:56	boot-config
drwx	4096	Oct 4 14:54	debug
drwx	4096	Oct 4 10:56	fastpkttx.backup
drwx	16384	Sep 26 22:56	lost+found
drwx	4096	Oct 8 08:25	persist
drwx	4096	Oct 4 11:01	schedule
-rw-	0	Oct 8 08:28	startup-config
-rw-	418919380	Sep 26 22:56	vEOS-lab.swi

```
-rw-          13          Oct 4 12:37 zerotouch-config
```

```
4093313024 bytes total (3229126656 bytes free)
```

Deleting a config file

```
localhost #dir
```

```
Directory of flash:/
```

```
-rw-          924          Oct 4 13:03 AsuFastPktTransmit.log
drwx         4096          Oct 4 10:56 Fossil
-rw-          568          Oct 4 13:03 SsuRestore.log
-rw-          568          Oct 4 13:03 SsuRestoreLegacy.log
-rw-          626          Oct 4 13:14 baseline-config
-rw-          24          Sep 26 22:56 boot-config
drwx         4096          Oct 4 13:04 debug
drwx         4096          Oct 4 10:56 fastpkttx.backup
drwx        16384          Sep 26 22:56 lost+found
drwx         4096          Oct 4 13:14 persist
drwx         4096          Oct 4 11:01 schedule
-rw-          678          Oct 4 13:11 startup-config
-rw-       418919380          Sep 26 22:56 vEOS-lab.swi
-rw-          13          Oct 4 12:37 zerotouch-config
```

```
4093313024 bytes total (3229323264 bytes free)
```

```
localhost #del flash://baseline-config
```

```
localhost #dir
```

```
Directory of flash:/
```

```
-rw-          924          Oct 4 13:03 AsuFastPktTransmit.log
drwx         4096          Oct 4 10:56 Fossil
-rw-          568          Oct 4 13:03 SsuRestore.log
-rw-          568          Oct 4 13:03 SsuRestoreLegacy.log
-rw-          24          Sep 26 22:56 boot-config
drwx         4096          Oct 4 13:04 debug
drwx         4096          Oct 4 10:56 fastpkttx.backup
drwx        16384          Sep 26 22:56 lost+found
drwx         4096          Oct 4 13:14 persist
drwx         4096          Oct 4 11:01 schedule
-rw-          678          Oct 4 13:11 startup-config
-rw-       418919380          Sep 26 22:56 vEOS-lab.swi
-rw-          13          Oct 4 12:37 zerotouch-config
```

```
4093313024 bytes total (3229327360 bytes free)
```

Replacing a config file

EOS allows replacing a running config file **without a reboot**.

All services which are not affected by any changes will continue to operate!

```
Spine-1#show run
! Command: show running-config
! device: Spine-1 (vEOS, EOS-4.22.2F)
!
! boot system flash:/vEOS-lab.swi
!
transceiver qsfp default-mode 4x10G
!
hostname Spine-1
!
spanning-tree mode mstp
!
no aaa root
!
username admin privilege 15 role network-admin secret sha512
$6$e2J8z/HDC03RmLPu$i02zAYE/7w0fySAKI0PbgL7r6PA6IUFSH6Bf7puI1Zi11eUP1bxQgr73JHItjH
vkGyJMj5X1vR0xvRp7E9iPf.
!
vrf instance MGMT
!
interface Ethernet1
!
interface Ethernet2
!
interface Ethernet3
  no switchport
  ip address 192.168.20.1/24
!
interface Ethernet4
  no switchport
  ip address 192.168.22.1/24
!
interface Ethernet5
  no switchport
  ip address 192.168.24.1/24
!
interface Ethernet6
  no switchport
  ip address 192.168.26.1/24
!
interface Ethernet7
!
interface Ethernet8
!
interface Ethernet9
!
interface Ethernet10
!
interface Ethernet11
!
interface Ethernet12
!
```

```

interface Management1
  vrf MGMT
  ip address 192.168.10.70/24
!
ip route vrf MGMT 0.0.0.0/0 192.168.10.1
!
no ip routing
no ip routing vrf MGMT
!
management telnet
  no shutdown
!
end
Spine-1#config replace flash:startup-config

```

```
> no ip routing vrf MGMT
```

```
! Preserving static routes. Use 'no ip routing delete-static-routes' to clear them. at line 49
```

```
Spine-1#show run
```

```

! Command: show running-config
! device: Spine-1 (vEOS, EOS-4.22.2F)
!
! boot system flash:/vEOS-lab.swi
!
transceiver qsfp default-mode 4x10G
!
hostname Spine-1
!
spanning-tree mode mstp
!
no aaa root
!
username admin privilege 15 role network-admin secret sha512
$6$e2J8z/HDC03RmLPu$i02zAYE/7wOfySAKI0PbgL7r6PA6IUfsh6Bf7puIlZi11eUP1bxQgr73JHItjH
vkGyJMj5X1vR0xvRp7E9iPf.
!
vrf instance MGMT
!
interface Ethernet1
!
interface Ethernet2
!
interface Ethernet3
!
interface Ethernet4
!
interface Ethernet5
!
interface Ethernet6
!
interface Ethernet7
!
interface Ethernet8
!
interface Ethernet9
!
interface Ethernet10

```

```
!  
interface Ethernet11  
!  
interface Ethernet12  
!  
interface Management1  
    vrf MGMT  
    ip address 192.168.10.70/24  
!  
ip route vrf MGMT 0.0.0.0/0 192.168.10.1  
!  
no ip routing  
no ip routing vrf MGMT  
!  
management telnet  
    no shutdown  
!  
end  
Spine-1#
```

Configuration Checkpoints

EOS allows to create several configuration checkpoints and to replace a running config **without a reboot**.

All services which are not affected by any changes will continue to operate!

Creating a checkpoint:

```
Spine-1#config checkpoint save test-change
Spine-1#show config checkpoints
Maximum number of checkpoints: 20
  Filename                Date                User
-----
ckp-20191008-0          2019-10-08 10:51:18  admin
  test-change           2019-10-08 11:42:43  admin
```

Restoring to a checkpoint:

```
Spine-1#config checkpoint restore test-change
```

```
> no ip routing vrf MGMT
```

```
! Preserving static routes. Use 'no ip routing delete-static-routes' to clear
them. at line 49
```

```
Spine-1#
```

Generating a Certificate and activating it on EOS

Sometimes you may have to import certificates from an external system into EOS.
In the following example we will do this for an **SSH Certificate** created on an Ubuntu server.
First you have to **create a key** on your Ubuntu Server

```
mkrausch@ubuntu18:~$ ssh-keygen -t rsa -b 4096
Generating public/private rsa key pair.
Enter file in which to save the key (/home/mkrausch/.ssh/id_rsa): testcert
Enter passphrase (empty for no passphrase):password
Enter same passphrase again:password
Your identification has been saved in testcert.
Your public key has been saved in testcert.pub.
The key fingerprint is:
SHA256:h70h70DcX1eVMEUoiiU3dfX0hpqZHRlu2/VQ4/Hbrog mkrausch@ubuntu18
The key's randomart image is:
+---[RSA 4096]-----+
|          .. +*+o |
|          . + .....o |
|          = o .   o |
|          . o o     = |
|          o S . . oo= |
|          . . * . *.o+ |
|          o . o * o.. |
|          o  * + oo |
|          .ooE.+ .++ |
+-----[SHA256]-----+
mkrausch@ubuntu18:~$ dir
testcert          cvprac_create_container.py  Desktop    most_used_port.py
pb.config.lines.simple.yaml  Templates
testcert.pub     cvprac_inventory.py        Documents  Music
Pictures          test1.py
authentication.yaml  cvprac_log                 Downloads  pb.collect.facts.yaml
Public            Videos
mkrausch@ubuntu18:~$
```

The next step is to **copy the key** to your EOS systems.
The easiest way is to do this directly from the EOS systems via CLI:

```
switch-1#copy scp://mkrausch:password@192.168.10.180/home/mkrausch/ansible.pub
flash:/testcert.pub
mkrausch@192.168.10.180's password:password
testcert.pub          100% 743   186.9KB/s   00:00
Copy completed successfully.
switch-1#dir
Directory of flash:/

-rw-          6930          Nov 29 09:26  AsuFastPktTransmit.log
drwx          4096          Oct 4 10:56   Fossil
-rw-          4260          Nov 29 09:26  SsuRestore.log
-rw-          4260          Nov 29 09:26  SsuRestoreLegacy.log
-rw-           743          Nov 29 10:07  testcert.pub
-rw-           24          Sep 26 22:56  boot-config
drwx          4096          Nov 21 09:26  debug
drwx          4096          Oct 4 10:56   fastpkttx.backup
```

```
drwx      16384      Sep 26 22:56  lost+found
drwx      4096       Nov 29 10:06  persist
drwx      4096       Oct 4 11:01   schedule
-rw-      2193       Nov 21 09:59  startup-config
-rw- 418919380     Sep 26 22:56  vEOS-lab.swi
-rw-      13        Oct 4 12:37   zerotouch-config
```

4093313024 bytes total (3225059328 bytes free)

In the last step you have to configure the switch so that the **Certificate will be assigned to a user**:

```
switch-1(config)#username testuser1 privilege 15 secret password
switch-1(config)#username testuser1 sshkey file flash:testcert.pub
```

Accessing the Linux system

EOS allows to access the Linux Shell directly from the CLI.

```
Spine-1#bash
```

```
Arista Networks EOS shell
```

```
[admin@Spine-1 ~]$ cd ..  
[admin@Spine-1 home]$ dir  
admin cvxrepl sessionuser  
[admin@Spine-1 home]$ cd ..  
[admin@Spine-1 /]$ dir  
bin dev export lib mnt opt proc rootfs-i386 sbin sys usr  
boot etc home media monitor persist root run srv tmp var  
[admin@Spine-1 /]$ exit  
logout  
Spine-1#
```

From the Linux environment you can even execute CLI commands:

```
Spine-1#bash
```

```
Arista Networks EOS shell
```

```
[admin@Spine-1 ~]$ Cli  
Spine-1>show interfaces management 1  
Management1 is up, line protocol is up (connected)  
Hardware is Ethernet, address is 0c7d.0623.cd00 (bia 0c7d.0623.cd00)  
Internet address is 192.168.10.70/24  
Broadcast address is 255.255.255.255  
IP MTU 1500 bytes , BW 1000000 kbit  
Full-duplex, 1Gb/s, auto negotiation: on, uni-link: n/a  
Up 16 minutes, 29 seconds  
Loopback Mode : None  
6 link status changes since last clear  
Last clearing of "show interface" counters never  
5 minutes input rate 0 bps (0.0% with framing overhead), 0 packets/sec  
5 minutes output rate 26 bps (0.0% with framing overhead), 0 packets/sec  
0 packets input, 0 bytes  
Received 0 broadcasts, 0 multicast  
0 runts, 0 giants  
0 input errors, 0 CRC, 0 alignment, 0 symbol, 0 input discards  
0 PAUSE input  
78 packets output, 3757 bytes  
Sent 4 broadcasts, 35 multicast  
0 output errors, 0 collisions  
0 late collision, 0 deferred, 0 output discards  
0 PAUSE output  
Spine-1>exit  
[admin@Spine-1 ~]$
```

The option “-c” executes a CLI command and then returns to the Linux Shell.

```
[admin@Spine-1 ~]$ Cli -c "show version" | grep Software
Software image version: 4.22.2F
[admin@Spine-1 ~]$
```

Configuration a range of parameters

EOS allows via **curly braces** to configure a range of parameters automatically. Within the curly braces there are three parameters which can be set:

{start,end,step}

In the following example 4 interfaces are automatically configured with 4 different IP subnets:

```
Spine-1(config)#interface ethernet 3-6
Spine-1(config-if-Et3-6)#no switchport
Spine-1(config-if-Et3-6)#ip address 192.168.{20,26,2}.1/24
Spine-1(config-if-Et3-6)#show run interfaces ethernet 3-6
interface Ethernet3
  no switchport
  ip address 192.168.20.1/24
interface Ethernet4
  no switchport
  ip address 192.168.22.1/24
interface Ethernet5
  no switchport
  ip address 192.168.24.1/24
interface Ethernet6
  no switchport
  ip address 192.168.26.1/24
Spine-1(config-if-Et3-6)#
```

Configuring sFlow

Configuring sFlow on EOS devices includes a **general** and an **Interface specific** configuration:

```
sflow sample 20
sflow polling-interval 10
sflow vrf MGMT destination 192.168.10.60
sflow vrf MGMT source 192.168.10.73
sflow run
!
vrf instance MGMT
!
interface Ethernet12
    switchport access vlan 10
    sflow enable
!
interface Management1
    vrf MGMT
    ip address 192.168.10.73/24
    no lldp receive
!
ip route vrf MGMT 0.0.0.0/0 192.168.10.1
!
no ip routing
no ip routing vrf MGMT
!
switch-3#show sflow
sFlow Configuration
-----
Destination(s):
  192.168.10.60:6343 (VRF: MGMT)
Source(s):
  192.168.10.73 (VRF: MGMT)
  :: (default) (VRF: MGMT)
Hardware Sample Rate for SW sFlow: 20
Polling Interval (sec): 10.0
Rewrite DSCP value: No

Status
-----
Running: Yes
Polling On: Yes (default)
Sampling On: Yes (default)
Send Datagrams:
  Yes (VRF: MGMT)
BGP Export:
  No (VRF: MGMT)
Hardware Sample Rate for SW sFlow: 16

Statistics
-----
Total Packets: 2520
Number of Samples: 140
Sample Pool: 2240
Hardware Trigger: 140
Number of Datagrams: 1259
```

Software Upgrade

You must download the firmware from Arista and copy it to your Server or USB stick. Please make sure that the software image ends on **“.swi”**.

Using USB Stick

You have to format the USB Stick with the FAT filesystem before you can use it!

```
Spine-1#copy usb1:/EOS-4.14.6M.swi flash:/EOS-4.14.6M.swi
Spine-1#conf t
Spine-1(config)#boot system flash:EOS-4.14.6M.swi
Spine-1(config)#reload
```

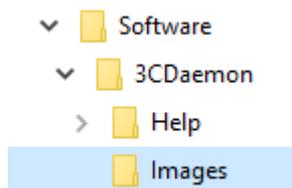
Using FTP

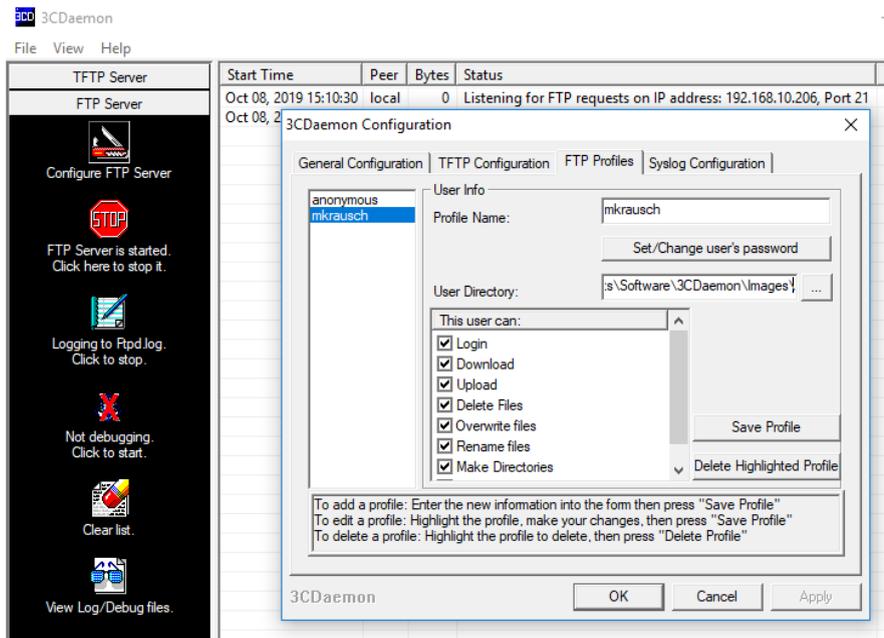
Ping Check via OOB Port:

```
Spine-1#ping vrf MGMT 192.168.10.206
PING 192.168.10.206 (192.168.10.206) 72(100) bytes of data.
 80 bytes from 192.168.10.206: icmp_seq=1 ttl=128 time=2.93 ms
 80 bytes from 192.168.10.206: icmp_seq=2 ttl=128 time=0.631 ms
 80 bytes from 192.168.10.206: icmp_seq=3 ttl=128 time=0.638 ms
 80 bytes from 192.168.10.206: icmp_seq=4 ttl=128 time=0.587 ms
 80 bytes from 192.168.10.206: icmp_seq=5 ttl=128 time=0.625 ms

--- 192.168.10.206 ping statistics ---
 5 packets transmitted, 5 received, 0% packet loss, time 16ms
 rtt min/avg/max/mdev = 0.587/1.083/2.936/0.926 ms, ipg/ewma 4.154/1.977 ms
Spine-1#
```

FTP Server Settings (e.g. 3C Daemon for Windows):





Software Update via OOB Port:

```
Spine-1#routing-context vrf MGMT
Spine-1(vrf:MGMT)#copy ftp://mkrausch:getaccess@192.168.10.206/EOS-4.21.1F.swi
flash:/EOS-4.21.1F.swi
Copy completed successfully.
Spine-1(vrf:MGMT)#
Spine-1(vrf:MGMT)#dir
Directory of flash:/
```

```
-rw-          2310          Oct 8 09:54  AsuFastPktTransmit.log
-rw-   700978970          Oct 8 13:22  EOS-4.21.1F.swi
drwx          4096          Oct 4 10:56  Fossil
-rw-          1420          Oct 8 09:54  SsuRestore.log
-rw-          1420          Oct 8 09:54  SsuRestoreLegacy.log
-rw-           24          Sep 26 22:56  boot-config
drwx          4096          Oct 4 14:54  debug
drwx          4096          Oct 4 10:56  fastpkttx.backup
drwx         16384          Sep 26 22:56  lost+found
drwx          4096          Oct 8 13:30  persist
drwx          4096          Oct 4 11:01  schedule
-rw-           900          Oct 4 13:49  startup-config
-rw-   618919380          Sep 26 22:56  EOS-4.20.1.swi
-rw-           13          Oct 4 12:37  zerotouch-config
```

```
4093313024 bytes total (3228876800 bytes free)
Spine-1(vrf:MGMT)#exit
```

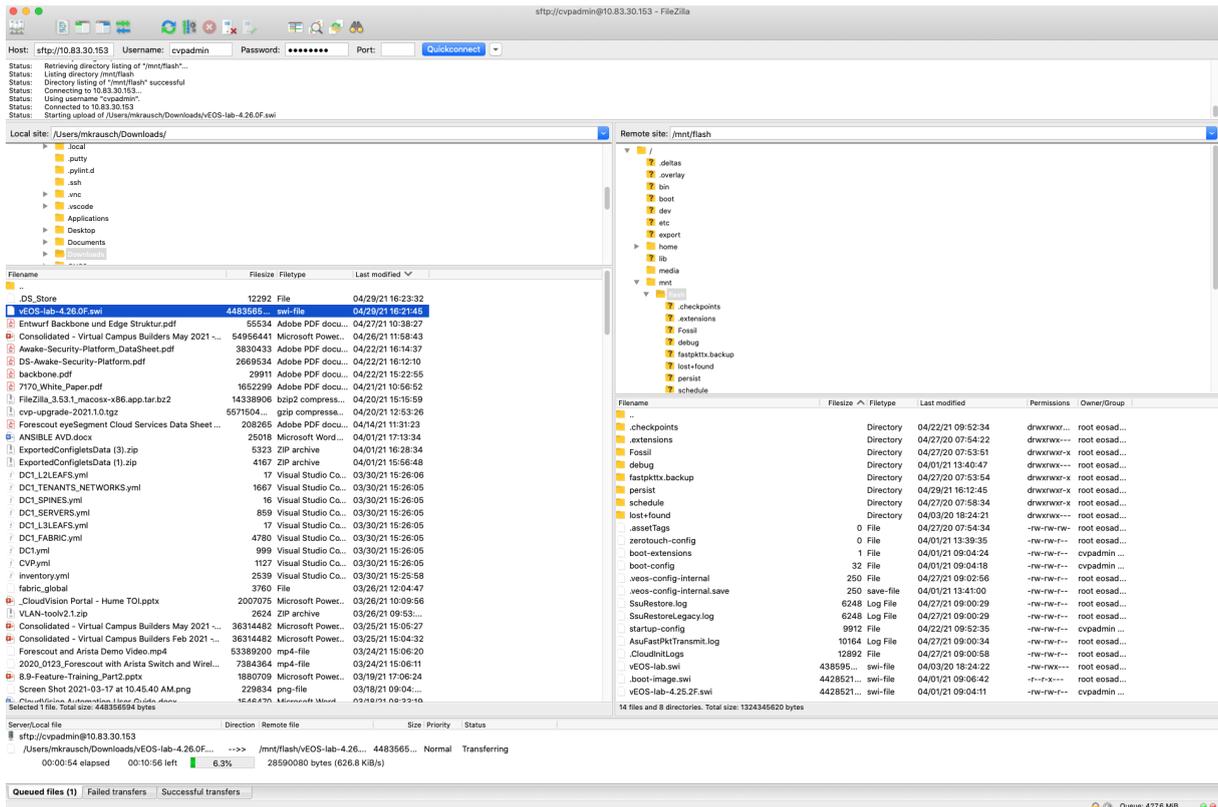
```
Spine-1 login: admin
Password:
Last login: Tue Oct 8 13:29:33 on ttyS0
Spine-1>en
Spine-1#conf t
Spine-1(config)#boot system flash:EOS-4.21.1F.swi
Spine-1(config)#reload
```

Alternatively you may use e.g. SFTP to transmit a file from your PC to an Arista system with a suitable client program like e.g. Filezilla.

Before connecting to the Arista system, first configure AAA so that you can connect to the system:

```
DC1-LEAF1A(config)#aaa authorization exec default local
```

Now you may connect to the switch with a suitable client program and copy the image to the folder /mnt/flash:



Accelerated Software Upgrade (ASU)

The Accelerated Software Upgrade process significantly decreases downtime and packet loss during a software upgrade. After the control plane has fully loaded, the data plane is restarted, causing approximately **30 seconds of downtime**.

After downloading the new Software Image to the systems perform the following command instead of “reload” to start the ASU process:

```
Arista-1#reload fast-boot
System configuration has been modified. Save? [yes/no/cancel/diff]:y
Proceed with reload? [confirm]y
Proceeding with reload
```

Smart System Upgrade (SSU)

The Smart System Upgrade (SSU) process includes the core functionality of Accelerated Software Upgrade (ASU), plus additional optimizations that permit a hitless restart of several features. SSU leverages protocols capable of **graceful restart** to minimize traffic loss during upgrade.

SSU is not compatible with VRRP! If VRRP is configured on the switch, another upgrade method must be used.

For hitless restart of BGP and MP-BGP, **BGP graceful restart** must first be enabled using the **graceful-restart** command:

```
Arista-1(config)#router bgp 65000
Arista-1(config-router-bgp)#graceful-restart
```

After downloading the new Software Image to the systems perform the following command instead of “reload” to start the SSU process:

```
Arista-1#reload hitless
System configuration has been modified. Save? [yes/no/cancel/diff]:y
Copy completed successfully.
Proceed with reload? [confirm]y
```

Maintenance Mode

Maintenance mode uses BGP to divert traffic away from the switch on which the maintenance tasks need to be performed, minimizing traffic impact while e.g. upgrading an EOS image or replacing a hardware.

There are various **built-in units** such as **System** and **Linecard<n>**.

Fixed systems contain only one built-in unit called **System**, which comprises the interface group containing all Ethernet interfaces and sub-interfaces.

There are also several built-in groups such as **AllEthernetInterface**, **Linecard1**, **Linecard2**, etc., **AllBgpNeighborVrf-<vrf_name>**.

AllEthernetInterface is the built-in interface group which contains all physical Ethernet interfaces and sub-interfaces on the switch, and is a part of **System unit**.

User configured Units and Groups:

You can also configure customized units containing user-defined groups and policies:

```
switch(config)#maintenance
switch(config-maintenance)# unit UNIT1
switch(config-unit-UNIT1)# group bgp BG1
switch(config-unit-UNIT1)# group interface IG1
switch(config-unit-UNIT1)# profile unit UP1
switch(config-unit-UNIT1)# exit
switch(config)#group interface IG1
switch(config-group-if-IG1)#interface Ethernet1
switch(config-group-if-IG1)#interface Port-Channel1,20
switch(config-group-if-IG1)#interface Vlan1-20
switch(config-group-if-IG1)#exit
switch(config)#group bgp BG1
switch(config-group-bgp-BG1)#neighbor 10.0.0.1
switch(config-group-bgp-BG1)#neighbor BGP_PG1
switch(config-group-bgp-BG1)#vrf vrf1
switch(config-group-bgp-BG1)#exit
```

Configuring Profiles:

Profiles are configurable maintenance mode elements that define **policies** for related software or hardware components to carry out maintenance mode operations.

There are default profiles like **Interface Profile**, **BGP Profile** and **Unit Profile** which are used in the absence of an explicit interface/BGP profile associated with the group.

You can also configure **User-defined Profiles**:

```
switch(config)#maintenance
switch(config-maintenance)#profile interface IP1
switch(config-profile-intf-IP1)#rate-monitoring load-interval 10
switch(config-profile-intf-IP1)#rate-monitoring threshold 100
switch(config-profile-intf-IP1)#shutdown max-delay 100
switch(config-profile-intf-IP1)#exit
switch(config-maintenance)#exit
switch(config)#group interface IG1
switch(config-group-if-IG1)#maintenance profile interface IP1
switch(config-group-if-IG1)#exit
```

On-boot Unit-level Maintenance Mode Configuration

To configure on-boot maintenance, you can use one of two methods:

- Use quiesce config
- Use on-boot profile

quiesce config:

```
switch(config)#maintenance
switch(config-maintenance)# unit System
switch(config-unit-System)# quiesce
switch(config-unit-System)# exit
switch(config-maintenance)# exit
switch(config)# copy running-config startup-config
Copy completed successfully
switch(config)# reload
Proceed with reload? [Confirm] yes
Connection to switch closed.
```

After the device comes up, you must execute the **no quiesce** command for the Unit to come out of maintenance mode.

```
switch(config)#maintenance
switch(config-maintenance)#unit System
switch(config-unit-foo)#no quiesce
```

on-boot profile:

In the next example the Unit will come up in maintenance mode when the device boots up and will exit maintenance mode once the specified duration of 300 seconds in the default profile is completed.

```
switch(config)#maintenance
switch(config-maintenance)#profile unit UP1
switch(config-profile-unit-UP1)#on-boot duration 300
switch(config-profile-unit-UP1)#exit
switch(config-maintenance)#profile unit UP1 default
switch(config-maintenance)# exit
switch(config)# copy running-config startup-config
Copy completed successfully
switch(config)# reload
Proceed with reload? [Confirm] yes
Connection to switch closed.
```

Interface-level Maintenance Mode Configuration:

```
switch(config)#maintenance
switch(config-maintenance)#profile interface IP1
switch(config-maint-if-Et5)#rate-monitoring load-interval 100
switch(config-maint-if-Et5)#rate-monitoring threshold 500
switch(config-maint-if-Et5)#shutdown max-delay 100
switch(config-maint-if-Et5)# exit
switch(config-maintenance)# profile interface IP1 default
switch(config-maintenance)#interface Ethernet 1
switch(config-maint-if-Et1)#quiesce
```

Remove the interface from maintenance mode once the service has been performed:

```
switch(config-maintenance)#interface Ethernet 1
switch(config-maint-if-Et1)#no quiesce
```

Installing Extensions

Arista EOS extensions can be packaged as normal RPMs or as a bundle of RPMs called a SWIX and can be installed on an EOS device to extend the functionality of the system in a very flexible way.

Upgrading TerminAttr

When using CloudVision it may be necessary from time to time to update the so called **TerminAttr** agent to a newer release and this also can be done by installing TerminAttr as an extension without the need to reboot the system.

First you will have to upload the TerminAttr swix file to the EOS Flash which can be done in the same way covered in the Software Upgrade Chapter using e.g. FileZilla.

Please check whether it has been copied successfully to the EOS system:

```
ROUTER1#dir
```

```
Directory of flash:/
```

```

-rw-          6006          Feb 10 07:31  AsuFastPktTransmit.log
drwx          4096          Jul 16  2021  Fossil
-rw-          3692          Feb 10 07:31  SsuRestore.log
-rw-          3692          Feb 10 07:31  SsuRestoreLegacy.log
-rw-       7151882          Feb 10 08:09  TerminAttr-1.16.7-1.swix
-rw-           24          Apr 28  2021  boot-config
drwx          4096          Jan 6  11:56  debug
drwx          4096          Jul 16  2021  fastpkttx.backup
drwx         16384          Apr 28  2021  lost+found
drwx          4096          Feb 10 08:08  persist
drwx          4096          Jul 16  2021  schedule
-rw-          1537          Jan 13 09:43  startup-config
-rw-    448307677          Apr 28  2021  vEOS-lab.swi
-rw-           0          Jul 16  2021  zerotouch-config
```

```
4093313024 bytes total (3148075008 bytes free)
```

Installing the new TerminAttr extension and make it persistent even after a reboot:

```
ROUTER1#show version detail | include Termin
```

```
TerminAttr          1.0.3          21984485.42601F
```

```
TerminAttr-core    v1.14.0          1
```

```
TerminAttr-lib      1.0.3          21984485.42601F
```

```
ROUTER1#conf t
```

```
ROUTER1(config)#daemon TerminAttr
```

```
ROUTER1(config-daemon-TerminAttr)#shutdown
```

```
ROUTER1(config-daemon-TerminAttr)#copy flash:TerminAttr-1.16.7-1.swix extension:
```

```
Copy completed successfully.
```

```
ROUTER1(config-daemon-TerminAttr)#extension TerminAttr-1.16.7-1.swix
```

```
Agents to be restarted:
```

```
Note: no agents to restart
```

```
ROUTER1(config-daemon-TerminAttr)#copy installed-extensions boot-extensions
```

```
Copy completed successfully.
```

```
ROUTER1(config-daemon-TerminAttr)#no shutdown
```

```
ROUTER1(config-daemon-TerminAttr)#show version detail | include Termin
```

```
TerminAttr          1.0.3          21984485.42601F
```

```
TerminAttr-core      v1.16.7      1
TerminAttr-lib       1.0.3       21984485.42601F
ROUTER1(config-daemon-TerminAttr)#
```

Port Configuration

Configuring the description and Port Speed:

```
interface Ethernet1
  description POD1-SPINE1
  speed forced 1000full
```

Configuring Breakout Ports

In order to break out e.g. a 100Gig Interface into 4 x 10Gig you just need to change the Port speed.

Configuring Breakout Ports does not require a reboot!

```
switch(config-if-Et54/1)#show run
```

```
...
!
interface Ethernet53/1
  shutdown
!
interface Ethernet54/1
!
...
```

```
switch(config)#show interfaces eth 54/1
Ethernet54/1 is down, line protocol is down (notconnect)
  Hardware is Ethernet, address is fcbd.670f.2432 (bia fcbd.670f.2432)
  Ethernet MTU 9214 bytes , BW 100000000 kbit
  Full-duplex, 100Gb/s, auto negotiation: off, uni-link: n/a
  Down 39 minutes, 21 seconds
  Loopback Mode : None
  3 link status changes since last clear
  Last clearing of "show interface" counters 0:43:59 ago
  5 seconds input rate 0 bps (0.0% with framing overhead), 0 packets/sec
  5 seconds output rate 0 bps (0.0% with framing overhead), 0 packets/sec
    0 packets input, 0 bytes
    Received 0 broadcasts, 0 multicast
    0 runts, 0 giants
    0 input errors, 0 CRC, 0 alignment, 0 symbol, 0 input discards
    0 PAUSE input
    0 packets output, 0 bytes
    Sent 0 broadcasts, 0 multicast
    0 output errors, 0 collisions
    0 late collision, 0 deferred, 0 output discards
    0 PAUSE output
```

```
switch(config)#interface ethernet 54/1
switch(config-if-Et54/1)#speed forced 10000full
switch(config-if-Et54/1)#interface ethernet 54/2
switch(config-if-Et54/2)#speed forced 10000full
switch(config-if-Et54/2)#show run
```

```
...
!
interface Ethernet53/1
```

```

    shutdown
!
interface Ethernet54/1
    speed forced 10000full
!
interface Ethernet54/2
    speed forced 10000full
!
interface Ethernet54/3
!
interface Ethernet54/4
!
...

```

```

switch(config-if-Et54/2)#show interfaces eth 54/1
Ethernet54/1 is up, line protocol is up (connected)
  Hardware is Ethernet, address is fcbd.670f.2432 (bia fcbd.670f.2432)
  Ethernet MTU 9214 bytes , BW 10000000 kbit
  Full-duplex, 10Gb/s, auto negotiation: off, uni-link: n/a
  Up 34 seconds
  Loopback Mode : None
  4 link status changes since last clear
  Last clearing of "show interface" counters 0:58:58 ago
  5 seconds input rate 732 bps (0.0% with framing overhead), 1 packets/sec
  5 seconds output rate 409 bps (0.0% with framing overhead), 0 packets/sec
    7 packets input, 1009 bytes
    Received 0 broadcasts, 7 multicast
    0 runts, 0 giants
    0 input errors, 0 CRC, 0 alignment, 0 symbol, 0 input discards
    0 PAUSE input
    18 packets output, 2438 bytes
    Sent 0 broadcasts, 18 multicast
    0 output errors, 0 collisions
    0 late collision, 0 deferred, 0 output discards
    0 PAUSE output

```

Configuring an IP address on a Port

```

interface Ethernet1
  no switchport
  ip address 192.168.20.1/24

```

Configuring a Port Profile

Instead of configuring each port individually you may configure a global port profile and assign it to ports.

```
!  
interface profile radiustest  
  command description OFFICE/GUEST  
  command switchport trunk native vlan 200  
  command spanning-tree portfast  
  command dot1x pae authenticator  
  command dot1x authentication failure action traffic allow vlan 200  
  command dot1x reauthentication  
  command dot1x port-control auto  
  command dot1x host-mode multi-host authenticated  
  command dot1x mac based authentication  
  command dot1x timeout reauth-period 28800  
!  
interface Ethernet1  
  profile radiustest  
!  
interface Ethernet2  
  profile radiustest  
!
```

Configuring a VLAN

Adding a VLAN:

```
vlan 10
  name VLAN-10
```

Assigning an IP address to a VLAN

If routing between VLANs should be possible IP routing must be globally enabled (disabled by default)!

```
interface Vlan10
  ip address 192.168.10.5/24
!
ip routing
```

Assigning a tagged Port to a VLAN

```
interface Ethernet1
  switchport trunk native vlan tag
  switchport trunk allowed vlan 30, 50
  switchport mode trunk
```

Assigning an untagged Port to a VLAN:

```
interface Ethernet2
  switchport access vlan 50
```

Configuring a Voice VLAN

In some cases a client may be connected to an IP phone which itself is connected to a port. If both devices are sending untagged packets to the Arista system it can assign the IP phone to a preconfigured Voice VLAN by detecting the phone via LLDP (enabled by default on the Arista systems)

```
!  
vlan 591  
    name Client-Network-591  
!  
vlan 691  
    name Voice-Network-691  
!  
!  
interface Ethernet1  
    switchport trunk native vlan 591  
    switchport phone vlan 691  
    switchport phone trunk untagged  
    switchport mode trunk phone  
    spanning-tree portfast  
!
```

Configuring a Port Channel

To configure an **LACP based Port Channel** please use mode **“active”**.
For a **static Port Channel** please use mode **“on”**

```
vlan 10
  name VLAN-10
!
interface Port-Channel10
  switchport trunk allowed vlan 10
  switchport mode trunk
!
interface Ethernet1
  channel-group 10 mode active
!
interface Ethernet2
  channel-group 10 mode active
```

IEEE 802.1Q tagged LACP BPDUs:

Natively, EOS discards tagged LACP PDUs.

Arista implemented a feature on EOS that allows it to accept tagged LACP PDUs for interoperability to other 3rd party LACP implementations.

```
!
interface Port-Channel10
  switchport trunk allowed vlan 10
  switchport mode trunk
  l2-protocol encapsulation dot1q vlan 10
!
```

Configuring MLAG

Example of how to configure an MLAG

Client VLAN: VLAN 10

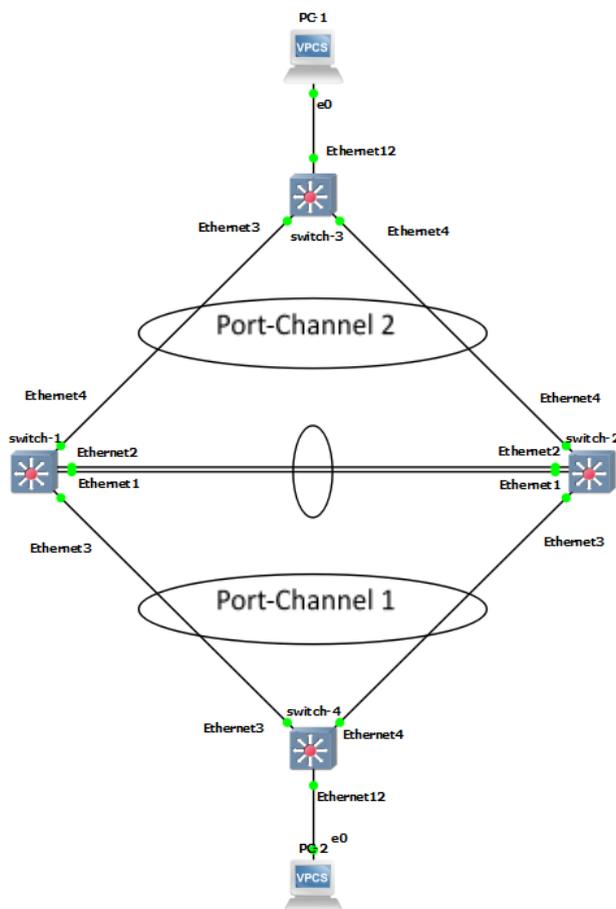
MLAG Interlink VLAN: 4094

MLAG Interlink Port-Channel: 1000

MLAG Client Port-Channels: 1, 2

MLAG Domain-ID: Arista-MLAG-1

The “no autostate” config keeps the L3 peering IP interface up even if no ports are active within the assigned VLAN.



Be aware that when using e.g. GNS3 with an vEOS-lab version the simulated Ethernet Ports may not go down when reloading a simulated system via CLI!

For simulated failover tests you better may power off the system completely via GNS3 “Stop” Button

Additional information can be found at:

<https://www.arista.com/en/products/multi-chassis-link-aggregation-mlag>

Switch-1:

```
hostname switch-1
!
vlan 10
    name client-VLAN-10
!
vlan 4094
    name MLAG-Interlink-VLAN
    trunk group MLAG-Peer
!
interface Port-Channel1
    switchport trunk allowed vlan 10
    switchport mode trunk
    mlag 1
!
interface Port-Channel2
    switchport trunk allowed vlan 10
    switchport mode trunk
    mlag 2
!
interface Port-Channel1000
    switchport mode trunk
    switchport trunk group MLAG-Peer
!
interface Ethernet1
    description MLAG-Interface
    channel-group 1000 mode active
!
interface Ethernet2
    description MLAG-Interface
    channel-group 1000 mode active
!
interface Ethernet3
    channel-group 1 mode active
!
interface Ethernet4
    channel-group 2 mode active
!
interface Vlan4094
    description MLAG-Interlink-IP
    no autostate
    ip address 192.168.50.1/24
!
mlag configuration
    domain-id Arista-MLAG-1
    local-interface Vlan4094
    peer-address 192.168.50.2
    peer-link Port-Channel1000
```

Switch 2:

```
hostname switch-2
!
vlan 10
    name client-VLAN-10
!
vlan 4094
    name MLAG-Interlink-VLAN
    trunk group MLAG-Peer
!
interface Port-Channel1
    switchport trunk allowed vlan 10
    switchport mode trunk
    mlag 1
!
interface Port-Channel2
    switchport trunk allowed vlan 10
    switchport mode trunk
    mlag 2
!
interface Port-Channel1000
    switchport mode trunk
    switchport trunk group MLAG-Peer
!
interface Ethernet1
    description MLAG-Interface
    channel-group 1000 mode active
!
interface Ethernet2
    description MLAG-Interface
    channel-group 1000 mode active
!
interface Ethernet3
    channel-group 1 mode active
!
interface Ethernet4
    channel-group 2 mode active
!
interface Vlan4094
    description MLAG-Interlink-IP
    no autostate
    ip address 192.168.50.2/24
!
mlag configuration
    domain-id Arista-MLAG-1
    local-interface Vlan4094
    peer-address 192.168.50.1
    peer-link Port-Channel1000
```

Switch-3:

```
hostname switch-3
!
vlan 10
    name client-VLAN-10
!
interface Port-Channel2
    switchport trunk allowed vlan 10
    switchport mode trunk
!
interface Ethernet3
    channel-group 2 mode active
!
interface Ethernet4
    channel-group 2 mode active
!
interface Ethernet12
    switchport access vlan 10
```

Switch-4:

```
hostname switch-4
!
vlan 10
    name client-VLAN-10
!
interface Port-Channel1
    switchport trunk allowed vlan 10
    switchport mode trunk
!
interface Ethernet3
    channel-group 1 mode active
!
interface Ethernet4
    channel-group 1 mode active
!
interface Ethernet12
    switchport access vlan 10
```

MLAG verification

```
switch-1#show mlag config-sanity
```

```
No global configuration inconsistencies found.
```

```
No per interface configuration inconsistencies found.
```

```
switch-1#show mlag
```

```
MLAG Configuration:
```

```
domain-id       : Arista-MLAG-1
local-interface : Vlan4094
peer-address    : 192.168.50.2
peer-link       : Port-Channel1000
peer-config     : consistent
```

```
MLAG Status:
```

```
state           : Active
negotiation status : Connected
peer-link status : Up
local-int status : Up
system-id       : 0e:34:13:15:50:af
dual-primary detection : Disabled
```

```
MLAG Ports:
```

```
Disabled       : 0
Configured     : 0
Inactive       : 0
Active-partial : 0
Active-full    : 2
```

```
switch-1#show mlag interfaces
```

mlag	desc	state	local	remote	local/remote status
1		active-full	Po1	Po1	up/up
2		active-full	Po2	Po2	up/up

Important note:

After an MLAG Peer has been rebooted the Ethernet interfaces will be by default in an error disable state for 300 secs:

```
switch-1#show interfaces status
```

Port	Name	Status	Vlan	Duplex	Speed	Type
Flags Encapsulation						
Et1	MLAG-Interface	connected	in Po1000	full	unconf	EbraTestPhyPort
Et2	MLAG-Interface	connected	in Po1000	full	unconf	EbraTestPhyPort
Et3		errdisabled	in Po1	full	unconf	EbraTestPhyPort
Et4		errdisabled	in Po2	full	unconf	EbraTestPhyPort
Et5		errdisabled	1	full	unconf	EbraTestPhyPort
Et6		errdisabled	1	full	unconf	EbraTestPhyPort
Et7		errdisabled	1	full	unconf	EbraTestPhyPort
Et8		errdisabled	1	full	unconf	EbraTestPhyPort
Et9		errdisabled	1	full	unconf	EbraTestPhyPort
Et10		errdisabled	1	full	unconf	EbraTestPhyPort
Et11		errdisabled	1	full	unconf	EbraTestPhyPort
Et12		errdisabled	1	full	unconf	EbraTestPhyPort

Ma1	connected	routed	a-full	a-1G	10/100/1000
Po1	notconnect	trunk	full	unconf	N/A
Po2	notconnect	trunk	full	unconf	N/A
Po1000	connected	trunk	full	unconf	N/A

You can see how much time is left until the interfaces will come back online:

```
switch-1#show mlag det | grep state
state          :          Active/Reload (0:00:20 left)
Last state change time      :          0:03:58 ago
P2p mount state changes
```

The default error disable time can be changed but should be **identical on both MLAG peers**:

```
switch-1(config)#mlag configuration
switch-1(config-mlag)#reload-delay 120
switch-1(config-mlag)#reload-delay non-mlag 60
```

Configuring MSTP

Default setup

Per default MSTP is enabled on all Arista systems with a default instance **MST0**.

Important:

Arista systems will send MST0 BPDUs only on the default VLAN (VLAN 1) and not on all VLANs!

Switch-2#show spanning-tree

MST0

```
Spanning tree enabled protocol mstp
Root ID    Priority    32768
           Address    0ced.fa1c.7a63
           Cost     0 (Ext) 2000 (Int)
           Port     1 (Ethernet1)
           Hello Time 2.000 sec  Max Age 20 sec  Forward Delay 15 sec

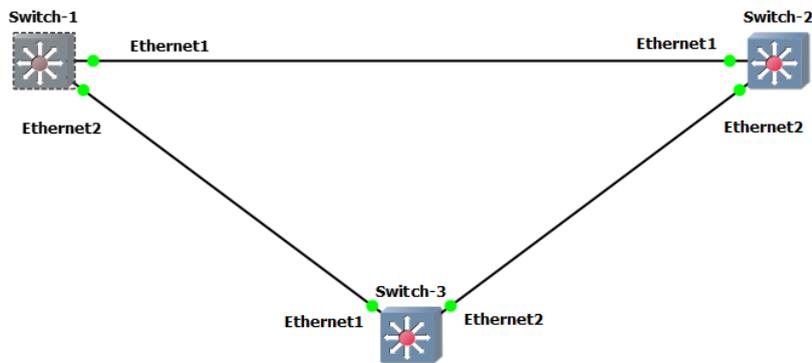
Bridge ID  Priority    32768 (priority 32768 sys-id-ext 0)
           Address    0ced.fa38.21fd
           Hello Time 2.000 sec  Max Age 20 sec  Forward Delay 15 sec
```

Interface	Role	State	Cost	Prio.Nbr	Type
Et1	root	forwarding	2000	128.1	P2p
Et2	alternate	discarding	2000	128.2	P2p
Et3	designated	forwarding	2000	128.3	P2p Edge
Et4	designated	forwarding	2000	128.4	P2p Edge
Et5	designated	forwarding	2000	128.5	P2p Edge
Et6	designated	forwarding	2000	128.6	P2p Edge
Et7	designated	forwarding	2000	128.7	P2p Edge
Et8	designated	forwarding	2000	128.8	P2p Edge
Et9	designated	forwarding	2000	128.9	P2p Edge
Et10	designated	forwarding	2000	128.10	P2p Edge
Et11	designated	forwarding	2000	128.11	P2p Edge
Et12	designated	forwarding	2000	128.12	P2p Edge

Example of how to configure MSTP

MSTP Instance 1: VLAN 10

MSTP Instance 2: VLAN 20, VLAN 30



Switch-1 (Root for MST1):

```
hostname Switch-1
!
spanning-tree mode mstp
spanning-tree mst 1 priority 8192
!
spanning-tree mst configuration
    name Demo-1
        instance 1 vlan 10
        instance 2 vlan 20,30
!
vlan 10
    name VLAN-10
!
vlan 20
    name VLAN-20
!
vlan 30
    name VLAN-30
!
interface Ethernet1
    switchport trunk allowed vlan
10,20,30
    switchport mode trunk
!
interface Ethernet2
    switchport trunk allowed vlan
10,20,30
    switchport mode trunk
!
```

Switch 2 (Root for MST2):

```
hostname Switch-2
!
spanning-tree mode mstp
spanning-tree mst 2 priority 8192
!
spanning-tree mst configuration
    name Demo-1
        instance 1 vlan 10
        instance 2 vlan 20,30
!
vlan 10
    name VLAN-10
!
vlan 20
    name VLAN-20
!
vlan 30
    name VLAN-30
!
interface Ethernet1
    switchport trunk allowed vlan
10,20,30
    switchport mode trunk
!
interface Ethernet2
    switchport trunk allowed vlan
10,20,30
    switchport mode trunk
!
```

Switch-3:

```
hostname Switch-3
!
spanning-tree mode mstp
!
spanning-tree mst configuration
  name Demo-1
  instance 1 vlan 10
  instance 2 vlan 20,30
!
vlan 10
  name VLAN-10
!
vlan 20
  name VLAN-20
!
vlan 30
  name VLAN-30
!
interface Ethernet1
  switchport trunk allowed vlan
10,20,30
  switchport mode trunk
!
interface Ethernet2
  switchport trunk allowed vlan
10,20,30
  switchport mode trunk
!
```

Switch-2#show spanning-tree

MST0

Spanning tree enabled protocol mstp

Root ID Priority 32768
 Address 0ced.fa1c.7a63
 Cost 0 (Ext) 2000 (Int)
 Port 1 (Ethernet1)
 Hello Time 2.000 sec Max Age 20 sec Forward Delay 15 sec

Bridge ID Priority 32768 (priority 32768 sys-id-ext 0)
 Address 0ced.fa38.21fd
 Hello Time 2.000 sec Max Age 20 sec Forward Delay 15 sec

Interface	Role	State	Cost	Prio.Nbr	Type
Et1	root	forwarding	2000	128.1	P2p
Et2	alternate	discarding	2000	128.2	P2p
Et3	designated	forwarding	2000	128.3	P2p Edge
Et4	designated	forwarding	2000	128.4	P2p Edge
Et5	designated	forwarding	2000	128.5	P2p Edge
Et6	designated	forwarding	2000	128.6	P2p Edge
Et7	designated	forwarding	2000	128.7	P2p Edge
Et8	designated	forwarding	2000	128.8	P2p Edge
Et9	designated	forwarding	2000	128.9	P2p Edge
Et10	designated	forwarding	2000	128.10	P2p Edge
Et11	designated	forwarding	2000	128.11	P2p Edge
Et12	designated	forwarding	2000	128.12	P2p Edge

MST1

Spanning tree enabled protocol mstp

Root ID Priority 8193
 Address 0ced.fa1c.7a63
 Cost 2000
 Port 1 (Ethernet1)
 Hello Time 0.000 sec Max Age 0 sec Forward Delay 0 sec

Bridge ID Priority 32769 (priority 32768 sys-id-ext 1)
 Address 0ced.fa38.21fd
 Hello Time 2.000 sec Max Age 20 sec Forward Delay 15 sec

Interface	Role	State	Cost	Prio.Nbr	Type
Et1	root	forwarding	2000	128.1	P2p
Et2	alternate	discarding	2000	128.2	P2p

MST2

Spanning tree enabled protocol mstp

Root ID Priority 8194
 Address 0ced.fa38.21fd
 This bridge is the root

Bridge ID Priority 8194 (priority 8192 sys-id-ext 2)
 Address 0ced.fa38.21fd
 Hello Time 2.000 sec Max Age 20 sec Forward Delay 15 sec

Interface	Role	State	Cost	Prio.Nbr	Type
Et1	designated	forwarding	2000	128.1	P2p

Et2

designated forwarding 2000

128.2

P2p

Configuring Link State Tracking

Link State Tracking kann be used to shut down **downstream** interfaces if more than the minimum configured **upstream** interfaces within a link-state group are going down. You can also configure a recovery delay time to make sure that all routing tables are loaded completely before enabling the downstream ports after the upstream ports have been recovered.

```
!  
link tracking group spine_uplinks  
  links minimum 2  
  recovery delay 30  
!  
interface Ethernet1  
  description P2P_LINK_TO_DC1-SPINE1_Ethernet5  
  no switchport  
  ipv6 enable  
  link tracking group spine_uplinks upstream  
!  
interface Ethernet2  
  description P2P_LINK_TO_DC1-SPINE2_Ethernet5  
  no switchport  
  ipv6 enable  
  link tracking group spine_uplinks upstream  
!  
interface Ethernet3  
  description P2P_LINK_TO_DC1-SPINE3_Ethernet5  
  no switchport  
  ipv6 enable  
  link tracking group spine_uplinks upstream  
!  
!  
interface Ethernet8  
  description Host-3  
  channel-group 8 mode active  
  link tracking group spine_uplinks downstream  
!
```

Configuring VRRP

VRRP default priority is 100.

Priorities can automatically be reduced by tracking the line protocol of an interface-

IP routing must be globally enabled (disabled by default)!

Additional information can be found at:

<https://www.arista.com/en/um-eos/eos-vrrp-and-varp>

Router 1 (Master):

```
hostname Router-1
!
track wanuplinkdown interface
Ethernet1 line-protocol
!
vlan 10
  name VLAN-10
!
interface Ethernet1
  switchport trunk allowed vlan 10
  switchport mode trunk
!
interface Ethernet2
  switchport trunk allowed vlan 10
  switchport mode trunk
!
interface Vlan10
  ip address 192.168.10.1/24
  vrrp 10 priority 120
  vrrp 10 ip 192.168.10.5
  vrrp 10 track wanuplinkdown
  decrement 40
!
ip routing
```

Router 2 (Backup):

```
hostname Router-2
!
!
!
!
vlan 10
  name VLAN-10
!
interface Ethernet1
  switchport trunk allowed vlan 10
  switchport mode trunk
!
interface Ethernet2
  switchport trunk allowed vlan 10
  switchport mode trunk
!
interface Vlan10
  ip address 192.168.10.2/24
  vrrp 10 ip 192.168.10.5
!
!
!
!
ip routing
```

Configuring Virtual ARP (VARP)

VARP is using **L3 IP Anycast Addresses** which are using the same MAC address and the same IP address on all configured nodes.

The `ip virtual-router mac-address` command assigns a virtual MAC address to the switch. The switch maps all virtual router IP addresses to this MAC address.

The address is receive-only and the switch never sends packets with this address as the source! IP routing must be globally enabled (disabled by default)!

Additional information can be found at:

<https://www.arista.com/en/um-eos/eos-vrrp-and-varp>

Router 1:

```
hostname Router-1
!
vlan 10
  name VLAN-10
!
interface Ethernet1
  switchport trunk allowed vlan 10
  switchport mode trunk
!
interface Ethernet2
  switchport trunk allowed vlan 10
  switchport mode trunk
!
interface Vlan10
  ip address 192.168.10.1/24
  ip virtual-router address
  192.168.10.5
!
ip virtual-router mac-address
00:1c:73:00:00:99
!
ip routing
```

Router 2:

```
hostname Router-2
!
vlan 10
  name VLAN-10
!
interface Ethernet1
  switchport trunk allowed vlan 10
  switchport mode trunk
!
interface Ethernet2
  switchport trunk allowed vlan 10
  switchport mode trunk
!
interface Vlan10
  ip address 192.168.10.2/24
  ip virtual-router address
  192.168.10.5
!
ip virtual-router mac-address
00:1c:73:00:00:99
!
ip routing
```

Configuring OSPF

Within OSPF the system sets the **router ID** to the first available alternative in the following list:

1. The router-id command under the Routing Protocol
2. The router-id command under Router General
3. The loopback IP address, if a loopback interface is active on the switch.
4. The highest IP address on the router.

IP Routing has to be enabled!

```
vlan 20
  name VLAN-20
!
vlan 30
  name VLAN-30
!
interface Ethernet1
  switchport access vlan 30
!
interface Ethernet12
  switchport access vlan 20
!
interface Vlan20
  ip address 192.168.20.100/24
  ip ospf authentication
  ip ospf authentication-key secretpw
!
interface Vlan30
  ip address 192.168.30.100/24
  ip ospf authentication
  ip ospf authentication-key secretpw
!
ip routing
no ip routing vrf MGMT
!
router ospf 1
  router-id 10.1.1.1
  redistribute connected
  redistribute static
  network 192.168.20.0/24 area 0.0.0.0
  network 192.168.30.0/24 area 0.0.0.0
  max-lsa 12000
!
```

Checking OSPF Connectivity

Router-1#show ip ospf neighbor

Neighbor ID	Instance	VRF	Pri	State	Dead Time	Address
Interface						
10.2.2.2	1	default	1	FULL/BDR	00:00:35	
192.168.30.200	Vlan30					

Router-1#show ip ospf interface brief

Interface	Instance	VRF	Area	IP Address	Cost	State
Nbrs						
Vl20	1	default	0.0.0.0	192.168.20.100/24	10	DR
0						
Vl30	1	default	0.0.0.0	192.168.30.100/24	10	DR
1						

Router-1#show ip ospf interface vlan 30

Vlan30 is up

Interface Address 192.168.30.100/24, instance 1, VRF default, Area 0.0.0.0

Network Type Broadcast, Cost: 10

Transmit Delay is 1 sec, State DR, Priority 1

Designated Router is 10.1.1.1

Backup Designated Router is 10.2.2.2

Timer intervals configured, Hello 10, Dead 40, Retransmit 5

Neighbor Count is 1

Simple authentication

Traffic engineering is disabled

Router-1#show ip ospf 1 0.0.0.0 database

OSPF Router with ID(10.1.1.1) (Instance ID 1) (VRF default)

Router Link States (Area 0.0.0.0)

Link ID	ADV Router	Age	Seq#	Checksum	Link count
10.2.2.2	10.2.2.2	1259	0x80000005	0xd249	2
10.1.1.1	10.1.1.1	1257	0x80000003	0x2676	2

Network Link States (Area 0.0.0.0)

Link ID	ADV Router	Age	Seq#	Checksum
192.168.30.100	10.1.1.1	1257	0x80000001	0x65be

Type-5 AS External Link States

Link ID	ADV Router	Age	Seq#	Checksum	Tag
192.168.10.0	10.1.1.1	1810	0x80000001	0x7bb5	0
0.0.0.0	10.2.2.2	1274	0x80000001	0x861b	0
0.0.0.0	10.1.1.1	1810	0x80000001	0x9b09	0
192.168.10.0	10.2.2.2	1274	0x80000001	0x66c7	0

QoS Configuration

Additional information can be found at:

<https://www.arista.com/en/um-eos/eos-quality-of-service>

Configuring Policy Maps

Arista systems can rewrite the CoS and DSCP field for outbound tagged/ IP packets.

You can configure different Policy maps which may **rewrite COS & DSCP** values for **incoming traffic** on **untrusted ports** and also define a certain assignment to a **traffic-class (queue)** different from the default class:

```
!  
policy-map type qos VIDEO  
  class class-default  
    set cos 2  
    set dscp 34  
    set traffic-class 4  
!  
policy-map type qos AUDIO  
  class AUDIOclass  
    set cos 4  
    set dscp 46  
    set traffic-class 3  
  !  
  class class-default  
!
```

Assigning a traffic class to a tx-queue

The default assignment of traffic classes to tx-queues can be changed

Arista (config) #qos map traffic-class 2 to tx-queue 1

Configuring a QoS profile

You may assign a bandwidth per queue when e.g. using weighted fair queueing for outgoing traffic on uplink ports:

```
qos profile Uplink  
  tx-queue 0  
    !! FT  
    no priority  
    bandwidth percent 5  
  !  
  tx-queue 1  
    !! DEFAULT  
    no priority  
    bandwidth percent 60  
  !  
  tx-queue 2  
    no priority
```

```

!
tx-queue 3
  !! QOS-OTHER
  no priority
  bandwidth percent 5
!
tx-queue 4
  !! VIDEO
  no priority
  bandwidth percent 30
!
tx-queue 5
  no priority!

```

QoS Classification for Queue assignments

You may assign specific COS and DSCP values which should be mapped in general to certain traffic classes/ queues:

```

!
no qos rewrite cos
qos map cos 4 5 to traffic-class 3
qos map cos 2 3 to traffic-class 4
qos map dscp 46 to traffic-class 3
qos map traffic-class 2 5 to cos 0
qos map traffic-class 4 to cos 2
qos map traffic-class 3 to cos 4
qos map traffic-class 3 to dscp 46
qos map traffic-class 4 to dscp 34
!

```

Configuring ACL based QoS

You may create ACL filters to define traffic patterns for prioritizations and assign them to a **traffic class**.

In this example ICMP traffic from a certain host will be matched for further QoS treatments:

```

!
ip access-list audio
  10 permit icmp host 10.97.165.188 any
!
class-map type qos match-any AUDIOclass
  match ip access-group audio
!

```

Configuring Interfaces for QoS

Ethernet and port channel interfaces support three QoS trust modes:

- **CoS Trust:** Ports use inbound packet CoS field contents to derive the traffic class.
- **DSCP Trust:** Ports use inbound packets DSCP field contents to derive the traffic class.
- **Untrusted:** Ports use their default values to derive the traffic class, ignoring packet contents.

The **default mode** setting is **CoS trust for switched ports and DSCP trust for routed ports**.

In this example we are assigning Port 1 to Video traffic, Port 3 to Audio traffic and Port 49 will be rate shaped as it is an uplink port. All Ports are set to untrusted mode so that set COS and DSCP values will be ignored.

```
!  
vlan 50  
    name VIDEO-VLAN  
!  
vlan 100  
    name AUDIO-VLAN  
!  
vlan 1600  
    name GUEST-VLAN  
!  
vlan 1700  
    name IoT-VLAN  
!  
interface Ethernet1  
    description VIDEO  
    switchport access vlan 50  
    service-policy type qos input VIDEO  
    no qos trust  
    spanning-tree portfast  
!  
interface Ethernet3  
    description AUDIO  
    switchport access vlan 100  
    service-policy type qos input AUDIO  
    no qos trust  
    spanning-tree portfast  
!  
interface Ethernet49  
    description Uplink Port  
    mtu 9214  
    speed forced 1000full  
    switchport trunk allowed vlan 50,100,1600,1700  
    switchport mode trunk  
    service-profile Uplink  
    shape rate 500000  
!
```

Multi-Domain Segmentation Services (MSS)

Configuring MSS-G

Group-Based Multi-Domain Segmentation Services (MSS-G) allows applying policies on network segments rather than interfaces, VLANs or subnets. Individual Hosts or networks can be classified into segments based on prefixes which allows the definition of policies between segments to allow or block traffic between them.

Detailed information about the MSS-G feature can be found at:

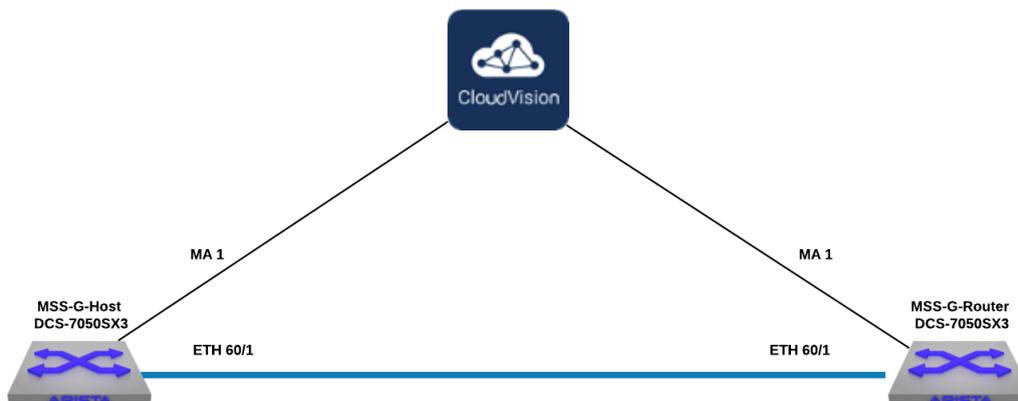
<https://eos.arista.com/eos-4-25-2f/group-based-multi-domain-segmentation-services-mss-group/>

As vEOS-lab does not support MSS-G two systems **7050SX3-48C12** with **EOS 4.27.1.1F** have been used to demonstrate MSS-G in the following example.

Although MSS-G can be configured directly on the systems via CLI here it has been provisioned via CloudVision Studios using **CVP 2021.2.2**.

Studios is still a Beta Feature in the used CVP version 2021.2.2 and has to be enabled on the General Setting tab.

Within the test setup one 7050SX3 system is serving as a host with a separate VRF configured for each VLAN interface and one 7050SX3 system is serving as an one armed Router where MSS-G rules are implemented.



MSS-G Host:

```
!  
username cvpadmin privilege 15 role network-admin secret password  
!  
daemon TerminAttr  
    exec /usr/bin/TerminAttr -ingestgrpcurl=10.83.30.160:9910 -cvcompression=gzip  
-ingestauth=token,/tmp/token  
-smashexcludes=ale,flexCounter,hardware,kni,pulse,strata  
-ingestexclude=/Sysdb/cell/1/agent,/Sysdb/cell/2/agent -ingestvrf=MGMT -taillogs  
    no shutdown  
!  
hostname MSS-G-Host  
!  
spanning-tree mode mstp  
!  
vlan 10  
    name VLAN-10  
!  
vlan 20  
    name VLAN-20  
!  
vlan 30  
    name VLAN-30  
!  
vlan 40  
    name VLAN-40  
!  
vlan 50  
    name VLAN-50  
!  
vrf instance MGMT  
!  
vrf instance vlan10  
!  
vrf instance vlan20  
!  
vrf instance vlan30  
!  
vrf instance vlan40  
!  
vrf instance vlan50  
!  
management api http-commands  
    protocol http  
    protocol unix-socket  
    no shutdown  
    !  
    vrf MGMT  
        no shutdown  
!  
interface Ethernet60/1  
    no shutdown
```

```

    switchport mode trunk
!
interface Management1
    no shutdown
    vrf MGMT
    ip address 10.100.164.109/24
!
interface Vlan10
    vrf vlan10
    ip address 192.168.10.1/24
!
interface Vlan20
    vrf vlan20
    ip address 192.168.20.1/24
!
interface Vlan30
    vrf vlan30
    ip address 192.168.30.1/24
!
interface Vlan40
    vrf vlan40
    ip address 192.168.40.1/24
!
interface Vlan50
    vrf vlan50
    ip address 192.168.50.1/24
!
ip routing
no ip routing vrf MGMT
ip routing vrf vlan10
ip routing vrf vlan20
ip routing vrf vlan30
ip routing vrf vlan40
ip routing vrf vlan50
!
ip route vrf MGMT 0.0.0.0/0 10.100.164.1
ip route vrf vlan10 0.0.0.0/0 192.168.10.250
ip route vrf vlan20 0.0.0.0/0 192.168.20.250
ip route vrf vlan30 0.0.0.0/0 192.168.30.250
ip route vrf vlan40 0.0.0.0/0 192.168.40.250
ip route vrf vlan50 0.0.0.0/0 192.168.50.250
!

```

MSS-G Router:

```
!  
username cvpadmin privilege 15 role network-admin secret password  
!  
daemon TerminAttr  
    exec /usr/bin/TerminAttr -ingestgrpcurl=10.83.30.160:9910 -cvcompression=gzip  
-ingestauth=token,/tmp/token  
-smashexcludes=ale,flexCounter,hardware,kni,pulse,strata  
-ingestexclude=/Sysdb/cell/1/agent,/Sysdb/cell/2/agent -ingestvrf=MGMT -taillogs  
    no shutdown  
!  
match-list input prefix-ipv4 segment-v4-VLAN10-20  
    match prefix-ipv4 192.168.10.0/24  
    match prefix-ipv4 192.168.20.0/24  
!  
match-list input prefix-ipv4 segment-v4-VLAN30-40  
    match prefix-ipv4 192.168.30.0/24  
    match prefix-ipv4 192.168.40.0/24  
!  
hostname MSS-G-Router  
!  
spanning-tree mode mstp  
!  
vlan 10  
    name VLAN-10  
!  
vlan 20  
    name VLAN-20  
!  
vlan 30  
    name VLAN-30  
!  
vlan 40  
    name VLAN-40  
!  
vlan 50  
    name VLAN-50  
!  
vrf instance MGMT  
!  
management api http-commands  
    protocol http  
    protocol unix-socket  
    no shutdown  
    !  
    vrf MGMT  
        no shutdown  
!  
interface Ethernet60/1  
    no shutdown  
    switchport mode trunk  
!
```

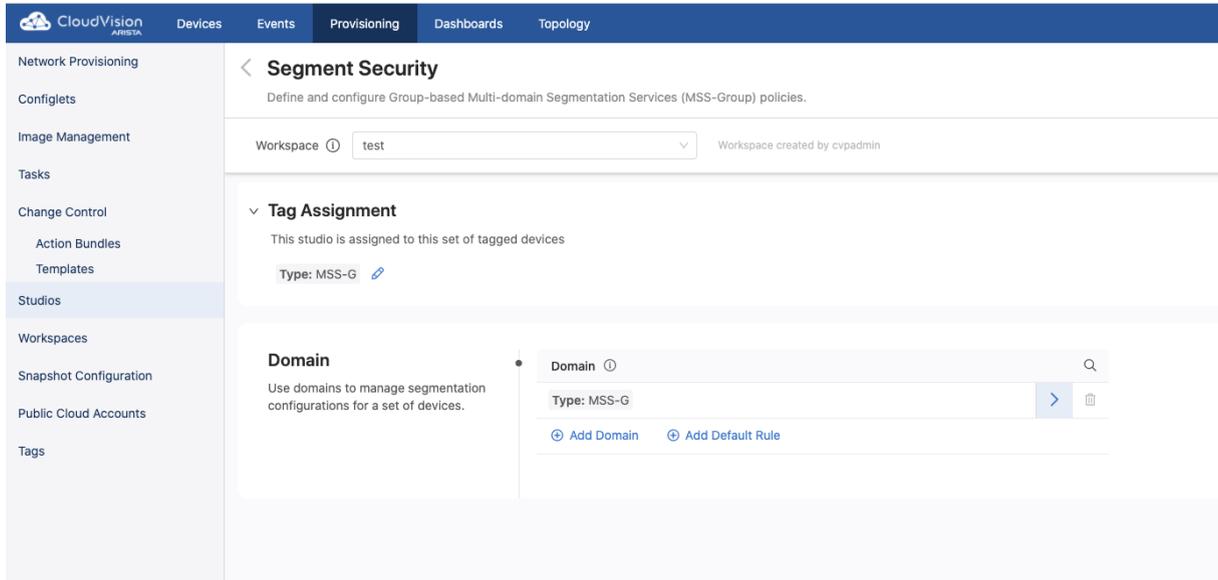
```

interface Management1
  no shutdown
  vrf MGMT
  ip address 10.100.164.108/24
!
interface Vlan10
  ip address 192.168.10.250/24
!
interface Vlan20
  ip address 192.168.20.250/24
!
interface Vlan30
  ip address 192.168.30.250/24
!
interface Vlan40
  ip address 192.168.40.250/24
!
interface Vlan50
  ip address 192.168.50.250/24
!
router segment-security
  no shutdown
  no segment policy policy-drop-all default
  !
  vrf default
    segment VLAN10-20
      definition
        match prefix-ipv4 segment-v4-VLAN10-20
      !
      policies
        from VLAN10-20 policy policy-drop-all
        from VLAN30-40 policy policy-drop-all
      !
    segment VLAN30-40
      definition
        match prefix-ipv4 segment-v4-VLAN30-40
      !
      policies
        from VLAN10-20 policy policy-drop-all
    !
  !
ip routing
no ip routing vrf MGMT
!
ip route vrf MGMT 0.0.0.0/0 10.100.164.1
!

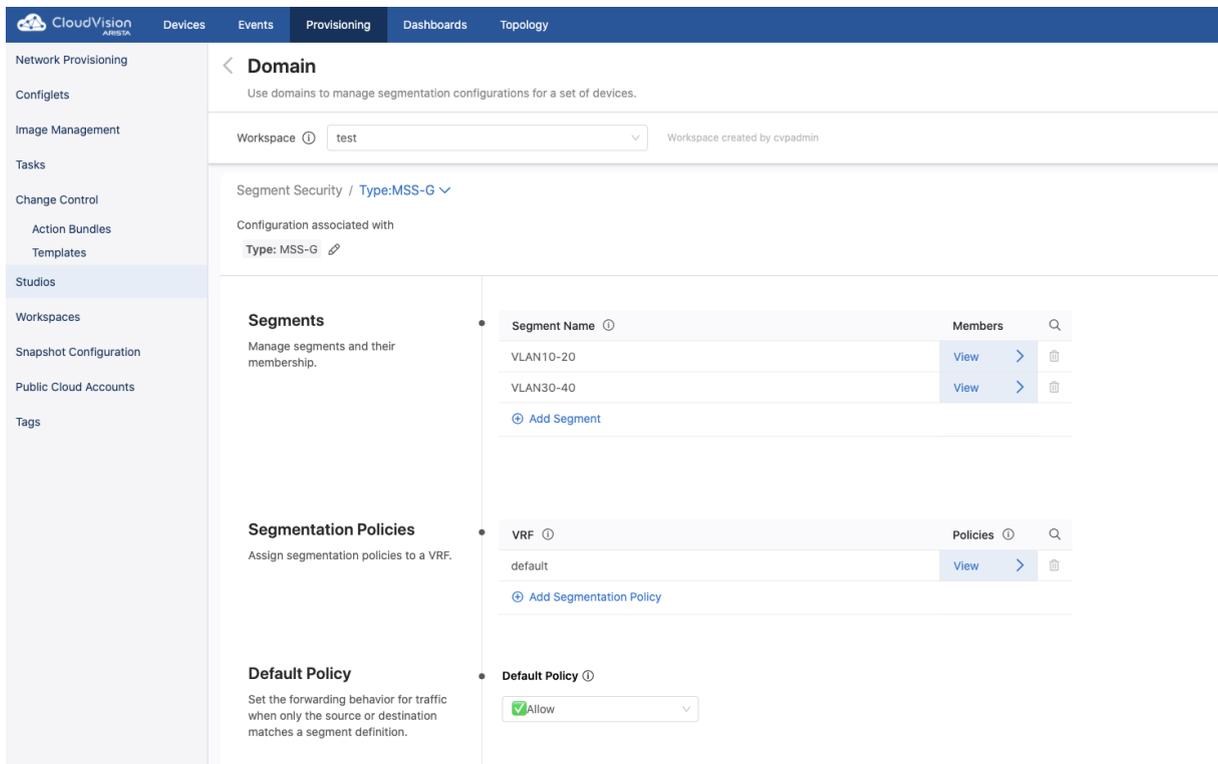
```

Configuring MSS-G on CloudVision with Studios “Segment Security”

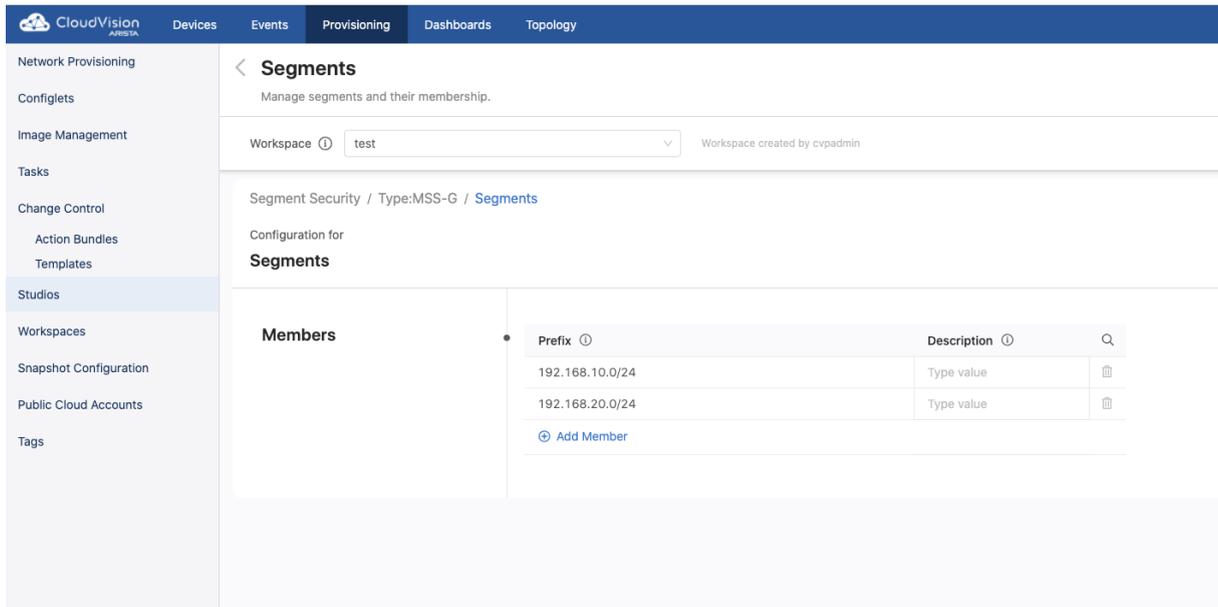
Segment Security is a built-in Studio which can be used to set up MSS-G on selected devices. Before using the Studio a **Tag** named “**Type: MSS-G**” has been assigned to **MSS-G-Router**. The used **Workbook** has the name “**test**”



The default forwarding policy has been set to “**Allow**”:



Segment "VLAN10-20" Members:

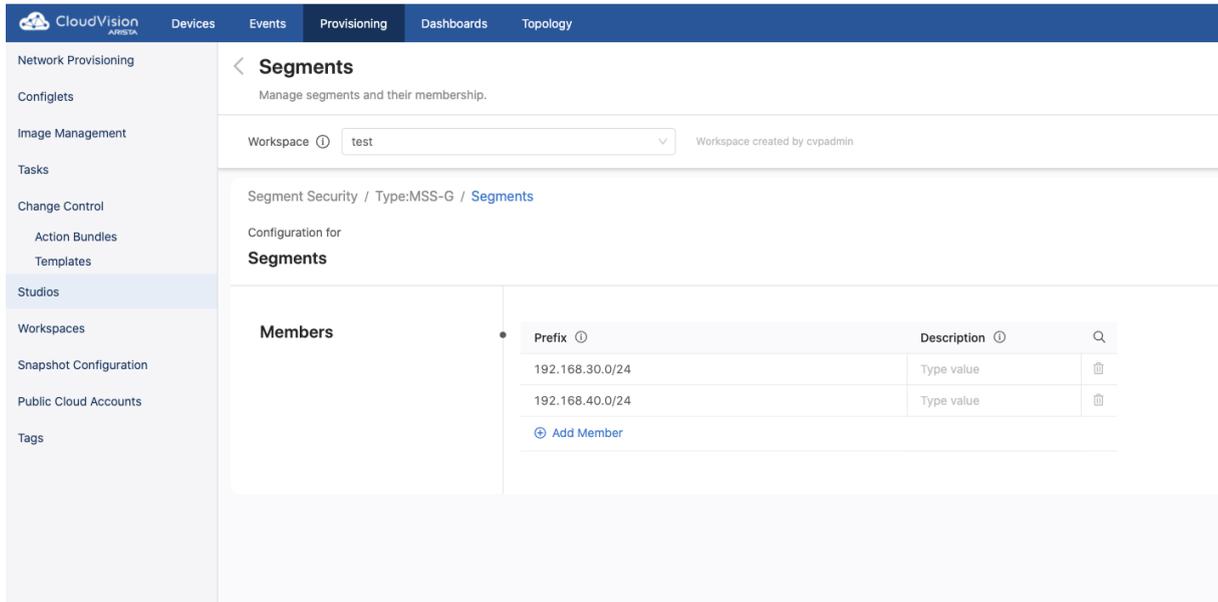


The screenshot shows the CloudVision Provisioning interface. The left sidebar contains navigation options: Network Provisioning, Configlets, Image Management, Tasks, Change Control (Action Bundles, Templates), Studios, Workspaces, Snapshot Configuration, Public Cloud Accounts, and Tags. The main content area is titled "Segments" and includes a workspace dropdown set to "test". Below this, the breadcrumb "Segment Security / Type:MSS-G / Segments" is visible. The "Configuration for Segments" section features a "Members" table with two entries:

Prefix	Description	
192.168.10.0/24	Type value	🗑️
192.168.20.0/24	Type value	🗑️

An "Add Member" link is located below the table.

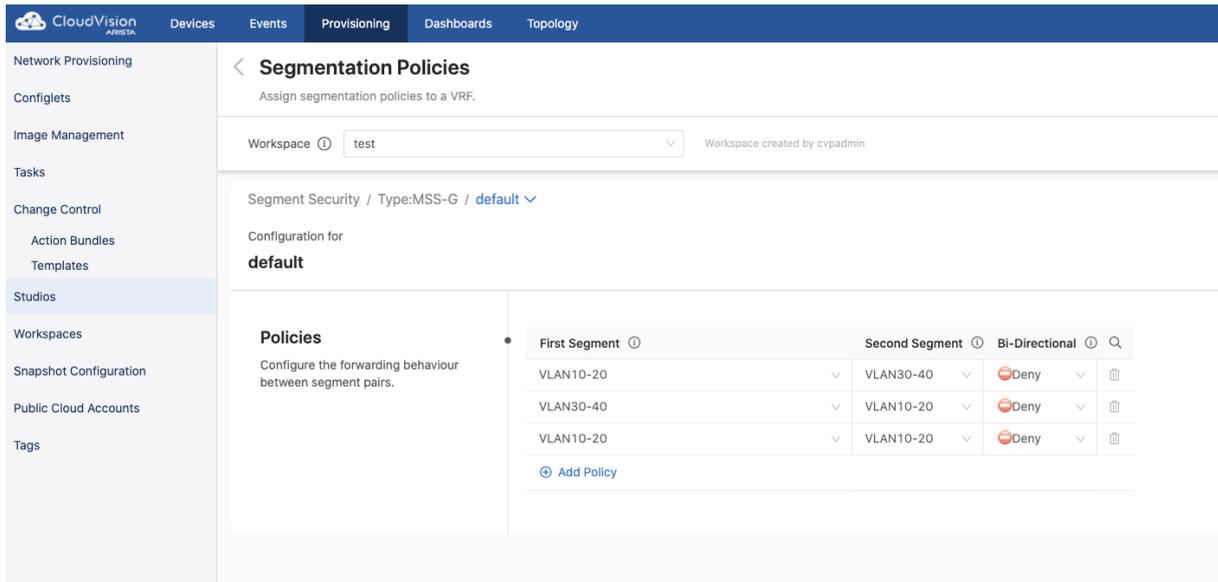
Segment "VLAN30-40" Members:



The screenshot shows the CloudVision Provisioning interface for a different segment. The layout is identical to the previous one, but the "Members" table contains two different entries:

Prefix	Description	
192.168.30.0/24	Type value	🗑️
192.168.40.0/24	Type value	🗑️

An "Add Member" link is located below the table.



Testing MSS-G

After assigning the Studio to the MSS-G Router device, hosts within VLAN 10 or VLAN 20 can only ping hosts within VLAN 50 - they even cannot ping other hosts in VLAN 10 or VLAN 20.

Hosts within VLAN 30 or VLAN 40 can ping each other and in addition they can ping hosts within VLAN 50 - they cannot ping hosts within VLAN 10 or VLAN 20.

MSS-G-Host#

```
MSS-G-Host#ping vrf vlan10 192.168.20.1
```

```
PING 192.168.20.1 (192.168.20.1) 72(100) bytes of data.
```

```
--- 192.168.20.1 ping statistics ---
```

```
5 packets transmitted, 0 received, 100% packet loss, time 40ms
```

```
MSS-G-Host#ping vrf vlan10 192.168.30.1
```

```
PING 192.168.30.1 (192.168.30.1) 72(100) bytes of data.
```

```
--- 192.168.30.1 ping statistics ---
```

```
5 packets transmitted, 0 received, 100% packet loss, time 40ms
```

```
MSS-G-Host#ping vrf vlan10 192.168.40.1
```

```
PING 192.168.40.1 (192.168.40.1) 72(100) bytes of data.
```

```
--- 192.168.40.1 ping statistics ---
```

```
5 packets transmitted, 0 received, 100% packet loss, time 40ms
```

```
MSS-G-Host#ping vrf vlan10 192.168.50.1
```

```
PING 192.168.50.1 (192.168.50.1) 72(100) bytes of data.
```

```
80 bytes from 192.168.50.1: icmp_seq=1 ttl=63 time=0.198 ms
```

```
80 bytes from 192.168.50.1: icmp_seq=2 ttl=63 time=0.068 ms
```

```
80 bytes from 192.168.50.1: icmp_seq=3 ttl=63 time=0.066 ms
```

```
80 bytes from 192.168.50.1: icmp_seq=4 ttl=63 time=0.066 ms
```

80 bytes from 192.168.50.1: icmp_seq=5 ttl=63 time=0.067 ms

--- 192.168.50.1 ping statistics ---

5 packets transmitted, 5 received, 0% packet loss, time 0ms

rtt min/avg/max/mdev = 0.066/0.093/0.198/0.052 ms, ipg/ewma 0.142/0.143 ms

MSS-G-Host#ping vrf vlan30 192.168.10.1

PING 192.168.10.1 (192.168.10.1) 72(100) bytes of data.

--- 192.168.10.1 ping statistics ---

5 packets transmitted, 0 received, 100% packet loss, time 40ms

MSS-G-Host#ping vrf vlan30 192.168.20.1

PING 192.168.20.1 (192.168.20.1) 72(100) bytes of data.

--- 192.168.20.1 ping statistics ---

5 packets transmitted, 0 received, 100% packet loss, time 40ms

MSS-G-Host#ping vrf vlan30 192.168.40.1

PING 192.168.40.1 (192.168.40.1) 72(100) bytes of data.

80 bytes from 192.168.40.1: icmp_seq=1 ttl=63 time=0.186 ms

80 bytes from 192.168.40.1: icmp_seq=2 ttl=63 time=0.067 ms

80 bytes from 192.168.40.1: icmp_seq=3 ttl=63 time=0.069 ms

80 bytes from 192.168.40.1: icmp_seq=4 ttl=63 time=0.073 ms

80 bytes from 192.168.40.1: icmp_seq=5 ttl=63 time=0.065 ms

--- 192.168.40.1 ping statistics ---

5 packets transmitted, 5 received, 0% packet loss, time 0ms

rtt min/avg/max/mdev = 0.065/0.092/0.186/0.047 ms, ipg/ewma 0.139/0.137 ms

MSS-G-Host#ping vrf vlan30 192.168.50.1

PING 192.168.50.1 (192.168.50.1) 72(100) bytes of data.

80 bytes from 192.168.50.1: icmp_seq=1 ttl=63 time=0.156 ms

80 bytes from 192.168.50.1: icmp_seq=2 ttl=63 time=0.064 ms

80 bytes from 192.168.50.1: icmp_seq=3 ttl=63 time=0.063 ms

80 bytes from 192.168.50.1: icmp_seq=4 ttl=63 time=0.062 ms

80 bytes from 192.168.50.1: icmp_seq=5 ttl=63 time=0.063 ms

--- 192.168.50.1 ping statistics ---

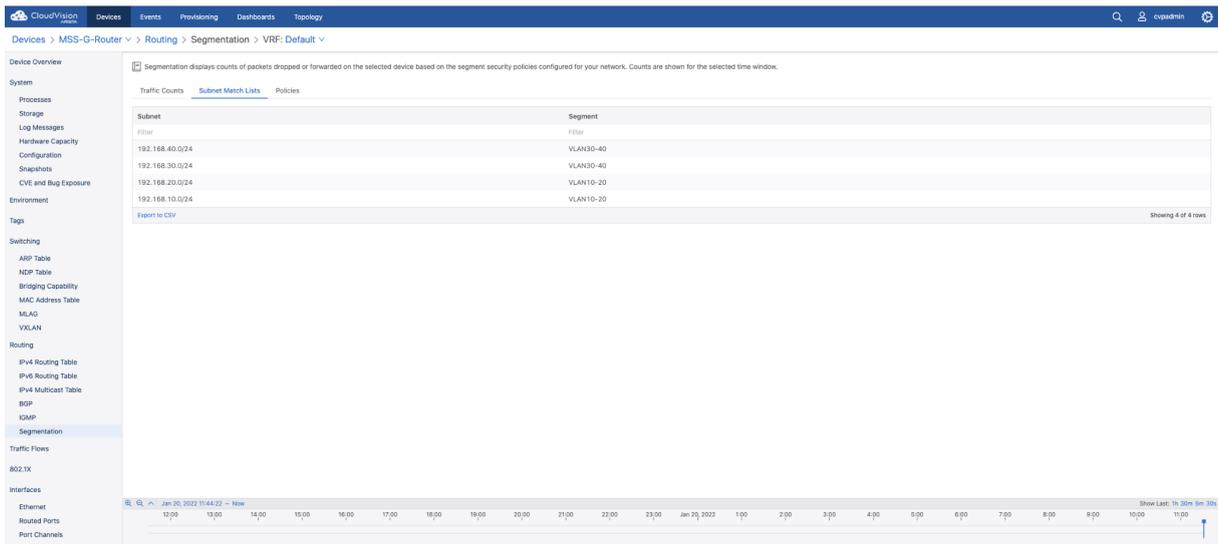
5 packets transmitted, 5 received, 0% packet loss, time 0ms

rtt min/avg/max/mdev = 0.062/0.081/0.156/0.038 ms, ipg/ewma 0.133/0.117 ms

MSS-G-Host#

Monitoring MSS-G in CloudVision

Within CloudVision the blocked Traffic and further information can be monitored:



CloudVision Devices Events Provisioning Dashboards Topology cupadm@

Devices > MSS-G-Router > Routing > Segmentation > VRF: Default

Device Overview

System

- Processes
- Storage
- Log Messages
- Hardware Capacity
- Configuration
- Snapshots
- CVE and Bug Exposure
- Environment
- Tags

Switching

- ARP Table
- NDP Table
- Bridging Capability
- MAC Address Table
- MLAG
- VLAN

Routing

- IPv4 Routing Table
- IPv6 Routing Table
- IPv4 Multicast Table
- BGP
- IGMP

Segmentation

Traffic Flows

802.1X

Interfaces

- Ethernet
- Routed Ports
- Port Channels

Segmentation displays counts of packets dropped or forwarded on the selected device based on the segment security policies configured for your network. Counts are shown for the selected time window.

Traffic Counts Subnet Match Lists Policies

Subnet	Segment
Filter	Filter
193.168.40.0/24	VLAN30-40
192.168.30.0/24	VLAN30-40
192.168.20.0/24	VLAN10-20
192.168.10.0/24	VLAN10-20

Export to CSV Showing 4 of 4 rows

Jan 30, 2022 11:44:32 - Now

12:00 13:00 14:00 15:00 16:00 17:00 18:00 19:00 20:00 21:00 22:00 23:00 Jan 20, 2022 1:00 2:00 3:00 4:00 5:00 6:00 7:00 8:00 9:00 10:00 11:00

Show Last: 1h 30m 5m 30s

MACsec Configuration

Configuring MACsec

To configure MACsec on an Arista device you will need to purchase a valid MACsec license. Please contact your Account team for detailed pricing information.

MACsec licenses are tied to a switch serial number so that every switch running MACsec requires a separate license of its own.

After a license has been purchased you will get a json file which you will have to download to your MACsec capable Arista device:

```
DC1-ITCORE1#dir
```

```
Directory of flash:/
```

```
-rw-          2541          Nov 10 17:11  AsuFastPktTransmit.log
-rw-   917565557          Oct 22 16:23  EOS-4.24.0F.swi
-rw-   949208936          Nov 10 13:51  EOS-4.24.3M.swi
drwx          4096          Oct 22 16:26  Fossil
-rw-          1562          Nov 10 17:11  SsuRestore.log
-rw-          1562          Nov 10 17:11  SsuRestoreLegacy.log
-rw-          9216          Nov 10 17:12  aquota.user
drwx          4096          Nov 10 17:11  archive
-rw-           27          Nov 10 13:51  boot-config
-rw-           1          Nov 10 13:51  boot-extensions
drwx          4096          Nov 10 17:23  debug
drwx          4096          Oct 22 16:26  fastpkttx.backup
-rw-          3215          Nov 11 09:42  license_MACsec_1.json
drwx         16384          Oct 22 16:22  lost+found
drwx          4096          Nov 11 09:42  persist
drwx          4096          Oct 22 16:39  schedule
-rw-         13626          Nov 11 09:13  startup-config
-rw-           0          Nov 10 11:38  zerotouch-config
```

```
235145605120 bytes total (231147270144 bytes free)
```

In the next step you will have to import the license on the Arista device:

```
DC1-ITCORE1#license import flash:license_MACsec_1.json
```

```
DC1-ITCORE1#show license
```

```
Customer name:      Customer Test1
```

```
System Serial number:  TM020260421
```

```
System MAC address:  fcbd.6711.b123
```

```
Domain name:        Unknown
```

```
Platform:           DCS-7808-CH
```

```
License feature:    MACsec
```

```

License parameter: None
Count:            2
Start:            2020-11-10 00:00:00
Expiration:       2021-01-09 00:00:00
Active:           yes
License source:   File

```

DC1-ITCORE1#

When looking into the license file itself you will find the **License serial number** where you will need the first 8 digits:

```

{
  "BindingInfo": {
    "DomainAddress": "",
    "Platform": {
      "Physical": [],
      "vEOS": []
    },
    "SerialNumber": "TMO20260421",
    "SystemMAC": ""
  },
  "CustomerName": " Customer Test1",
  "Features": {
    "MACsec": [
      {
        "BehaviorModifier": {
          "DaysAllowedPastExpiration": 0
        },
        "Count": 2,
        "Valid": {
          "NotAfter": "2021-01-09T00:00:00Z",
          "NotBefore": "2020-11-10T00:00:00Z"
        },
        "Value": ""
      }
    ]
  },
  "LicenseFileVersion": "1.2",
  "LicenseSerialNumber": "37387f86-624d-4de5-bad4-712ab3df8ea2",
  "Signature": {
    "Hash": "04220550374482b6f93e794222ab284bcd6b84f7d41d1f87502b543308df90c6",
    "Signature":
"3045022045617e72667b5a12995a0346506978f144d13f081bf075854c8a4d539a58ac1a022100c72
b41a4eb0fc936f335c84429c49b6ff7684155f1ac8a9c712abb5e6019b639",
    "SigningCertPEM": "-----BEGIN
CERTIFICATE-----\nMIIF6TCCA9GgAwIBAgITFgAAAAAN8An0siquXTAAAAAAAANBzANBgkqhkiG9w0BAQsF
\nADA6MTgwNgYDVQQDEy9Bcm1zdGEgTmV0d29ya3MgSW50ZXJ1eWwgSVQgUm9vdCBD\NzXJ0IEF1dGhvcm
l0eTAeFw0xNzA1MzEyMjU0MjhaFw0yNzA1MzEyMzA0MjhaMGkx\nEzARBgoJkiaJk/IsZAEZFgNjb20xHj
AcBgoJkiaJk/IsZAEZFg5hcm1zdGFuZXR3R3\nb3JrczEyMDAGA1UEAxMpQXJpc3RlShVQQtSUNBIEVDRFNBIIE

```

```

lzc3VpbmcgQ2VydCBB\ndXRob3JpdHkwWTATBgcqhkJOPQIBBggqhkjOPQMBBwNCAARPqNqFCrbuLJ1Ewk
Kg\n3RLdwkzx8kZxtnMmU0xTT1sLN8oNKMp4yFanvVFxwd4PydTeI1JUAZQ5a73dXqom\npHk2o4ICgJCC
An4wEAYJKwYBBAGCNxUBBAMCAQAwHQYDVR00BBYEFHO110dbJLhN\nGJaktFHcGpd1P31KMFcGA1UdIARQ
ME4wTAYMKwYBAgerBwEEAgEDMDwwOgYIKwYB\nbQUHAgEWLmh0dHA6Ly9pdC1wa2kuYXJpc3RhbmV0d29y
a3MuY29tL3BraS9jcHMu\naHRtbAAwGQYJKwYBBAGCNxQCBAwCgBTAHUAYgBDAEEwCwYDVR0PBAQDAgGG
MA8G\nA1UdEwEB/wQFMAMBAf8wHwYDVR0jBBgwFoAULe/a5u/PHHdRv1oCqSjNlAct4cow\ngYEGA1UdHw
R6MHgwdqB0oHKGNmh0dHA6Ly9pdC1wa2kuYXJpc3RhbmV0d29ya3Mu\nY29tL3BraS9Bcm1zdGFJVC1ST0
9ULmNybiY4aHR0cDovL210LXBraTAXLmFyaXN0\nYw51dHdvcmtzLmNvbS9wa2kvQXJpc3RhSVQ0tUk9PVC
5jcmwwggESBggrBgEFBQcB\nAQSCAQQwggEAMH0GCCsGAQUFBzACHnFodHRwOi8vaXQtcGtpLmFyaXN0YW
51dHdv\ncmtzLmNvbS9wa2kvQXJpc3RhSVQ0tUk9PVEFyaXN0YSUyME51dHdvcmtzJTIwSW50\nZXJlYyYwLm
MjBjVCUyMFJvb3Q1MjBDZXJ0JTIwQXV0aG9yaXR5LmNydDB/BggrBgEF\nbQcwAoZzaHR0cDovL210LXBra
aTAXLmFyaXN0YW51dHdvcmtzLmNvbS9wa2kvQXJpc3RhSVQ0tUk9PVEFyaXN0YSUyME51dHdvcmtzJTIw
SW50ZXJlYyYwLmMjBjVCUyMFJv\nb3Q1MjBDZXJ0JTIwQXV0aG9yaXR5LmNydDANBgkqhkiG9w0BAQsFAAOC
AgEAotuJ\n/hLxK1JOs85pYfbDR8bg5HzsEVH0rc/fjUf85e3riGMh+PaQHj5L++Ah9cMmUujh\n+bPq8y
crjhYwi0IZGLjwJuGWHQ2TMXqB4o+1mKgchpR0gA31pcZCANT5atRghrQ\nnhTMHN3L2CZRDN3JSCD1xQb
W/WVDY1Hv6IpWkd2orem/lgQfKVvV1keB3YPJn5Hka\nh1x37mksQ9KEh7v52Tira5JnP67mUdT1C+gvdG
F3DJk3Lg6GWX9Uxo1vG28AmJOU\nn0n28ek5Ynh0T3uQ+jkMoJEIlyH1fKZ6zyK0sf+yLRb7brkfssZDrRI
atxKEkv60c\nnh4kXO2mvmMJxQDf7VvGXEC3fSRURLwPz//6JMx942iOKsES8ZT9nT2q9MxJXfInn\nn3EcK
GmPWKQR4n2qHfmq6sfk2eFBUYIrZBm9RUBvbyLZLC0v2KxJ7FFZ9LV1jp5An\nAyHLJUMqqw/kvUUvUq1
bI/PtEO1Nc9Ndt/3yeh+HByzIw8/f+gjKkUjQpVncuqS\nnkFotBPNNj/LjbQD40R/tJ0z/8sPXCGJuo4mE
9s/MwnWmkAHxpZyCcMB1Np3LkJK\nnFHcsVb36Vc1v5XWDe5AxU+0sQjEB4LGP7nYo8wjvSZIpyXRiAmD
RGuAGi/W/W3F\nn6hEQ661JK4KPJvoQsMqYa0/TkZPIXEAAbgEDkmj0=\n-----END CERTIFICATE-----"
}
}

```

In order to configure MACsec on the Arista device just copy **the first 8 digits** of the License serial number within the json file to configure the mac security license on the system. You also will have to configure a **mac security profile** on each port where you want to enable MACsec – **especially on each physical port which is part of a Port Channel:**

```

mac security
  license productTest 37387f86
  !
  profile macsec-test
    cipher aes256-gcm
    key 0abc1234 7 055B575F731C1D594D5542425D
    mka session rekey-period 180
  !
interface Port-Channel1
  description PortChannel1_ITCORE2
  no switchport
  ip address 154.4.1.105/30
  !
interface Ethernet3/15/1
  description P2P_LINK_TO_DC1-SPINE1_Ethernet12/1
  no switchport
  ipv6 enable
  ipv6 address fe80::1/64 link-local
  mac security profile macsec-test
  !

```

```

interface Ethernet3/10/1
  description DC1-ITCORE2_e3/10/1
  channel-group 1 mode active
  mac security profile macsec-test
!
interface Ethernet4/11/1
  description DC1-ITCORE2_e3/11/1
  channel-group 1 mode active
  mac security profile macsec-test
!

```

Checking MACsec Configuration

Please make sure that MACsec is configured on both ports which are physically linked together before checking the status..

```
DC1-ITCORE1#show mac security status
```

```

Active Profiles:      1
Data Delay Protection: No
FIPS Mode:           No
Secured Interfaces:  5
License:              Enabled

```

```
DC1-SPINE1A#sho mac security interface
```

Interface	SCI	Controlled Port	Key in Use
Ethernet3/10/1	fc:bd:67:5d:f6:96::41 60104eb49ec46f1d60f31ccb:13	True	
Ethernet3/15/1	fc:bd:67:5d:f6:9a::45 9a02f9105ea310d1e52961fc:13	True	
Ethernet4/11/1	fc:bd:67:5d:f6:9e::49 3a4df6e7991369b609f1f596:11	True	

```
DC1-ITCORE1#
```

Using eAPI

eAPI communicates over **HTTP** or **HTTPS** and delivers outputs in **JSON format**.

Additional information can be found at:

https://www.arista.com/assets/data/pdf/Whitepapers/Arista_eAPI_FINAL.pdf

EOS setup

You have to **enable API management** on the EOS systems and this has to be enabled on configured **VRFs** also if used!

```
username admin privilege 15 role network-admin secret arista
!
vrf instance MGMT
!
interface Management1
  vrf MGMT
  ip address 192.168.1.199/24
!
ip route vrf MGMT 0.0.0.0/0 192.168.1.1
!
management api http-commands
  no shutdown
!
  vrf MGMT
    no shutdown
```

JSON output via CLI

You can directly read outputs in **JSON format**:

```
Router-1#show interfaces ethernet 1 | json
{
  "interfaces": {
    "Ethernet1": {
      "lastStatusChangeTimestamp": 1653643100.5469024,
      "lanes": 0,
      "name": "Ethernet1",
      "interfaceStatus": "connected",
      "autoNegotiate": "unknown",
      "loopbackMode": "loopbackNone",
      "l2Mru": 0,
      "interfaceStatistics": {
        "inBitsRate": 0.0,
        "inPktsRate": 0.0,
        "outBitsRate": 0.0,
        "updateInterval": 300.0,
        "outPktsRate": 0.0
      },
      "mtu": 1500,
      "hardware": "ethernet",
      "duplex": "duplexFull",
      "bandwidth": 1000000000,
      "forwardingModel": "routed",
```

```

"lineProtocolStatus": "up",
"l3MtuConfigured": false,
"interfaceCounters": {
  "outBroadcastPkts": 2,
  "outUcastPkts": 3500,
  "totalOutErrors": 0,
  "inMulticastPkts": 7822,
  "counterRefreshTime": 1653659071.724403,
  "inBroadcastPkts": 10921,
  "outputErrorsDetail": {
    "deferredTransmissions": 0,
    "txPause": 0,
    "collisions": 0,
    "lateCollisions": 0
  },
  "inOctets": 2635170,
  "outDiscards": 0,
  "outOctets": 3891260,
  "inUcastPkts": 3002,
  "inTotalPkts": 21745,
  "inputErrorsDetail": {
    "runtFrames": 0,
    "rxPause": 0,
    "fcsErrors": 0,
    "alignmentErrors": 0,
    "giantFrames": 0,
    "symbolErrors": 0
  },
  "linkStatusChanges": 2,
  "outMulticastPkts": 533,
  "totalInErrors": 0,
  "inDiscards": 0
},
"interfaceAddress": [
  {
    "secondaryIpsOrderedList": [],
    "broadcastAddress": "255.255.255.255",
    "virtualSecondaryIps": {},
    "dhcp": false,
    "secondaryIps": {},
    "primaryIp": {
      "maskLen": 24,
      "address": "192.168.1.199"
    },
    "virtualSecondaryIpsOrderedList": [],
    "virtualIp": {
      "maskLen": 0,
      "address": "0.0.0.0"
    }
  }
],
"physicalAddress": "50:00:00:cb:38:c2",
"description": ""
}
}
}

```

eAPI Web Interface

To see inputs and outputs detailed within a Web Interface just point your Browser to the IP address of the EOS system:

https://192.168.1.199/explorer.html

A login Window will pop up where you will have to type in your **username** and **password**.

Now you can type in “show” commands and directly see the output in JSON format:

The screenshot displays the ARISTA Command API Explorer web interface. At the top, there are navigation tabs: Explorer, Overview, and Command Documentation. Below these are buttons for Simple Request and Script Editor. The main section is titled "Simple eAPI request editor" and contains a form for crafting requests. The API Endpoint is set to https://192.168.1.199/command-api, and the Version is 1. The Commands field contains "show interfaces Ethernet 1". Other settings include Format (json), Timestamps (false), AutoComplete (false), ExpandAliases (false), IncludeErrorDetail (false), and ID (EapiExplorer-1). A Submit POST request button is located at the bottom right of the form.

Request Viewer

```
1 {
2   "jsonrpc": "2.0",
3   "method": "runCmds",
4   "params": {
5     "format": "json",
6     "timestamps": false,
7     "autoComplete": false,
8     "expandAliases": false,
9     "includeErrorDetail": false,
10    "cmds": [
11      "show interfaces Ethernet 1"
12    ],
13    "version": 1
14  },
15  "id": "EapiExplorer-1"
16 }
```

Response Viewer

```
1 {
2   "jsonrpc": "2.0",
3   "id": "EapiExplorer-1",
4   "result": [
5     {
6       "interfaces": {
7         "Ethernet1": {
8           "lastStatusChangeTimestamp": 1653643100.5469022,
9           "lanes": 0,
10          "name": "Ethernet1",
11          "interfaceStatus": "connected",
12          "autoNegotiate": "unknown",
13          "loopbackMode": "loopbackNone",
14          "l2Mru": 0,
15          "interfaceStatistics": {
16            "inBitsRate": 0,
17            "inPktsRate": 0,
18            "outBitsRate": 0,
19            "updateInterval": 300,
20            "outPktsRate": 0
21          },
22          "mtu": 1500,
23          "hardware": "ethernet",
24          "duplex": "duplexFull",
25          "bandwidth": 1000000000,
```

Script Editor:

Within the Script Editor you can easily test Scripts using the **Javascript programming language**.

The screenshot displays the ARISTA Command API interface. At the top, there is a navigation bar with 'ARISTA Command API', 'Explorer', 'Overview', and 'Command Documentation'. Below this, there are tabs for 'Simple Request' and 'Script Editor', with 'Script Editor' being the active tab.

The main content area is titled 'Interactive eAPI scripting environment'. It contains a paragraph explaining that users can interactively play with eAPI using JavaScript. Below the text is a link for 'Show API documentation'.

The script editor area shows a list of 'EXAMPLE SCRIPTS' on the left, including 'Print switch address', 'Show active intfs', 'Show configured ACLs', 'Create example ACL', and 'Remove example ACL'. The 'Print switch address' script is selected and its code is displayed in the main editor area:

```
1 // This script prints out the switch's MAC address
2
3 // Create an EapiClient object...
4 var eapi = new EapiClient();
5 // and form the eAPI request:
6 var request = eapi.runCmds({'version': 1, 'cmds': ['show version']});
7
8 request.done(function(result){
9     // Hooray, the switch replied with data! Since we sent 1 command,
10    // the data we care about is at result[0]. Extract the MAC address
11    // and print it to the console:
12    var macAddr = result[0]["systemMacAddress"];
13    logMessage("The switch's system MAC address is " + macAddr);
14 });
```

Below the script editor, there is a 'SCRIPT OUTPUT' section with checkboxes for 'Show requests' and 'Show responses', and a 'Run Script' button. The output shows the following JSON:

```
Request [id: EapiExplorer-0]: show version
{
  "jsonrpc": "2.0",
  "method": "runCmds",
  "params": {
    "version": 1,
    "cmds": [
      "show version"
    ]
  },
  "id": "EapiExplorer-0"
}

Response [id: EapiExplorer-0]: show version
{
  "jsonrpc": "2.0",
  "id": "EapiExplorer-0",
  "result": [
    {
      "memTotal": 4002492,
      "uptime": 16362.91,
      "modelName": "vEOS",
      "internalVersion": "4.26.0.1F-21994874.42601F",
    }
  ]
}
```

```
▼ Response [id: EapiExplorer-0]: show version
  "systemMacAddress": "50:00:00:cb:38:c2",
  "bootupTimestamp": 1653642931,
  "memFree": 3072120,
  "version": "4.26.0.1F",
  "configMacAddress": "00:00:00:00:00:00",
  "isIntlVersion": false,
  "internalBuildId": "e41b7ab2-f5ed-45cb-ba9c-f320cb81332f",
  "hardwareRevision": "",
  "hwMacAddress": "00:00:00:00:00:00",
  "architecture": "i686"
}
```

The switch's system MAC address is 50:00:00:cb:38:c2

Remote Python scripting

It is possible to write complete scripts on **external Systems** which can be run remotely on EOS devices by using **Python**.

Within this Manual we will be using an external **Ubuntu Server 20.04.3**

Prerequisites:

```
mkrausch@ubuntu:~$ sudo apt-get update
```

```
mkrausch@ubuntu:~$ sudo apt-get install python3-pip
```

```
Reading package lists... Done
```

```
Building dependency tree
```

```
Reading state information... Done
```

```
The following additional packages will be installed:
```

```
binutils binutils-common binutils-x86-64-linux-gnu build-essential dpkg-dev
```

```
fakeroot g++ g++-9 gcc gcc-9
```

```
libalgorithm-diff-perl libalgorithm-diff-xs-perl libalgorithm-merge-perl
```

```
libasan5 libatomic1 libbinutils
```

```
libc-dev-bin libc6 libc6-dbg libc6-dev libcrypt-dev libctf-nobfd0 libctf0
```

```
libexpat1-dev libfakeroot
```

```
libgcc-9-dev libitm1 liblsan0 libpython3-dev libpython3.8-dev libquadmath0
```

```
libstdc++-9-dev libtsan0
```

```
libubsan1 linux-libc-dev make manpages-dev python-pip-whl python3-dev
```

```
python3-wheel python3.8-dev
```

```
zlib1g-dev
```

```
Suggested packages:
```

```
binutils-doc debian-keyring g++-multilib g++-9-multilib gcc-9-doc gcc-multilib
```

```
autoconf automake libtool
```

```
flex bison gcc-doc gcc-9-multilib gcc-9-locales glibc-doc libstdc++-9-doc
```

```
make-doc
```

```
The following NEW packages will be installed:
```

```
binutils binutils-common binutils-x86-64-linux-gnu build-essential dpkg-dev
```

```
fakeroot g++ g++-9 gcc gcc-9
```

```
libalgorithm-diff-perl libalgorithm-diff-xs-perl libalgorithm-merge-perl
```

```
libasan5 libatomic1 libbinutils
```

```
libc-dev-bin libc6-dev libcrypt-dev libctf-nobfd0 libctf0 libexpat1-dev
```

```
libfakeroot libgcc-9-dev libitm1
```

```
liblsan0 libpython3-dev libpython3.8-dev libquadmath0 libstdc++-9-dev libtsan0
```

```
libubsan1 linux-libc-dev
```

```
make manpages-dev python-pip-whl python3-dev python3-pip python3-wheel
```

```
python3.8-dev zlib1g-dev
```

```
The following packages will be upgraded:
```

```
libc6 libc6-dbg
```

```
2 upgraded, 41 newly installed, 0 to remove and 153 not upgraded.
```

```
Need to get 66.3 MB of archives.
```

```
After this operation, 149 MB of additional disk space will be used.
```

```
Do you want to continue? [Y/n] y
```

```
Get:1 http://de.archive.ubuntu.com/ubuntu focal-updates/main amd64 libc6-dbg amd64  
2.31-0ubuntu9.9 [20.0 MB]
```

```
Get:2 http://de.archive.ubuntu.com/ubuntu focal-updates/main amd64 libc6 amd64  
2.31-0ubuntu9.9 [2722 kB]
```

```
...
```

```
Setting up build-essential (12.8ubuntu1.1) ...
```

```
Setting up python3-dev (3.8.2-0ubuntu2) ...
```

```
Processing triggers for man-db (2.9.1-1) ...
```

Processing triggers for libc-bin (2.31-0ubuntu9.7) ...

```
mkrausch@ubuntu:~$ sudo pip install jsonrpclib
Collecting jsonrpclib
  Downloading jsonrpclib-0.2.1-py3-none-any.whl (15 kB)
Installing collected packages: jsonrpclib
Successfully installed jsonrpclib-0.2.1
```

Using a Python Editor to write a Program:

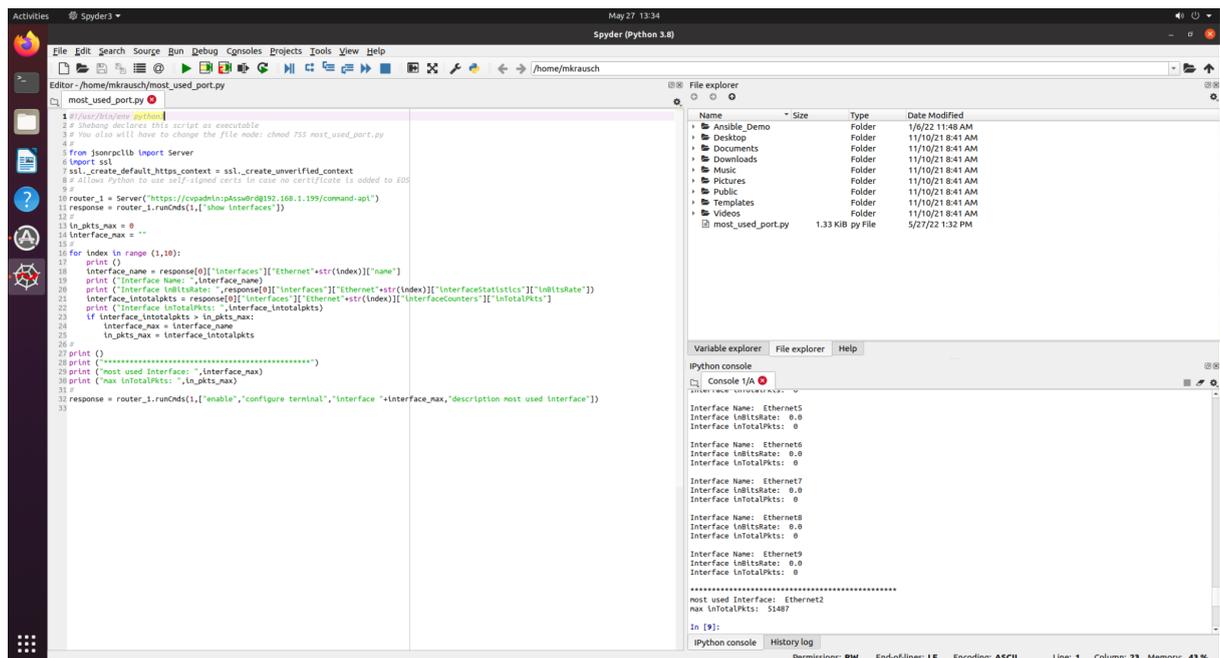
When using an **Ubuntu Server** you will need to install the **Ubuntu Desktop** first:

```
mkrausch@ubuntu:~$ sudo apt-get install ubuntu-desktop
```

We are using here the Editor “**Spyder**” which we have to install first on the Ubuntu Server

```
mkrausch@ubuntu:~$ sudo apt-get install spyder
```

On the installed Ubuntu Desktop you can start the **Spyder Editor** to create a Python script, edit an existing one or even run the script and watch the outputs in a Console Window:



```
1 #!/usr/bin/env python
2 # Shuang declares this script as executable
3 # You also will have to change the file mode: chmod 755 most_used_port.py
4
5 from jsonrpclib import Server
6 import ssl
7 ssl.create_default_https_context = ssl._create_unverified_context
8 # Allows Python to use self-signed certs. In case no certificate is added to EOS
9 #
10 router_1 = Server("https://cvsadmin:Assw@192.168.1.199/command-api")
11 response = router_1.runCmds(1,["show interfaces"])
12
13 in_pkts_max = 0
14 interface_max = ""
15 #
16 for index in range(1,10):
17     print ()
18     interface_name = response[0][["interfaces"][["Ethernet"+str(index)]]["name"]]
19     print ("Interface Name: ",interface_name)
20     print ("Interface InBitsRate: ",response[0][["interfaces"][["Ethernet"+str(index)]]["interfaceStatistics"][["inBitsRate"]])
21     interface_inTotalPkts = response[0][["interfaces"][["Ethernet"+str(index)]]["interfaceCounters"][["inTotalPkts"]]
22     print ("Interface inTotalPkts: ",interface_inTotalPkts)
23     if interface_inTotalPkts > in_pkts_max:
24         interface_max = interface_name
25         in_pkts_max = interface_inTotalPkts
26 #
27 print ()
28 print (*****)
29 print ("most used interface: ",interface_max)
30 print ("max InTotalPkts: ",in_pkts_max)
31 #
32 response = router_1.runCmds(1,["enable","configure terminal","interface "+interface_max,"description most used interface"])
33
```

Interface Name	InBitsRate	InTotalPkts
Ethernet5	0.0	0
Ethernet6	0.0	0
Ethernet7	0.0	0
Ethernet8	0.0	0
Ethernet9	0.0	0

most used interface: Ethernet2
max InTotalPkts: 51497

Example Python Script:

This quite simple Python script can be run on any remote machine connected to the OOB Management Port of the Arista system and displays the Ethernet Ports including the assigned in-Bits-rate and in-total-Packets rates and derives the “most used Port”.

In this concrete example the **in-total-packets-rate** is used as within a Lab environment it’s sometimes not possible to generate a significant amount of data to use the in-Bits-rate - especially when using vEOS on GNS3.

Last but not least, the script will change the description on the Arista System to “most used interface”.

```
#!/usr/bin/env python3
# Shebang declares this script as executable
# You also will have to change the file mode: chmod 755 most_used_port.py
#
from jsonrpclib import Server
import ssl
ssl._create_default_https_context = ssl._create_unverified_context
# Allows Python to use self-signed certs in case no certificate is added to EOS
#
router_1 = Server("https://cvpadmin:pAssw0rd@192.168.1.199/command-api")
response = router_1.runCmds(1,["show interfaces"])
#
in_pkts_max = 0
interface_max = ""
#
for index in range (1,10):
    print ()
    interface_name = response[0]["interfaces"]["Ethernet"+str(index)]["name"]
    print ("Interface Name: ",interface_name)
    print ("Interface inBitsRate:
",response[0]["interfaces"]["Ethernet"+str(index)]["interfaceStatistics"]["inBitsRate"])
    interface_intotalpkts =
response[0]["interfaces"]["Ethernet"+str(index)]["interfaceCounters"]["inTotalPkts
"]
    print ("Interface inTotalPkts: ",interface_intotalpkts)
    if interface_intotalpkts > in_pkts_max:
        interface_max = interface_name
        in_pkts_max = interface_intotalpkts
#
print ()
print ("*****")
print ("most used Interface: ",interface_max)
print ("max inTotalPkts: ",in_pkts_max)
#
response = router_1.runCmds(1,["enable","configure terminal","interface
"+interface_max,"description most used interface"])
```

Output:

Interface Name: Ethernet1
Interface inBitsRate: 0.0
Interface inTotalPkts: 20296

Interface Name: Ethernet2
Interface inBitsRate: 0.0
Interface inTotalPkts: 51178

Interface Name: Ethernet3
Interface inBitsRate: 0.0
Interface inTotalPkts: 0

Interface Name: Ethernet4
Interface inBitsRate: 0.0
Interface inTotalPkts: 0

Interface Name: Ethernet5
Interface inBitsRate: 0.0
Interface inTotalPkts: 0

Interface Name: Ethernet6
Interface inBitsRate: 0.0
Interface inTotalPkts: 0

Interface Name: Ethernet7
Interface inBitsRate: 0.0
Interface inTotalPkts: 0

Interface Name: Ethernet8
Interface inBitsRate: 0.0
Interface inTotalPkts: 0

Interface Name: Ethernet9
Interface inBitsRate: 0.0
Interface inTotalPkts: 0

most used Interface: Ethernet2
max inTotalPkts: 51178

New Port description on the Arista system:

```
Router-1(config-if-Et2)#show interfaces ethernet 2 status
Port      Name              Status      Vlan      Duplex  Speed  Type
Flags Encapsulation
Et2       most used interface connected  routed   full   1G     EbraTestPhyPort
```

Executable Python script:

By using the Shebang “#!/usr/bin/env python3” within the Python script it is also possible to declare it as an executable code within Ubuntu but you also will have to **change the file mode**:

```
mkrausch@ubuntu:~$ dir
Ansible_Demo Desktop Documents Downloads Music Pictures Public Templates
Videos most_used_port.py
mkrausch@ubuntu:~$ chmod 755 most_used_port.py
mkrausch@ubuntu:~$ ./most_used_port.py
```

```
Interface Name: Ethernet1
Interface inBitsRate: 0.0
Interface inTotalPkts: 20296
```

```
Interface Name: Ethernet2
Interface inBitsRate: 0.0
Interface inTotalPkts: 51178
```

```
Interface Name: Ethernet3
Interface inBitsRate: 0.0
Interface inTotalPkts: 0
```

```
Interface Name: Ethernet4
Interface inBitsRate: 0.0
Interface inTotalPkts: 0
```

```
Interface Name: Ethernet5
Interface inBitsRate: 0.0
Interface inTotalPkts: 0
```

```
Interface Name: Ethernet6
Interface inBitsRate: 0.0
Interface inTotalPkts: 0
```

```
Interface Name: Ethernet7
Interface inBitsRate: 0.0
Interface inTotalPkts: 0
```

```
Interface Name: Ethernet8
Interface inBitsRate: 0.0
Interface inTotalPkts: 0
```

```
Interface Name: Ethernet9
Interface inBitsRate: 0.0
Interface inTotalPkts: 0
```

```
*****
most used Interface: Ethernet2
```

max inTotalPkts: 51178

Ansible Automation

Ansible has an **Agent-less architecture** so that you don't need to run a daemon on the clients/ EOS systems. Arista Networks and Ansible have partnered to bring the strength and agility of Ansible to your network.

eos_config is a **core module managed by Ansible** network team.

As this module is part of the core, there is no need to install additional Ansible module with **ansible-galaxy**.

Additional information can be found at:

<https://www.ansible.com/integrations/networks/arista>

Basic Components of Ansible:

Tower:

A commercial non free of charge **Web GUI** for Ansible

Control node:

Any machine with Ansible installed is a Control Node. You can run commands and playbooks, invoking `/usr/bin/ansible` or `/usr/bin/ansible-playbook`, from any control node. You can use any computer that has **Python** installed on it as a control.

You can have multiple control nodes but **you cannot use a Windows machine as a control node**.

Managed nodes:

The network devices (and/or servers) you manage with Ansible. Managed nodes are also sometimes called "hosts". **Ansible is not installed on managed nodes**.

Inventory:

A list of managed nodes. An inventory file is also sometimes called a "**hostfile**". Your inventory can specify information like IP address for each managed node. An inventory can also organize managed nodes, creating and nesting groups for easier scaling.

Modules:

The units of code Ansible executes. Each module has a particular use, from administering users on a specific type of database to managing VLAN interfaces on a specific type of network device. You can invoke a single module with a task, or invoke several different modules in a playbook.

Modules can be written in any language and communicate in standard **JSON**.

Tasks:

The units of action in Ansible. You can execute a single task once with an ad-hoc command.

Playbooks:

Ordered lists of tasks, saved so you can run those tasks in that order repeatedly. Playbooks can include **variables** as well as tasks. Playbooks are written in **YAML** and are easy to read, write, share and understand.

Installing Ansible on Ubuntu

In this example we are using **Ubuntu Server 18.04.3**:

```
mkrausch@ubuntu18:~$ sudo su
[sudo] password for mkrausch:
root@ubuntu18:/home/mkrausch# apt-add-repository ppa:ansible/ansible
  Ansible is a radically simple IT automation platform that makes your applications
  and systems easier to deploy. Avoid writing scripts or custom code to deploy and
  update your applications— automate in a language that approaches plain English,
  using SSH, with no agents to install on remote systems.
```

```
http://ansible.com/
```

```
  More info: https://launchpad.net/~ansible/+archive/ubuntu/ansible
```

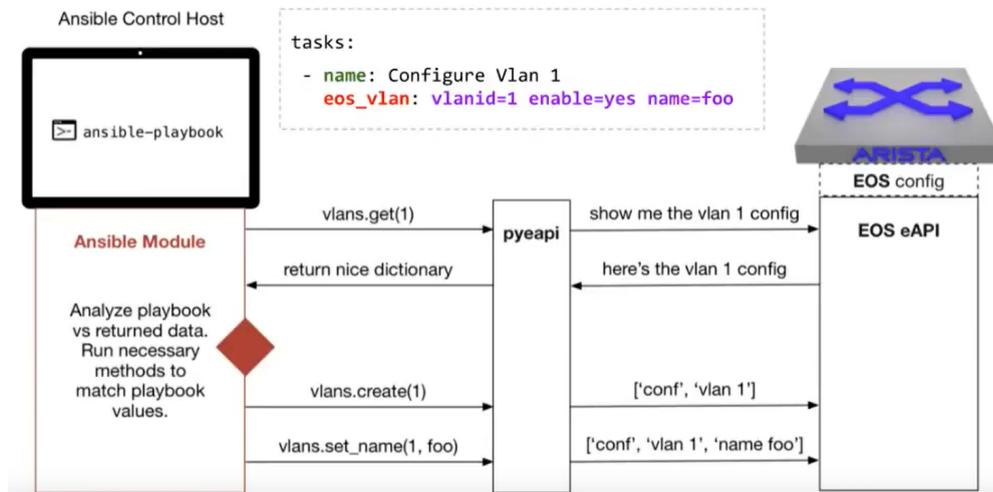
```
  Press [ENTER] to continue or Ctrl-c to cancel adding it.
```

```
Hit:1 http://security.ubuntu.com/ubuntu bionic-security InRelease
Get:2 http://ppa.launchpad.net/ansible/ansible/ubuntu bionic InRelease [15.9 kB]
Hit:3 http://de.archive.ubuntu.com/ubuntu bionic InRelease
Hit:4 http://de.archive.ubuntu.com/ubuntu bionic-updates InRelease
Hit:5 http://de.archive.ubuntu.com/ubuntu bionic-backports InRelease
Get:6 http://ppa.launchpad.net/ansible/ansible/ubuntu bionic/main i386 Packages
[688 B]
Get:7 http://ppa.launchpad.net/ansible/ansible/ubuntu bionic/main amd64 Packages
[688 B]
Get:8 http://ppa.launchpad.net/ansible/ansible/ubuntu bionic/main Translation-en
[472 B]
Fetched 17.7 kB in 1s (28.2 kB/s)
Reading package lists... Done
root@ubuntu18:/home/mkrausch# apt update
Hit:1 http://security.ubuntu.com/ubuntu bionic-security InRelease
Hit:2 http://de.archive.ubuntu.com/ubuntu bionic InRelease
Hit:3 http://ppa.launchpad.net/ansible/ansible/ubuntu bionic InRelease
Hit:4 http://de.archive.ubuntu.com/ubuntu bionic-updates InRelease
Hit:5 http://de.archive.ubuntu.com/ubuntu bionic-backports InRelease
Reading package lists... Done
Building dependency tree
Reading state information... Done
86 packages can be upgraded. Run 'apt list --upgradable' to see them.
root@ubuntu18:/home/mkrausch# apt install ansible
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following packages were automatically installed and are no longer required:
  libatkmm-1.6-1v5 libcairomm-1.0-1v5 libgtkmm-3.0-1v5 libnih-dbus1
  libpangomm-1.4-1v5
Use 'sudo apt autoremove' to remove them.
The following additional packages will be installed:
  python-httpplib2 python-paramiko python-pyasn1 python-yaml sshpass
Suggested packages:
  python-gssapi
The following NEW packages will be installed:
  ansible python-httpplib2 python-paramiko python-pyasn1 python-yaml sshpass
0 upgraded, 6 newly installed, 0 to remove and 86 not upgraded.
Need to get 6,093 kB of archives.
After this operation, 59.6 MB of additional disk space will be used.
Do you want to continue? [Y/n] y
```

```
Get:1 http://ppa.launchpad.net/ansible/ansible/ubuntu bionic/main amd64 ansible
all 2.9.1-1ppa~bionic [5,775 kB]
Get:2 http://de.archive.ubuntu.com/ubuntu bionic/main amd64 python-yaml amd64
3.12-1build2 [115 kB]
Get:3 http://de.archive.ubuntu.com/ubuntu bionic/main amd64 python-pyasn1 all
0.4.2-3 [46.7 kB]
Get:4 http://de.archive.ubuntu.com/ubuntu bionic-updates/main amd64
python-paramiko all 2.0.0-1ubuntu1.2 [110 kB]
Get:5 http://de.archive.ubuntu.com/ubuntu bionic-updates/main amd64
python-httpplib2 all 0.9.2+dfsg-1ubuntu0.1 [34.7 kB]
Get:6 http://de.archive.ubuntu.com/ubuntu bionic/universe amd64 sshpass amd64
1.06-1 [10.5 kB]
Fetched 6,093 kB in 1s (8,885 kB/s)
Selecting previously unselected package python-yaml.
(Reading database ... 166044 files and directories currently installed.)
Preparing to unpack .../0-python-yaml_3.12-1build2_amd64.deb ...
Unpacking python-yaml (3.12-1build2) ...
Selecting previously unselected package python-pyasn1.
Preparing to unpack .../1-python-pyasn1_0.4.2-3_all.deb ...
Unpacking python-pyasn1 (0.4.2-3) ...
Selecting previously unselected package python-paramiko.
Preparing to unpack .../2-python-paramiko_2.0.0-1ubuntu1.2_all.deb ...
Unpacking python-paramiko (2.0.0-1ubuntu1.2) ...
Selecting previously unselected package python-httpplib2.
Preparing to unpack .../3-python-httpplib2_0.9.2+dfsg-1ubuntu0.1_all.deb ...
Unpacking python-httpplib2 (0.9.2+dfsg-1ubuntu0.1) ...
Selecting previously unselected package sshpass.
Preparing to unpack .../4-sshpass_1.06-1_amd64.deb ...
Unpacking sshpass (1.06-1) ...
Selecting previously unselected package ansible.
Preparing to unpack .../5-ansible_2.9.1-1ppa~bionic_all.deb ...
Unpacking ansible (2.9.1-1ppa~bionic) ...
Setting up python-yaml (3.12-1build2) ...
Setting up python-pyasn1 (0.4.2-3) ...
Setting up python-httpplib2 (0.9.2+dfsg-1ubuntu0.1) ...
Setting up python-paramiko (2.0.0-1ubuntu1.2) ...
Setting up sshpass (1.06-1) ...
Setting up ansible (2.9.1-1ppa~bionic) ...
Processing triggers for man-db (2.8.3-2ubuntu0.1) ...
root@ubuntu18:/home/mkrausch#
```

Installing Python Client for eAPI (pyeapi)

Pyeapi is an abstraction of the EOS system CLI and converts JSON objects from Ansible into eAPI and vice versa.



When using SSH to connect to the EOS system you would have to install `pyeapi` on the EOS systems. It is much easier connecting via **eAPI (http/ https)** to the EOS systems because you can **install pyeapi directly on the Ubuntu Server**.

```
root@ubuntu18:/home/mkrausch# pip install pyeapi
Collecting pyeapi
  Downloading
    https://files.pythonhosted.org/packages/bb/03/58f8bc4ba77ee33fd762fa50915a20304179
    fa2613827318baf2a08a327e/pyeapi-0.8.2.tar.gz (133kB)
    100% |#####| 143kB 3.3MB/s
Collecting netaddr (from pyeapi)
  Downloading
    https://files.pythonhosted.org/packages/ba/97/ce14451a9fd7bdb5a397abf99b24a1a6bb7a
    1a440b019bebd2e9a0dbec74/netaddr-0.7.19-py2.py3-none-any.whl (1.6MB)
    100% |#####| 1.6MB 642kB/s
Building wheels for collected packages: pyeapi
  Running setup.py bdist_wheel for pyeapi ... done
  Stored in directory:
    /root/.cache/pip/wheels/c2/c8/fc/a09c4c224b67653243990f59efb8d5e3722ca84e86dcd6526
    8
Successfully built pyeapi
Installing collected packages: netaddr, pyeapi
Successfully installed netaddr-0.7.19 pyeapi-0.8.2
root@ubuntu18:/home/mkrausch#
```

Setting up EOS systems

You have to **enable eAPI** access on the EOS systems. In this case we will use secure **HTTPS** access.

Running-Config:

```
hostname switch-1
!
spanning-tree mode mstp
!
no aaa root
!
username admin privilege 15 role network-admin secret arista
!
vrf instance MGMT
!
interface Management1
    vrf MGMT
    ip address 192.168.10.71/24
    no lldp receive
!
ip route vrf MGMT 0.0.0.0/0 192.168.10.1
!
management api http-commands
    no shutdown
!
    vrf MGMT
        no shutdown
!
```

Setting up an Inventory File

The **inventory file** contains information about the hosts to be managed via Ansible.

Hosts can be organized into groups and subgroups.

The inventory file is also often used to set **variables** that will be valid only for specific hosts or groups, in order to be used within playbooks and templates.

```
root@ubuntu18:/home/mkrausch# nano /etc/ansible/hosts
```

```
*****
```

```
GNU nano 2.9.3  
/etc/ansible/hosts
```

```
# This is the default ansible 'hosts' file.  
#  
# It should live in /etc/ansible/hosts  
#  
# - Comments begin with the '#' character  
# - Blank lines are ignored  
# - Groups of hosts are delimited by [header] elements  
# - You can enter hostnames or ip addresses  
# - A hostname/ip can be a member of multiple groups  
  
# Ex 1: Ungrouped hosts, specify before any group headers.  
  
## green.example.com  
## blue.example.com  
## 192.168.100.1  
## 192.168.100.10  
  
[lab]  
192.168.10.71  
192.168.10.72  
192.168.10.73  
192.168.10.74  
  
[lab:vars]  
ansible_network_os=eos  
ansible_httpapi_use_ssl=true  
ansible_httpapi_validate_certs=no  
ansible_connection=httpapi  
ansible_user='admin'  
ansible_password='arista'  
  
# Ex 2: A collection of hosts belonging to the 'webservers' group  
  
## [webservers]  
## alpha.example.org  
## beta.example.org  
## 192.168.1.100  
## 192.168.1.110  
  
# If you have multiple hosts following a pattern you can specify  
# them like this:
```

```

## www[001:006].example.com

# Ex 3: A collection of database servers in the 'dbservers' group

## [dbservers]
##
## db01.intranet.mydomain.net
## db02.intranet.mydomain.net
## 10.25.1.56
## 10.25.1.57

# Here's another example of host ranges, this time there are no
# leading 0s:

## db-[99:101]-node.example.com

```

```
*****
```

```

root@ubuntu18:/home/mkrausch# ansible-inventory --list -y
all:
  children:
    lab:
      hosts:
        192.168.10.71:
          ansible_connection: httpapi
          ansible_httpapi_use_ssl: 'true'
          ansible_httpapi_validate_certs: 'no'
          ansible_network_os: eos
          ansible_password: arista
          ansible_user: admin
        192.168.10.72:
          ansible_connection: httpapi
          ansible_httpapi_use_ssl: 'true'
          ansible_httpapi_validate_certs: 'no'
          ansible_network_os: eos
          ansible_password: arista
          ansible_user: admin
        192.168.10.73:
          ansible_connection: httpapi
          ansible_httpapi_use_ssl: 'true'
          ansible_httpapi_validate_certs: 'no'
          ansible_network_os: eos
          ansible_password: arista
          ansible_user: admin
        192.168.10.74:
          ansible_connection: httpapi
          ansible_httpapi_use_ssl: 'true'
          ansible_httpapi_validate_certs: 'no'
          ansible_network_os: eos
          ansible_password: arista
          ansible_user: admin
      ungrouped: {}

```

Checking connectivity to EOS devices via Ansible

After the inventory list has been configured you should test basic connectivity to the listed EOS devices:

```
root@ubuntu18:/home/mkrausch# ansible all -m ping
[DEPRECATION WARNING]: Distribution Ubuntu 18.04 on host 192.168.10.71 should use /usr/bin/python3, but is using /usr/bin/python for backward compatibility with prior Ansible releases. A future Ansible release will default to using the discovered platform python for this host. See https://docs.ansible.com/ansible/2.9/reference_appendices/interpreter_discovery.html for more information. This feature will be removed in version 2.12. Deprecation warnings can be disabled by setting deprecation_warnings=False in ansible.cfg.
192.168.10.71 | SUCCESS => {
  "ansible_facts": {
    "discovered_interpreter_python": "/usr/bin/python"
  },
  "changed": false,
  "ping": "pong"
}
[DEPRECATION WARNING]: Distribution Ubuntu 18.04 on host 192.168.10.73 should use /usr/bin/python3, but is using /usr/bin/python for backward compatibility with prior Ansible releases. A future Ansible release will default to using the discovered platform python for this host. See https://docs.ansible.com/ansible/2.9/reference_appendices/interpreter_discovery.html for more information. This feature will be removed in version 2.12. Deprecation warnings can be disabled by setting deprecation_warnings=False in ansible.cfg.
192.168.10.73 | SUCCESS => {
  "ansible_facts": {
    "discovered_interpreter_python": "/usr/bin/python"
  },
  "changed": false,
  "ping": "pong"
}
[DEPRECATION WARNING]: Distribution Ubuntu 18.04 on host 192.168.10.74 should use /usr/bin/python3, but is using /usr/bin/python for backward compatibility with prior Ansible releases. A future Ansible release will default to using the discovered platform python for this host. See https://docs.ansible.com/ansible/2.9/reference_appendices/interpreter_discovery.html for more information. This feature will be removed in version 2.12. Deprecation warnings can be disabled by setting deprecation_warnings=False in ansible.cfg.
192.168.10.74 | SUCCESS => {
  "ansible_facts": {
    "discovered_interpreter_python": "/usr/bin/python"
  },
  "changed": false,
  "ping": "pong"
}
[DEPRECATION WARNING]: Distribution Ubuntu 18.04 on host 192.168.10.72 should use /usr/bin/python3, but is using /usr/bin/python for backward compatibility with prior Ansible releases. A
```

future Ansible release will default to using the discovered platform python for this host. See https://docs.ansible.com/ansible/2.9/reference_appendices/interpreter_discovery.html for more information. This feature will be removed in version 2.12. Deprecation warnings can be disabled by setting `deprecation_warnings=False` in `ansible.cfg`.

```
192.168.10.72 | SUCCESS => {
  "ansible_facts": {
    "discovered_interpreter_python": "/usr/bin/python"
  },
  "changed": false,
  "ping": "pong"
}
root@ubuntu18:/home/mkrausch#
```

Sending CLI commands to EOS devices via Ansible

After the inventory list is configured and the basic connectivity has been tested you can easily send **CLI commands** directly to all listed EOS devices:

```
root@ubuntu18:/home/mkrausch# ansible lab -m eos_command -a "commands='show vlan
brief'"
[DEPRECATION WARNING]: Distribution Ubuntu 18.04 on host 192.168.10.74 should use
/usr/bin/python3, but is using /usr/bin/python for backward compatibility with
prior Ansible releases. A
future Ansible release will default to using the discovered platform python for
this host. See
https://docs.ansible.com/ansible/2.9/reference_appendices/interpreter_discovery.ht
ml for more
information. This feature will be removed in version 2.12. Deprecation warnings
can be disabled by setting deprecation_warnings=False in ansible.cfg.
192.168.10.74 | SUCCESS => {
  "ansible_facts": {
    "discovered_interpreter_python": "/usr/bin/python"
  },
  "changed": false,
  "stdout": [
    "VLAN  Name                               Status  Ports\n-----\n1
-----\n1
default                                active  Et1, Et2, Et5, Et6, Et7, Et8\n
Et9, Et10, Et11\n10  client-VLAN-10          active  Et12, Po10\n25
VLAN-25                                active"
  ],
  "stdout_lines": [
    [
      "VLAN  Name                               Status  Ports",
      "-----",
      "1      default                                active  Et1, Et2, Et5, Et6,
Et7, Et8",
      "10     client-VLAN-10          active  Et9, Et10, Et11",
      "25     VLAN-25                                active  Et12, Po10",
      "25     VLAN-25                                active"
    ]
  ]
}
[DEPRECATION WARNING]: Distribution Ubuntu 18.04 on host 192.168.10.73 should use
/usr/bin/python3, but is using /usr/bin/python for backward compatibility with
prior Ansible releases. A
future Ansible release will default to using the discovered platform python for
this host. See
https://docs.ansible.com/ansible/2.9/reference_appendices/interpreter_discovery.ht
ml for more
information. This feature will be removed in version 2.12. Deprecation warnings
can be disabled by setting deprecation_warnings=False in ansible.cfg.
192.168.10.73 | SUCCESS => {
  "ansible_facts": {
    "discovered_interpreter_python": "/usr/bin/python"
  },
  "changed": false,
  "stdout": [
    "VLAN  Name                               Status  Ports\n-----
```

```

-----\n1
default          active  Et1, Et2, Et5, Et6, Et7, Et8\n
Et9, Et10, Et11\n10  client-VLAN-10          active  Et12, Po2\n25
VLAN-25          active"
  ],
  "stdout_lines": [
    [
      "VLAN  Name                               Status  Ports",
      "-----",
      "1      default                               active  Et1, Et2, Et5, Et6,
Et7, Et8",
      "10     client-VLAN-10                          active  Et9, Et10, Et11",
      "25     VLAN-25                                   active  Et12, Po2",
    ]
  ]
}
[DEPRECATION WARNING]: Distribution Ubuntu 18.04 on host 192.168.10.72 should use
/usr/bin/python3, but is using /usr/bin/python for backward compatibility with
prior Ansible releases. A
future Ansible release will default to using the discovered platform python for
this host. See
https://docs.ansible.com/ansible/2.9/reference\_appendices/interpreter\_discovery.ht
ml for more
information. This feature will be removed in version 2.12. Deprecation warnings
can be disabled by setting deprecation_warnings=False in ansible.cfg.
192.168.10.72 | SUCCESS => {
  "ansible_facts": {
    "discovered_interpreter_python": "/usr/bin/python"
  },
  "changed": false,
  "stdout": [
    "VLAN  Name                               Status  Ports\n-----\n1
default          active  Et5, Et6, Et7, Et8, Et9, Et10\n
Et11, Et12, PEt5, PEt6, PEt7\n
PEt8, PEt9, PEt10, PEt11, PEt12\n
Po1000\n10  client-VLAN-10          active  Po1, Po2, Po1000\n25
VLAN-25          active  Po1000\n4094  MLAG-Interlink-VLAN
active  Cpu, Po1000"
  ],
  "stdout_lines": [
    [
      "VLAN  Name                               Status  Ports",
      "-----",
      "1      default                               active  Et5, Et6, Et7, Et8,
Et9, Et10",
      "10     client-VLAN-10                          active  Et11, Et12, PEt5,
PEt6, PEt7",
      "25     VLAN-25                                   active  PEt8, PEt9, PEt10,
Po1000",
      "4094  MLAG-Interlink-VLAN                      active  Po1, Po2, Po1000",
      "4094  MLAG-Interlink-VLAN                      active  Po1000",
      "4094  MLAG-Interlink-VLAN                      active  Cpu, Po1000"
    ]
  ]
}

```

```

    ]
}
[DEPRECATION WARNING]: Distribution Ubuntu 18.04 on host 192.168.10.71 should use
/usr/bin/python3, but is using /usr/bin/python for backward compatibility with
prior Ansible releases. A
future Ansible release will default to using the discovered platform python for
this host. See
https://docs.ansible.com/ansible/2.9/reference_appendices/interpreter_discovery.ht
ml for more
information. This feature will be removed in version 2.12. Deprecation warnings
can be disabled by setting deprecation_warnings=False in ansible.cfg.
192.168.10.71 | SUCCESS => {
  "ansible_facts": {
    "discovered_interpreter_python": "/usr/bin/python"
  },
  "changed": false,
  "stdout": [
    "VLAN  Name                               Status  Ports\n-----\n1
default                                     active  Et5, Et6, Et7, Et8, Et9, Et10\n
Et11, Et12, PEt5, PEt6, PEt7\n
PEt8, PEt9, PEt10, PEt11, PEt12\n
Po1000\n10  client-VLAN-10                        active  Po1, Po2, Po1000\n25
VLAN-25                                     active  Po1000\n4094 MLAG-Interlink-VLAN
active  Cpu, Po1000"
  ],
  "stdout_lines": [
    [
      "VLAN  Name                               Status  Ports",
      "-----",
      "1      default                             active  Et5, Et6, Et7, Et8,
Et9, Et10",
      "                                           Et11, Et12, PEt5,
PEt6, PEt7",
      "                                           PEt8, PEt9, PEt10,
PEt11, PEt12",
      "                                           Po1000",
      "10     client-VLAN-10                       active  Po1, Po2, Po1000",
      "25     VLAN-25                               active  Po1000",
      "4094   MLAG-Interlink-VLAN                 active  Cpu, Po1000"
    ]
  ]
}
root@ubuntu18:/home/mkrausch#

```

Ansible Playbooks

Following are some simple Ansible Playbooks and their outputs.

You may find additional sophisticated Arista EOS Ansible Playbooks on the internet!

Playbook "pb.collect.facts.yaml":

```
---
- name: Run commands on remote LAB devices
  hosts: lab
  connection: network_cli
  gather_facts: false

  tasks:
    - name: Collect all facts from device
      eos_facts:
        gather_subset:
          - all
      register: facts

    - name: Display result
      debug:
        msg: "Model is {{facts.ansible_facts.ansible_net_model}} and it is running
        {{facts.ansible_facts.ansible_net_version}}"
```

Output:

```
root@ubuntu18:/home/mkrausch# ansible-playbook pb.collect.facts.yaml

PLAY [Run commands on remote LAB devices]
*****
*****

TASK [Collect all facts from device]
*****
*****
[WARNING]: default value for `gather_subset` will be changed to `min` from
`!config` v2.11 onwards

[DEPRECATION WARNING]: Distribution Ubuntu 18.04 on host 192.168.10.72 should use
/usr/bin/python3, but is using /usr/bin/python for backward compatibility with
prior Ansible releases. A
future Ansible release will default to using the discovered platform python for
this host. See
https://docs.ansible.com/ansible/2.9/reference\_appendices/interpreter\_discovery.ht
ml for more
information. This feature will be removed in version 2.12. Deprecation warnings
can be disabled by setting deprecation_warnings=False in ansible.cfg.
ok: [192.168.10.72]
[DEPRECATION WARNING]: Distribution Ubuntu 18.04 on host 192.168.10.74 should use
/usr/bin/python3, but is using /usr/bin/python for backward compatibility with
prior Ansible releases. A
future Ansible release will default to using the discovered platform python for
this host. See
https://docs.ansible.com/ansible/2.9/reference\_appendices/interpreter\_discovery.ht
```

```

ml for more
information. This feature will be removed in version 2.12. Deprecation warnings
can be disabled by setting deprecation_warnings=False in ansible.cfg.
ok: [192.168.10.74]
[DEPRECATION WARNING]: Distribution Ubuntu 18.04 on host 192.168.10.71 should use
/usr/bin/python3, but is using /usr/bin/python for backward compatibility with
prior Ansible releases. A
future Ansible release will default to using the discovered platform python for
this host. See
https://docs.ansible.com/ansible/2.9/reference_appendices/interpreter_discovery.ht
ml for more
information. This feature will be removed in version 2.12. Deprecation warnings
can be disabled by setting deprecation_warnings=False in ansible.cfg.
ok: [192.168.10.71]
[DEPRECATION WARNING]: Distribution Ubuntu 18.04 on host 192.168.10.73 should use
/usr/bin/python3, but is using /usr/bin/python for backward compatibility with
prior Ansible releases. A
future Ansible release will default to using the discovered platform python for
this host. See
https://docs.ansible.com/ansible/2.9/reference_appendices/interpreter_discovery.ht
ml for more
information. This feature will be removed in version 2.12. Deprecation warnings
can be disabled by setting deprecation_warnings=False in ansible.cfg.
ok: [192.168.10.73]

```

TASK [Display result]

```

*****
*****
*****

```

```

ok: [192.168.10.73] => {
  "msg": "Model is vEOS and it is running 4.22.2F"
}
ok: [192.168.10.74] => {
  "msg": "Model is vEOS and it is running 4.22.2F"
}
ok: [192.168.10.72] => {
  "msg": "Model is vEOS and it is running 4.22.2F"
}
ok: [192.168.10.71] => {
  "msg": "Model is vEOS and it is running 4.22.2F"
}

```

PLAY RECAP

```

*****
*****
*****

```

```

192.168.10.71      : ok=2    changed=0    unreachable=0    failed=0
skipped=0    rescued=0    ignored=0
192.168.10.72      : ok=2    changed=0    unreachable=0    failed=0
skipped=0    rescued=0    ignored=0
192.168.10.73      : ok=2    changed=0    unreachable=0    failed=0
skipped=0    rescued=0    ignored=0
192.168.10.74      : ok=2    changed=0    unreachable=0    failed=0
skipped=0    rescued=0    ignored=0

```

root@ubuntu18:/home/mkrausch#

Playbook "pb.config.lines.simple.yaml":

This Playbook changes the **hostname** and the **description of interface Ethernet 1** and adds an **NTP Server**.

Please make sure to set **"become: yes"** to enable privilege mode!

```
---
- name: Run commands on remote LAB devices
  hosts: all
  connection: network_cli
  become: yes
  gather_facts: false

  tasks:
    - name: Configure device hostname from lines
      eos_config:
        lines:
          - "hostname {{inventory_hostname}}-Ansible"
          - "ntp server 216.239.35.8"

    - name: Configure interface description
      eos_config:
        lines:
          - description Automated by Ansible
        parents: interface Ethernet 1
        replace: block
```

Output:

```
root@ubuntu18:/home/mkrausch# ansible-playbook pb.config.lines.simple.yaml
```

```
PLAY [Run commands on remote LAB devices]
*****
*****

TASK [Configure device hostname from lines]
*****
*****

[DEPRECATION WARNING]: Distribution Ubuntu 18.04 on host 192.168.10.74 should use
/usr/bin/python3, but is using /usr/bin/python for backward compatibility with
prior Ansible releases. A
future Ansible release will default to using the discovered platform python for
this host. See
https://docs.ansible.com/ansible/2.9/reference_appendices/interpreter_discovery.ht
ml for more
information. This feature will be removed in version 2.12. Deprecation warnings
can be disabled by setting deprecation_warnings=False in ansible.cfg.
changed: [192.168.10.74]
[DEPRECATION WARNING]: Distribution Ubuntu 18.04 on host 192.168.10.73 should use
/usr/bin/python3, but is using /usr/bin/python for backward compatibility with
prior Ansible releases. A
future Ansible release will default to using the discovered platform python for
this host. See
https://docs.ansible.com/ansible/2.9/reference_appendices/interpreter_discovery.ht
```

```

ml for more
information. This feature will be removed in version 2.12. Deprecation warnings
can be disabled by setting deprecation_warnings=False in ansible.cfg.
changed: [192.168.10.73]
[DEPRECATION WARNING]: Distribution Ubuntu 18.04 on host 192.168.10.71 should use
/usr/bin/python3, but is using /usr/bin/python for backward compatibility with
prior Ansible releases. A
future Ansible release will default to using the discovered platform python for
this host. See
https://docs.ansible.com/ansible/2.9/reference_appendices/interpreter_discovery.ht
ml for more
information. This feature will be removed in version 2.12. Deprecation warnings
can be disabled by setting deprecation_warnings=False in ansible.cfg.
changed: [192.168.10.71]
[DEPRECATION WARNING]: Distribution Ubuntu 18.04 on host 192.168.10.72 should use
/usr/bin/python3, but is using /usr/bin/python for backward compatibility with
prior Ansible releases. A
future Ansible release will default to using the discovered platform python for
this host. See
https://docs.ansible.com/ansible/2.9/reference_appendices/interpreter_discovery.ht
ml for more
information. This feature will be removed in version 2.12. Deprecation warnings
can be disabled by setting deprecation_warnings=False in ansible.cfg.
changed: [192.168.10.72]

```

TASK [Configure interface description]

```

*****
*****
changed: [192.168.10.74]
changed: [192.168.10.72]
changed: [192.168.10.71]
changed: [192.168.10.73]

```

PLAY RECAP

```

*****
*****
*****
192.168.10.71      : ok=2    changed=2    unreachable=0    failed=0
skipped=0    rescued=0    ignored=0
192.168.10.72      : ok=2    changed=2    unreachable=0    failed=0
skipped=0    rescued=0    ignored=0
192.168.10.73      : ok=2    changed=2    unreachable=0    failed=0
skipped=0    rescued=0    ignored=0
192.168.10.74      : ok=2    changed=2    unreachable=0    failed=0
skipped=0    rescued=0    ignored=0

```

```

root@ubuntu18:/home/mkrausch#

```

The Hostnames and the description of Interface Ethernet 1 have successfully been changed and a new NTP Server has been added by Ansible on all EOS systems listed within the Inventory List:

```

192.168.10.71-ansible#show run
! Command: show running-config
! device: 192.168.10.71-ansible (vEOS, EOS-4.22.2F)
!
! boot system flash:/vEOS-lab.swi

```

```
...
!  
hostname 192.168.10.71-Ansible  
!  
ntp server 216.239.35.8  
!  
spanning-tree mode mstp  
...  
!  
interface Ethernet1  
    description Automated by Ansible  
    channel-group 1000 mode active  
...  
!  
end
```

CloudVision Management

CloudVision is a network management framework for EOS devices supporting **workload orchestration, workflow automation and telemetry**.

Additional information can be found at:

<https://www.arista.com/en/cg-cv/cv-introduction-to-cloudvision>

Default User credentials for **web access**:

Username: **cvpadmin**

Password: **cvpadmin**

Default User credentials for **SSH access**:

Username: **root**

Password: **set via Console during first setup of CVP**

CloudVision password recovery

If you might have lost your cvpadmin password you can login to CloudVision via SSH using the root account and reset the password:

```
mkrausch@mkrausch ~ % ssh root@10.83.30.160
root@10.83.30.160's password:
Last login: Wed Jan 20 09:24:13 2021 from 10.95.66.56
-bash: warning: setlocale: LC_CTYPE: cannot change locale (UTF-8): No such file or
directory
[root@mkrauschcvp ~]# cd /cvpi/tools/
[root@mkrauschcvp tools]# ./update-mgmt-password -password newpassword123
```

EOS setup for management access

To manage an EOS device via CloudVision you have to configure an **User/ Password on the systems which you are using when logging in to CloudVision**.

In addition, you will have to **enable API management** on the EOS systems and this has to be enabled on configured **VRFs** also if used!

```
username admin privilege 15 role network-admin secret arista
!
vrf instance MGMT
!
interface Management1
 vrf MGMT
 ip address 192.168.10.71/24
 no lldp receive
!
ip route vrf MGMT 0.0.0.0/0 192.168.10.1
!
management api http-commands
 no shutdown
```

```
!  
vrf MGMT  
  no shutdown
```

Adding a device to CloudVision

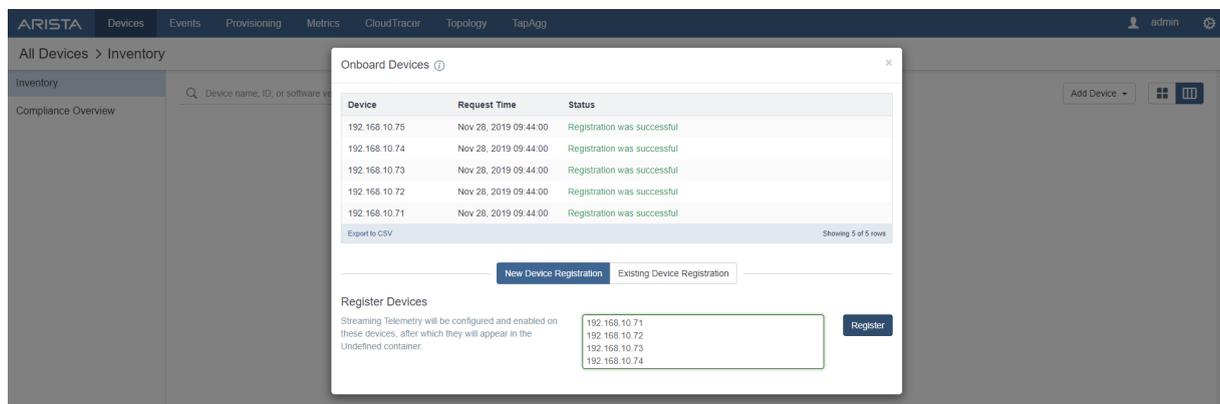
When not using ZTP, clicking on **“Add Device”** and typing in the correct IP address (here the OOB Port) followed by a click on **“Register”** will start the registration process.

Even if CloudVision has found new devices automatically you will have to register these devices so that they will be assigned to the **“undefined Container”** by default!

The following dynamic config will be pushed automatically to newly registered devices

daemon TerminAttr

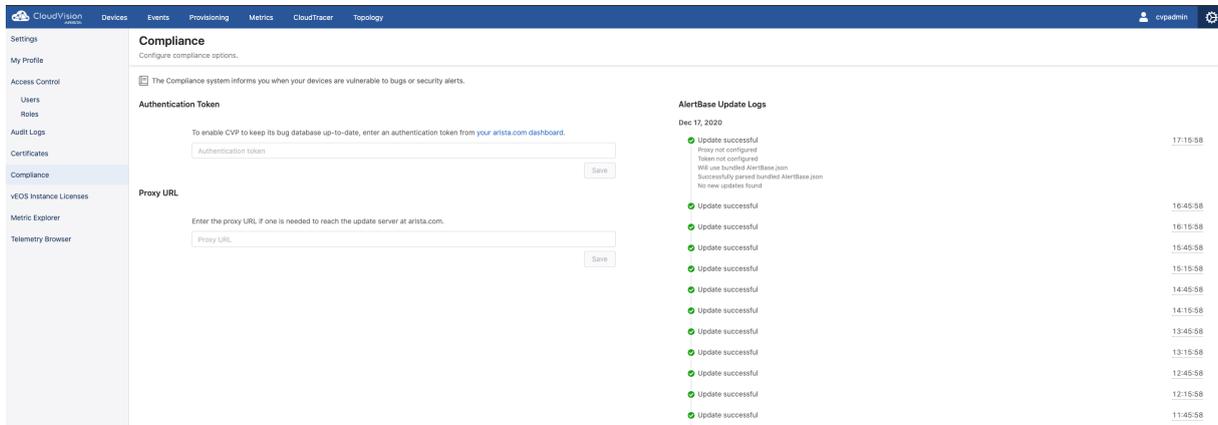
```
exec /usr/bin/TerminAttr -ingestgrpcurl=192.168.10.60:9910 -cvcompression=gzip
-taillogs -ingestauth=key,magickey
-smashexcludes=ale,flexCounter,hardware,kni,pulse,strata
-ingestexclude=/Sysdb/cell/1/agent,/Sysdb/cell/2/agent -ingestvrf=MGMT
no shutdown
```



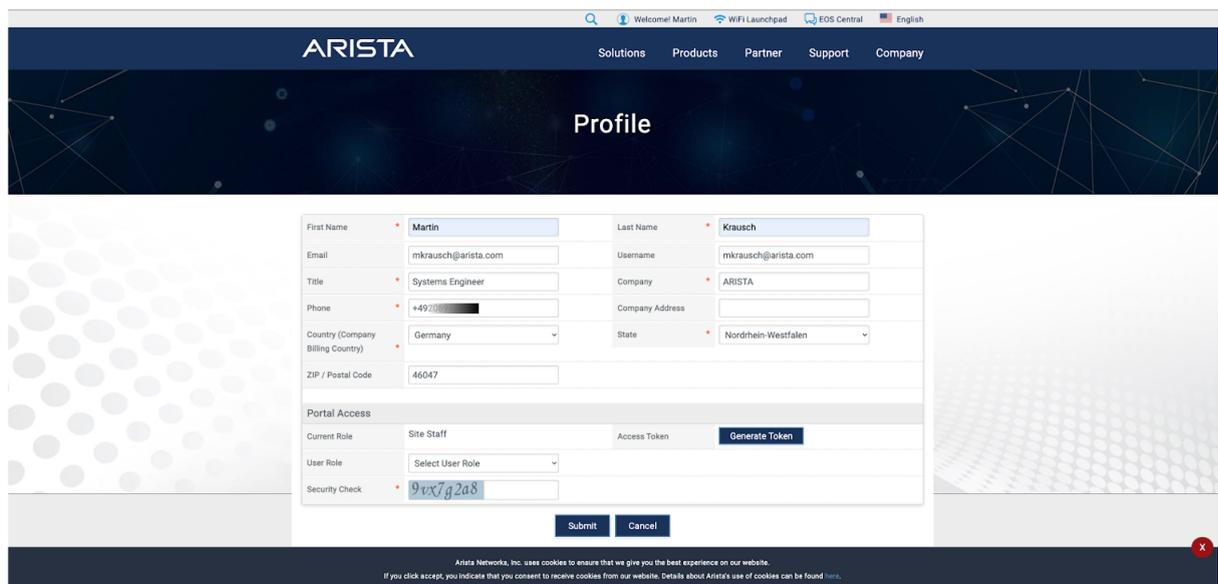
After a successful registration CloudVision will check automatically for any known bugs or vulnerability updates and will display an icon next to the affected systems:



In order to keep the bug database up to date and receive real-time assessments on exposure to software defects and security vulnerabilities, an **automated sync** can be configured between CloudVision and <https://www.arista.com> using a token-based authentication and a proxy URL. Select **Settings & Tools/ Compliance**:



By clicking on the link [“your arista.com dashboard.”](#) you will be redirected to [“https://www.arista.com/en/users/profile”](https://www.arista.com/en/users/profile) where you can generate a Token:



By clicking on **Generate Token** a Token will be created:

The screenshot shows the 'Profile' page in the Arista network management interface. The user's profile information is as follows:

First Name	Martin	Last Name	Krausch
Email	mkrausch@arista.com	Username	mkrausch@arista.com
Title	Systems Engineer	Company	ARISTA
Phone	+49 231 231 231 231	Company Address	
Country (Company Billing Country)	Germany	State	Nordrhein-Westfalen
ZIP / Postal Code	46047		

Under the 'Portal Access' section:

Current Role	Site Staff	Access Token	974a86057b4637506838383838383838
			Regenerate Token
User Role	Select User Role		
Security Check	9vx7g2a8		

Buttons: **Submit**, **Cancel**

Copy & Paste this Token into the CloudVision **Authentication Token Field** and click **Save**:

The screenshot shows the 'Compliance' page in the CloudVision interface. The 'Authentication Token' field contains the token: 974a86057b4637506838383838383838. A 'Save' button is visible next to the field.

The 'AlertBase Update Logs' section shows a list of successful updates:

Time	Status
17:15:58	Update successful
16:45:58	Update successful
16:15:58	Update successful
15:45:58	Update successful
15:15:58	Update successful
14:45:58	Update successful
14:15:58	Update successful
13:45:58	Update successful
13:15:58	Update successful
12:45:58	Update successful
12:15:58	Update successful
11:45:58	Update successful
11:15:58	Update successful

CloudVision automatically will update the **AlertBase.json** file containing the latest Bug information **every 30 minutes**:

The screenshot shows the 'Compliance' page in the CloudVision interface. The 'Authentication Token' field now has a 'Reveal token' button and a 'Delete' button next to it.

The 'AlertBase Update Logs' section shows a list of successful updates:

Time	Status
08:15:58	Update successful
07:45:58	Update successful
07:15:58	Update successful
06:45:58	Update successful
06:15:58	Update successful
05:45:58	Update successful
05:15:58	Update successful

By clicking on the icon you can see additional details:

Bug ID	Summary	Severity ↑	Version Introduced	Version(s) Fixed
400990	CVE-2019-14810	High	4.17.0	4.20.15, 4.21.2.4, 4.21.8, 4.22.0.2, 4.23.0

When Devices are added correctly you should see the LLDP based topology when selecting the Tab “Topology”:

Topology Overview

Displaying 4 managed and 0 other devices

Network Filters

Management network:

VLAN membership: ID or range (e.g. "1, 3-4")

Link Overlay: None

Devices

- switch-1 (A48DEA23582C3E75472243455A...)
- switch-2 (A3AEE902DR225FE643020FAAC...)
- switch-3 (7FA95403E68303594C8BD7A97...)
- switch-4 (6C58ED4M97E7F7E20254893B01A...)

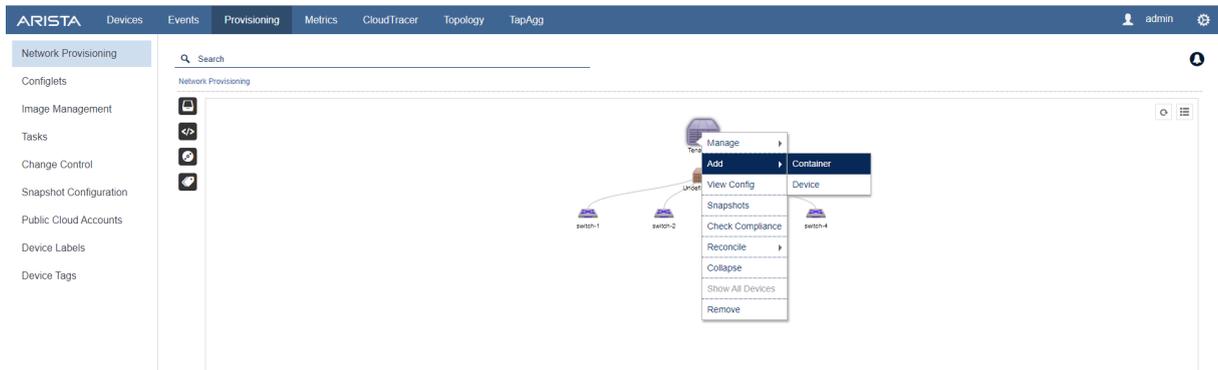
Within the Tab “Provisioning” the new devices will first be assigned by default to an “undefined Container”:

Network Provisioning

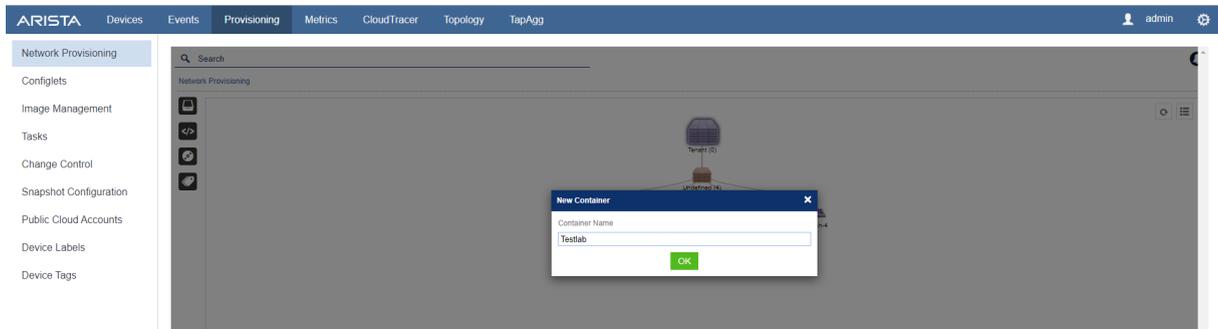
Undefined (0)

- switch-1
- switch-2
- switch-3
- switch-4

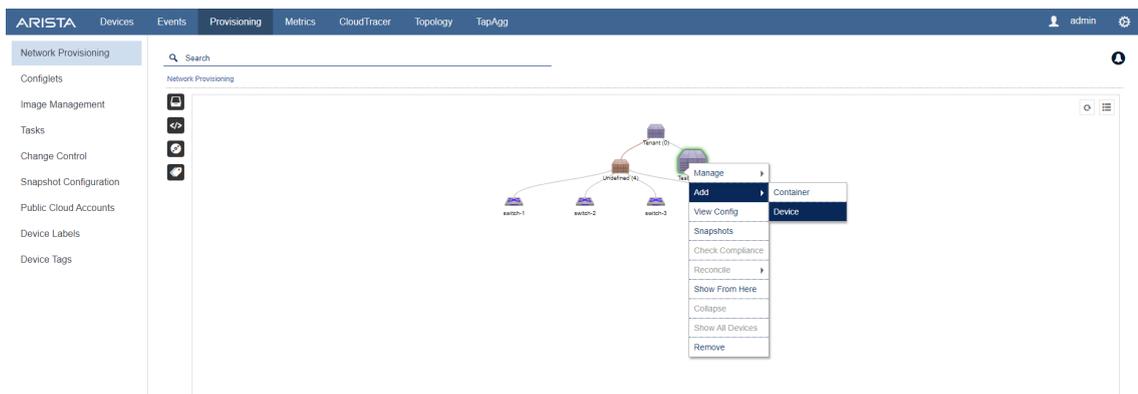
You can create a new Container (used for SW and Config changes) and place the devices into it. Right click on Tenant and then **Add -> Container**:



Assign a suitable name to the new Container (here: "Testlab") and press ok:



Right Click on the newly created Container and select **Add -> Device**:



Select all the devices you want to place into the new Container and click on the **Add** button:

The screenshot shows the ARISTA Network Provisioning interface. The 'Undefined Devices' table contains the following data:

Name	Serial No	IP Address	Mac Address	Model	Version
<input checked="" type="checkbox"/> switch-1	A48DEA23582C3E75472143455A487287	192.168.10.71	0c:34:13:15:50:a1		4.22.2F
<input checked="" type="checkbox"/> switch-2	A3AEE002D8225FE64302FAAC85A78B3	192.168.10.72	0c:34:13:56:43:c2		4.22.2F
<input checked="" type="checkbox"/> switch-3	7FA995403E8830594C8BD7A974EEAEF	192.168.10.73	0c:34:13:4e:c5:67		4.22.2F
<input checked="" type="checkbox"/> switch-4	6C58EDA497E72E0254893801A5FDFR2	192.168.10.74	0c:34:13:d1:52:73		4.22.2F

At the bottom of the table, there are 'Add' and 'Back' buttons.

Click **Save**:

The screenshot shows the ARISTA Network Provisioning interface with a network topology diagram. The diagram features a central 'Tenant (0)' node connected to two 'Undefined (0)' nodes, which are each connected to four 'switch' nodes. At the bottom of the interface, there are 'Preview', 'Save', and 'Cancel' buttons.

Click on **“Pending Tasks”**:

The screenshot shows the ARISTA Network Provisioning interface with a network topology diagram. The diagram features a central 'Tenant (0)' node connected to two 'Undefined (4)' nodes, which are each connected to four 'switch' nodes. In the top right corner, there is a 'Pending Tasks' notification icon.

Select the devices which should be moved to the new Container and click on **“Create Change Control with x Tasks”**.

This is to prevent unwanted movements of systems within large Data Centers without double accepting on executing these tasks!

The screenshot shows the ARISTA Tasks page. The left sidebar contains navigation options: Network Provisioning, Configlets, Image Management, **Tasks**, Change Control, Snapshot Configuration, Public Cloud Accounts, Device Tags, and Tag Management. The main content area is titled 'Tasks' and includes a sub-header 'View tasks and assign tasks to new change controls.' Below this are two buttons: '+ Create Change Control with 4 Tasks' and 'Cancel 4 Tasks'. The 'Assignable Tasks' table lists four tasks:

Task ID	Device	Created By	Type	Last Updated ↓	Status
47	switch-3 MAC: 0c:34:13:4e:c5:b7 IP: 192.168.10.73	admin	Add Device	a minute ago	● Pending
46	switch-4 MAC: 0c:34:13:d1:52:73 IP: 192.168.10.74	admin	Add Device	a minute ago	● Pending
45	switch-1 MAC: 0c:34:13:19:50:a1 IP: 192.168.10.71	admin	Add Device	a minute ago	● Pending
44	switch-2 MAC: 0c:34:13:5d:43:c2 IP: 192.168.10.72	admin	Add Device	a minute ago	● Pending

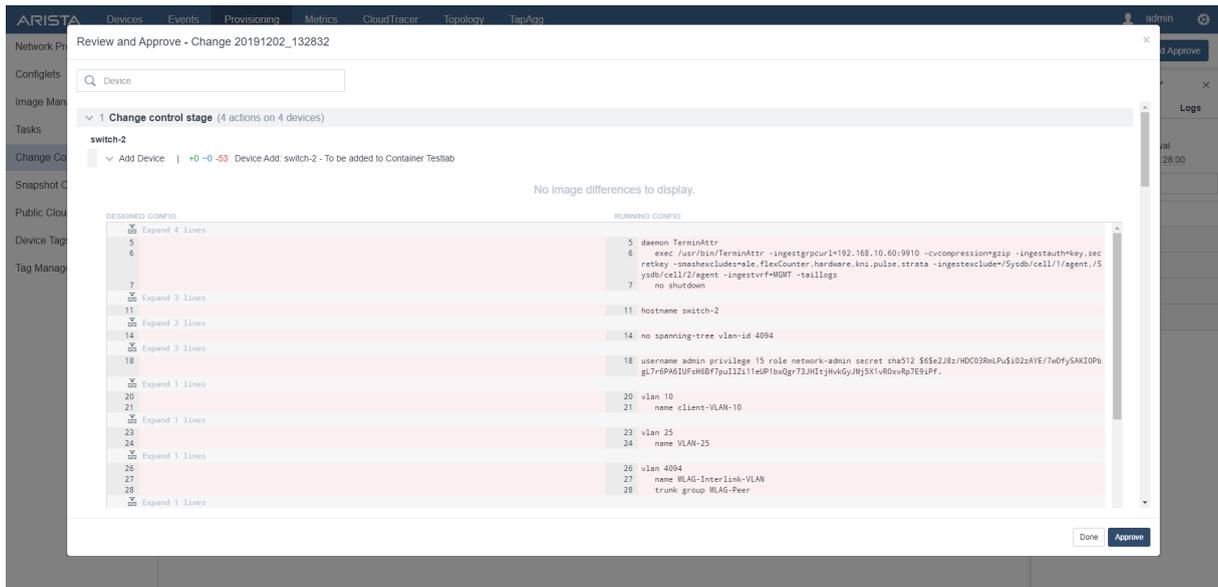
Below the assignable tasks is the 'Assigned Tasks' table:

Task ID	Device	Created By	Type	Last Updated ↓	Status	Change Control
43	Leaf-2 MAC: 0c:7e:d1:85:c5:b6 IP: 192.168.10.74	admin	Update Config	7 days ago	● Completed	Change 20191125_145528
42	Leaf-1 MAC: 0c:7e:d1:c1:92:48 IP: 192.168.10.73	admin	Update Config	7 days ago	● Completed	Change 20191125_145414
41	Leaf-2 MAC: 0c:7e:d1:85:c5:b6 IP: 192.168.10.74	admin	Update Config	7 days ago	● Completed	Change 20191125_140035

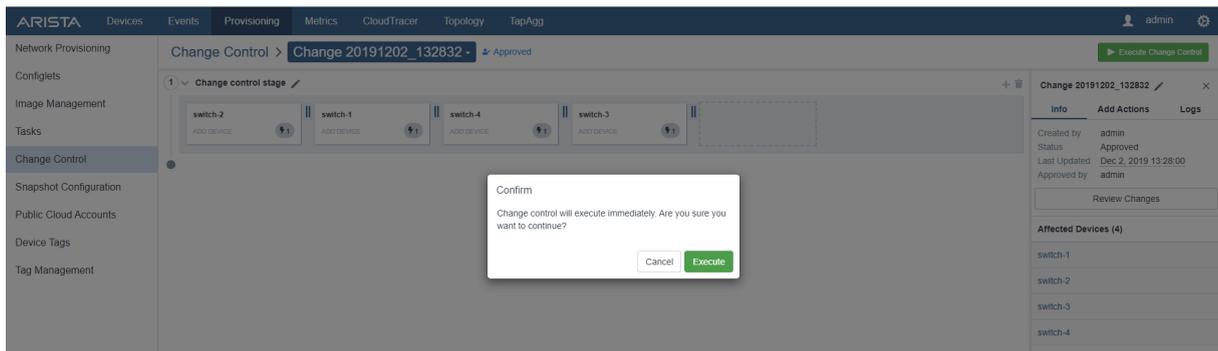
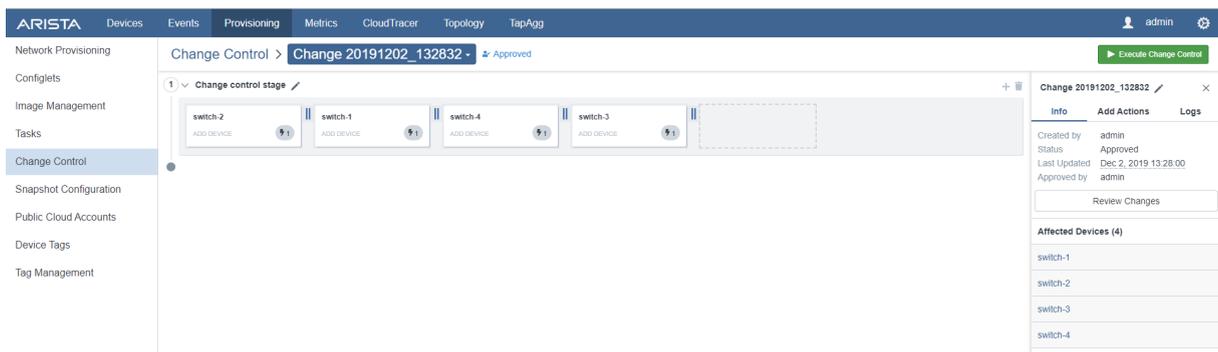
Click on **“Review and Approve”**:

The screenshot shows the ARISTA Change Control page. The left sidebar is the same as in the previous screenshot. The main content area is titled 'Change Control > Change 20191202_132832 - ✓ Changes saved'. Below this is a 'Review and Approve' button. The main content area displays a 'Change control stage' with four device cards: switch-2, switch-1, switch-4, and switch-3, each with an 'ADD DEVICE' button and a status indicator. The right sidebar shows the 'Change 20191202_132832' details, including 'Info', 'Add Actions', and 'Logs' tabs. The 'Info' tab is active, showing 'Created by: admin', 'Status: Pending Approval', and 'Last Updated: Dec 2, 2019 13:28:00'. Below this is a 'Review Changes' button. The 'Affected Devices (4)' list includes switch-1, switch-2, switch-3, and switch-4.

Double check if possible configuration changes are o.k. and click on **“Approve”**:



Click on **“Execute Change Control”** and then on **“Execute”**:



Now the new systems are assigned to the new Container:

The screenshot shows the ARISTA Change Control interface. The main header includes 'ARISTA' and navigation tabs: 'Devices', 'Events', 'Provisioning', 'Metrics', 'CloudTracer', 'Topology', and 'TapAgg'. The user is logged in as 'admin'. The current view is 'Change Control' for a change titled 'Change 20191202_132832', which is marked as 'Completed'. Below this, a 'Change control stage' shows four switches: switch-2, switch-1, switch-4, and switch-3. Each switch has an 'ADD DEVICE' button and a progress indicator. On the right, a summary panel for the change includes: 'Info' (Created by: admin, Status: Completed, Last Updated: Dec 2, 2019 13:28:00, Started: Dec 2, 2019 13:33:02, Approved by: admin), 'Action Status' (Succeeded (4/4)), and 'Affected Devices (4)' (switch-1, switch-2, switch-3, switch-4).

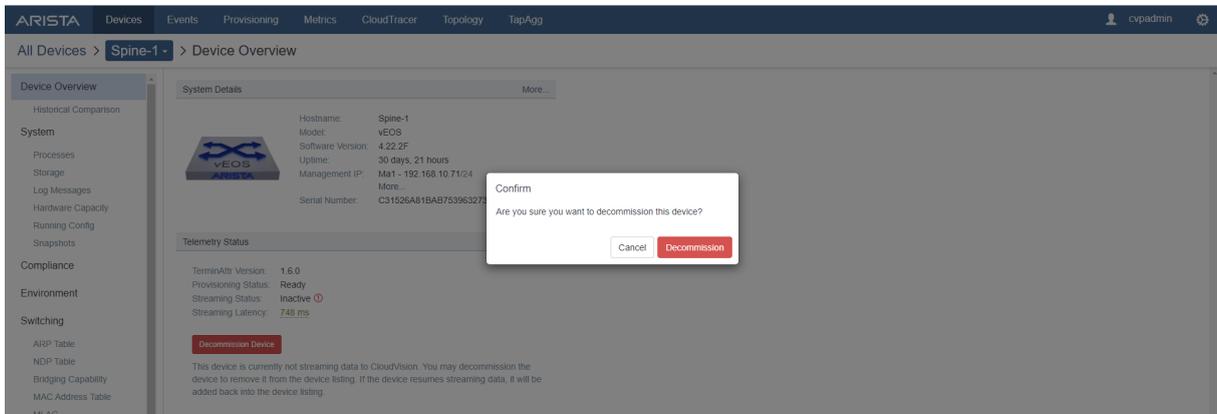
The screenshot shows the ARISTA Network Provisioning interface. The main header includes 'ARISTA' and navigation tabs: 'Devices', 'Events', 'Provisioning', 'Metrics', 'CloudTracer', 'Topology', and 'TapAgg'. The user is logged in as 'admin'. The current view is 'Network Provisioning'. A search bar is visible. The main area displays a network topology diagram. At the top is a 'Tenant (4)' node. Below it are two nodes: 'Undefined (0)' and 'Testlab (4)'. The 'Testlab (4)' node is connected to four switch nodes: 'switch-1', 'switch-2', 'switch-3', and 'switch-4'. At the bottom of the diagram area are buttons for 'Preview', 'Save', and 'Cancel'.

Removing a device from CloudVision

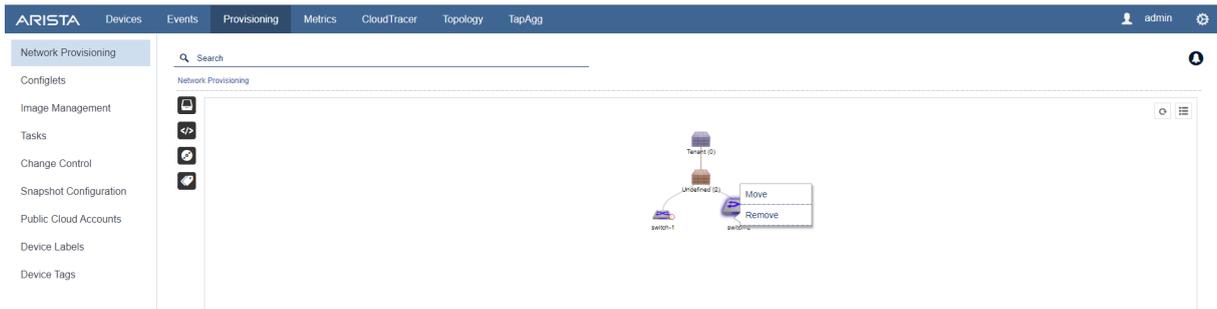
If a device has been offline for a longer time simply select a device from the Inventory List and click on the Device name:



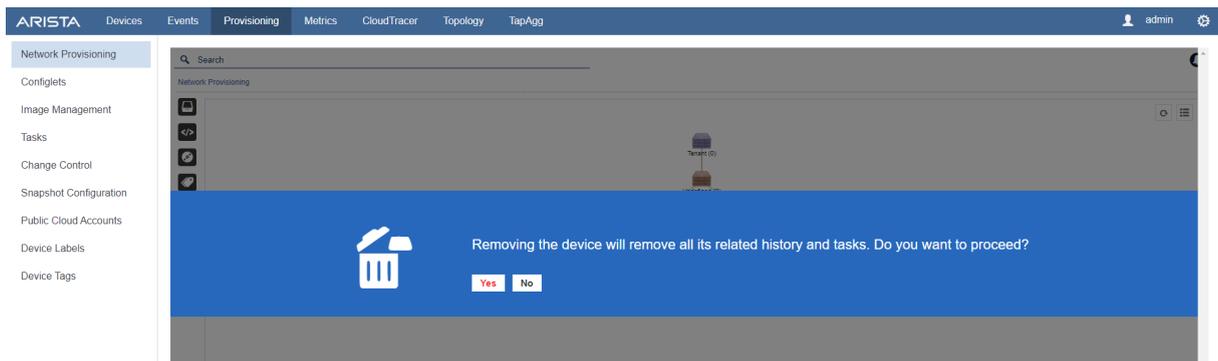
Click on **“Decommission Device”** and confirm the Decommission:



Go to the Tab **“Provisioning”**, right click on a device and select **“Remove”**:



Confirm with “Yes”:



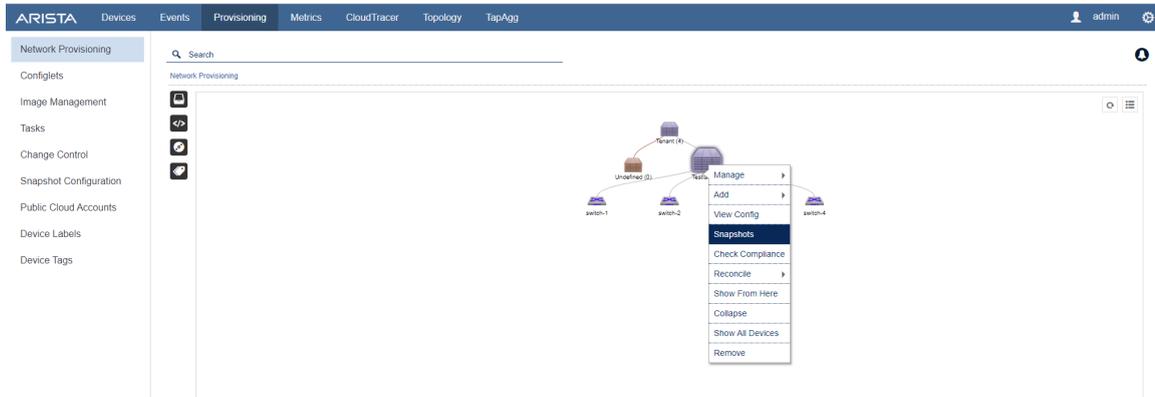
Don't forget to click “save” at the bottom of the tab



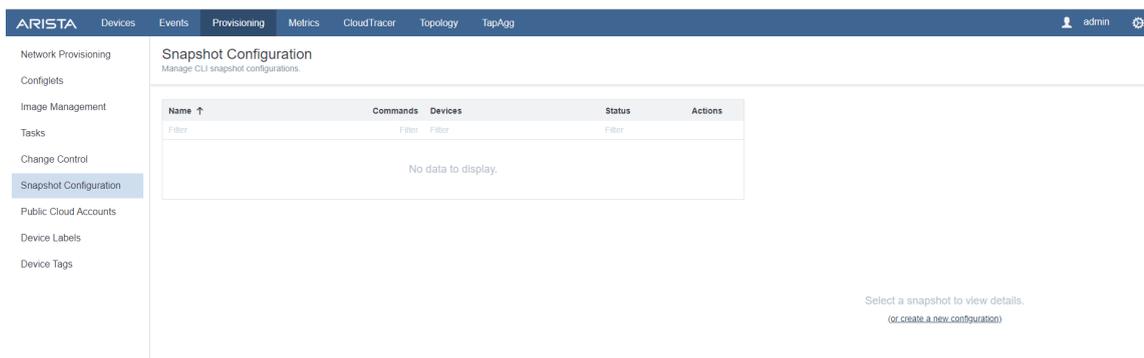
Snapshots:

You can configure CloudVision to create **regular Snapshots** of Devices by **executing command on a regular basis** to view differences within the configurations:

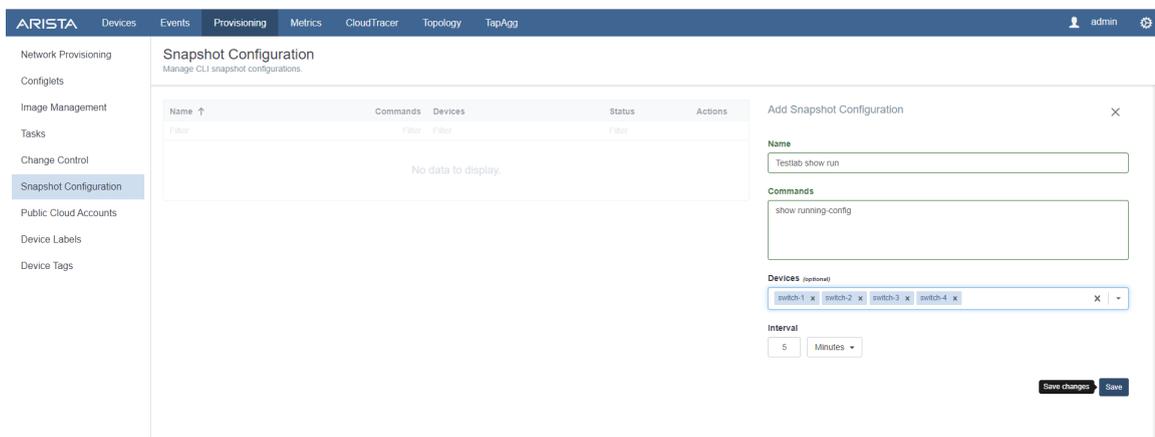
Right Click on a Container and select **“Snapshots”**



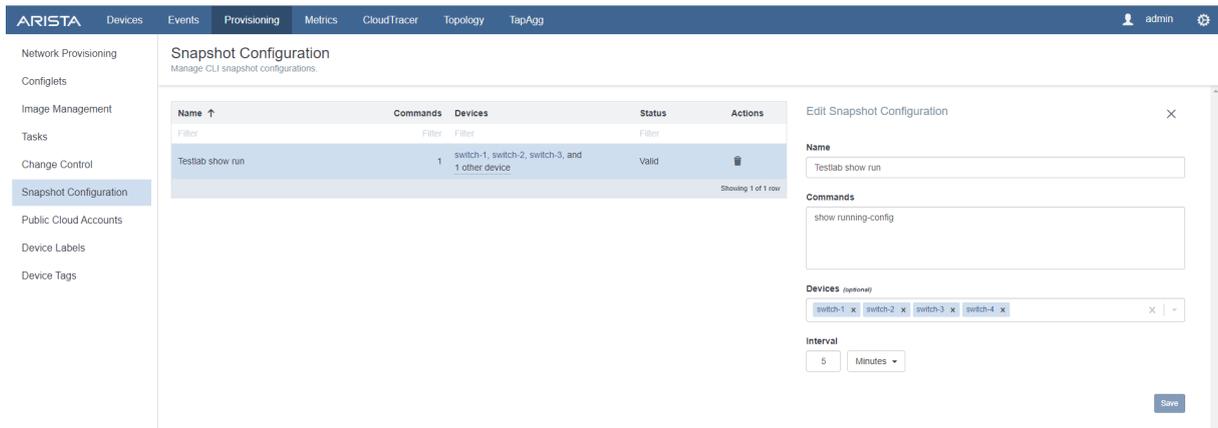
Click on **“Create a new Configuration”**:



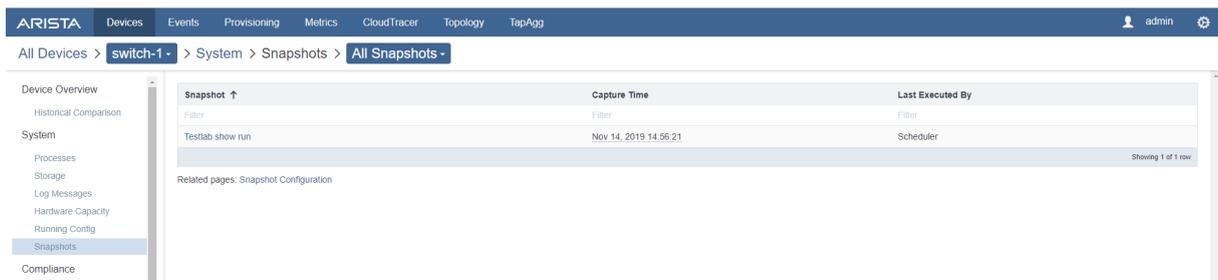
Fill in all the requested information but it is important that **you must not use any shortcuts for EOS commands as this will create an error** (e.g. you must use “show interfaces management 1” instead of “sho int ma 1”). Select the devices and click **“Save”**:



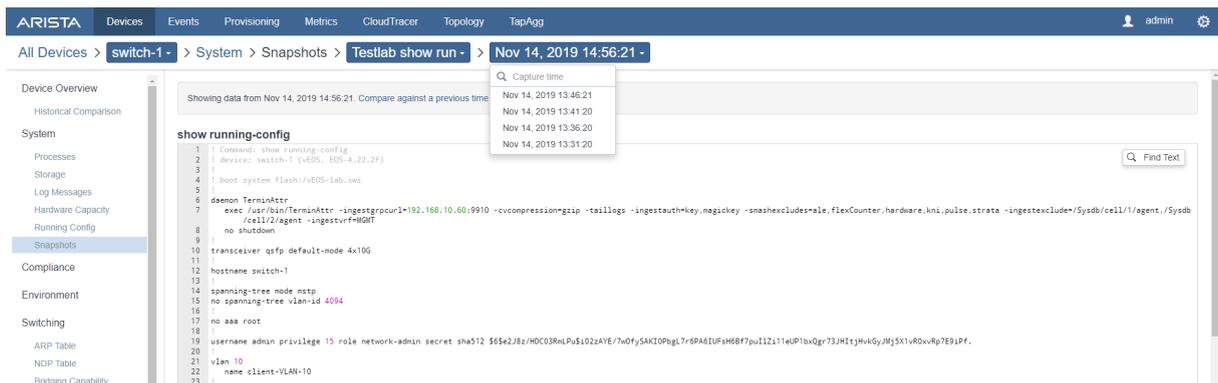
The Snapshot first will be displayed with the status **“Pending”** but after the configured time the status will change to **“Valid”** as soon as the first Snapshot has been created:



When clicking on a device you can view the Snapshot outputs and even differences compared to other Snapshots:



Click on a configured Snapshot and select the desired time where the Snapshot has been created:



Click on “Compare against a previous time” for Snapshot comparison:

ARISTA | Devices | Events | Provisioning | Metrics | CloudTracer | Topology | TapAgg | admin

All Devices > switch-1 > System > Snapshots > Testlab show run - > Nov 14, 2019 14:56:21 -

Device Overview

- Historical Comparison
- System
- Processes
- Storage
- Log Messages
- Hardware Capacity
- Running Config
- Snapshots
- Compliance
- Environment
- Switching
 - ARP Table
 - NDRP Table
 - Bridging Capability
 - MAC Address Table
 - MLAG
 - VXLAN

Comparing data from Nov 14, 2019 14:51:21 against data from Nov 14, 2019 14:56:21. Return to normal view

Compare time

show running-co

Line	Nov 14, 2019 14:51:21	Nov 14, 2019 14:56:21
75	Expand 76 lines	
77	no lldp re	77 no lldp receive
78	!	78 !
79	interface Vlan	
80	description	
81	no autostr	
82	ip address	
83	!	
84	ip route vrf	79 ip route vrf NGMT 0.0.0.0/0 192.168.10.1
85	!	80 !
86	no ip routing	81 no ip routing
87	no ip routing	82 no ip routing vrf NGMT
88	!	83 !
89	mlag configur	
90	domain-id	
91	local-inte	
92	peer-addre	
93	peer-link	
94	reload-del	
95	reload-del	
96	!	
97	management api http-commands	84 management api http-commands
98	no shutdown	85 no shutdown
99	Expand 8 lines	

Unified Split

Configlets

Configlets are parts of a configuration and can be applied to devices or containers.

Containers are a logical entity used to group network devices and to define a hierarchy to which Configlets can be applied. When you apply a Configlet to a Container, the Configlet is automatically applied to all of the devices in the container's **hierarchy**.

Users can create new configlets or import them.

There are **static** and **Builder based** Configlets whereas the static ones don't contain any variables (e.g. configuring a common fixed syslog Server on several systems) and the Builder based Configlets are using **Input Forms** which makes them more flexible.

Configlets

Name	Containers	Devices	Notes	Type - All	Created By	Created Date
EX13_deviceAPI_MgmtIntBuilder	0	0	Add Note	Builder	admin	2019-11-15 14:17:16
EX1_Form_MgmtIntBuilder	0	0	Add Note	Builder	admin	2019-11-15 13:15:44
EX6_BasicStatic	0	0	Add Note	Static	cvpadmin	2019-11-15 14:30:44
Fabric_builder_v5.1_1	0	0	Add Note	Builder	admin	2019-11-15 13:51:28
FreePorts	0	0	Add Note	Builder	admin	2019-11-15 13:19:06
SYS_TelemetryBuilderV2	0	0	Add Note	Builder	cvp.system	2019-10-02 15:36:23

1 - 6 of 6 << 1 of 1 >>

Important Note:

Configlets may set the running config back to default and then apply only the configured changes so that the systems may lose connectivity in case there is no reconciliation done upfront of the running configuration! You may see this as a warning within the Change Control Tab before approving applying a Configlet to systems

The screenshot shows the ARISTA Change Control interface. The main area displays a change control stage for 'Change 20191202_134329'. It lists four switches: switch-1, switch-3, switch-4, and switch-2, each with an 'UPDATE CONFIG' button and a status indicator. The right sidebar shows the change control details, including the change ID, status (Pending Approval), and a list of affected devices (switch-1, switch-2, switch-3, switch-4).

Sometimes it may be necessary to allow Python to use self-signed certs in case no certificate is added to the EOS device! In this case the attached lines of code have to be added to the Python script:

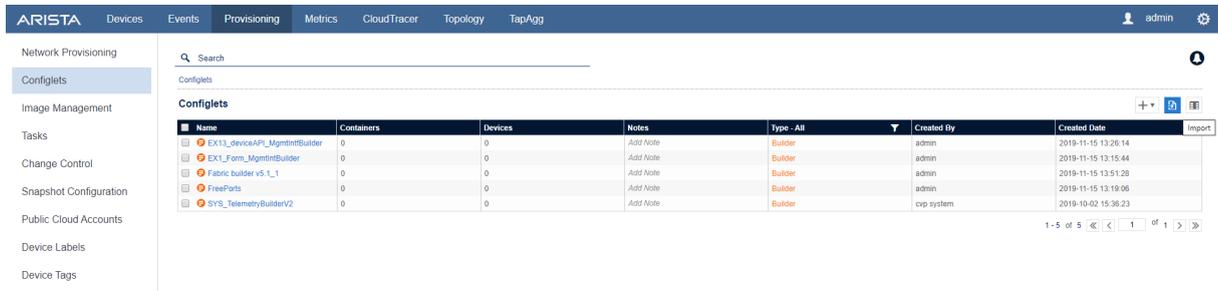
```
import ssl
ssl._https_verify_certificates(False)
```

Importing Configlets:

Already created example Configlets can be found on GitHub at:

<https://github.com/aristanetworks/CloudVisionPortal-Examples>

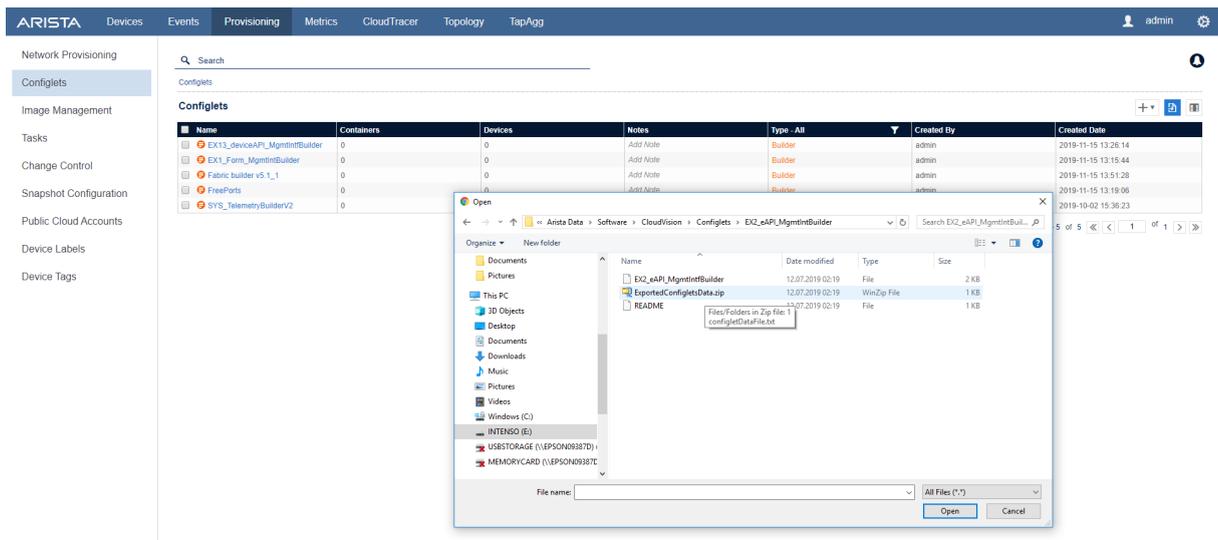
Under the Tab “Provisioning” select “Configlets and click on the “Import” button:



The screenshot shows the ARISTA CloudVision Portal interface. The top navigation bar includes 'ARISTA', 'Devices', 'Events', 'Provisioning', 'Metrics', 'CloudTracer', 'Topology', and 'TapAgg'. The left sidebar lists various management options like 'Network Provisioning', 'Configlets', 'Image Management', etc. The main content area displays a table of Configlets.

Name	Containers	Devices	Notes	Type	Created By	Created Date	Import
EX13_deviceAPI_MgmtIntfBuilder	0	0	Add Note	Builder	admin	2019-11-15 13:26:14	
EX1_Form_MgmtIntfBuilder	0	0	Add Note	Builder	admin	2019-11-15 13:15:44	
Fabric builder v5_1_1	0	0	Add Note	Builder	admin	2019-11-15 13:51:28	
FreePorts	0	0	Add Note	Builder	admin	2019-11-15 13:19:06	
SYS_TelemetryBuilderV2	0	0	Add Note	Builder	cvp system	2019-10-02 15:36:23	

Select a “ExportedConfigletsData.zip” File to import a Configlet:



The screenshot shows the same ARISTA CloudVision Portal interface as above, but with a file explorer window open. The file explorer is displaying the contents of a folder named 'EX13_deviceAPI_MgmtIntfBuilder'. The file 'ExportedConfigletsData.zip' is highlighted, indicating it is selected for import.

After the successful import the new Configlet can be reviewed and edited:

The screenshot shows the ARISTA Configlet editor for 'EX2_eAPI_MgmtIntBuilder'. The 'Main Script' tab contains the following code:

```
1 import json
2 from cvplibrary import CVPGlobalVariables, GlobalVariableNames
3
4 #Just to test:
5 # - uncomment the setElement line below
6 # - set the mac address to a known device's mac
7 # - Press Generate button
8 #CVPGlobalVariables.setElement( "00:0C:29:A9:2E:2F" );
9
10 ztp = CVPGlobalVariables.getValue( GlobalVariableNames.ZTP_STATE );
11 ip = CVPGlobalVariables.getValue( GlobalVariableNames.CVP_IP );
12
13- if ztp == "true":
14     user = CVPGlobalVariables.getValue( GlobalVariableNames.ZTP_USERNAME );
15     passwd = CVPGlobalVariables.getValue( GlobalVariableNames.ZTP_PASSWORD );
16- else:
17     user = CVPGlobalVariables.getValue( GlobalVariableNames.CVP_USERNAME );
18     passwd = CVPGlobalVariables.getValue( GlobalVariableNames.CVP_PASSWORD );
19
20 url = "https://%s:%s@%s/command-api" % ( user, passwd, ip )
21 ss = jsonplib.Server( url )
22
23 hostname = ss.runCmds( [ [ "enable", [ "cmd": "show hostname" ] ] ] )
24 intfs = ss.runCmds( [ [ "enable", [ "cmd": "show interface Management 1" ] ] ] )
25 mgmtip = intfs[ 'interfaces' ][ 'Management1' ][ 'interfaceAddress' ][ 0 ][ 'primaryIp' ][
26     'address' ];
27 mgmtMask = intfs[ 'interfaces' ][ 'Management1' ][ 'interfaceAddress' ][ 0 ][ 'primaryIp' ][
28     'maskLen' ];
29
30 print "hostname %s" % hostname[ 'fqdn' ]
31 print "interface Management 1"
32 print " ip address %s/%s" % ( mgmtip, mgmtMask )
33
```

Click "Save" to update the changed Configlet:

The screenshot shows the ARISTA Configlet editor for 'EX13_deviceAPI_MgmtIntBuilder'. The 'Main Script' tab contains the following code:

```
1 from cvplibrary import CVPGlobalVariables, GlobalVariableNames
2 from cvplibrary import Device
3 import ssl
4 ssl_https_verify_certificates(False)
5
6 device_ip = CVPGlobalVariables.getValue( GlobalVariableNames.CVP_IP )
7 ztp = CVPGlobalVariables.getValue( GlobalVariableNames.ZTP_STATE );
8
9- if ztp == "true":
10     user = CVPGlobalVariables.getValue( GlobalVariableNames.ZTP_USERNAME );
11     passwd = CVPGlobalVariables.getValue( GlobalVariableNames.ZTP_PASSWORD );
12- else:
13     user = CVPGlobalVariables.getValue( GlobalVariableNames.CVP_USERNAME );
14     passwd = CVPGlobalVariables.getValue( GlobalVariableNames.CVP_PASSWORD );
15
16 device = Device( device_ip, username=user, password=passwd )
17 cmdList = [ "enable", "show hostname", "show interfaces Management1" ]
18 cmdResponse = device.runCmds( cmdList )
19 hostname = cmdResponse[1][ 'response' ][ 'hostname' ]
20 intfs = cmdResponse[2][ 'response' ];
21 mgmtip = intfs[ 'interfaces' ][ 'Management1' ][ 'interfaceAddress' ][ 0 ][ 'primaryIp' ][
22     'address' ];
23 mgmtMask = intfs[ 'interfaces' ][ 'Management1' ][ 'interfaceAddress' ][ 0 ][ 'primaryIp' ][
24     'maskLen' ];
25 mgmt = "Management1"
26 print "hostname %s" % hostname
27 print "interface Management1"
28 print " ip address %s/%s" % ( mgmtip, mgmtMask )
29
```

At the bottom of the interface, there are buttons for 'Save as Draft', 'Save', 'Save As', and 'Cancel'.

Click “Generate” to test the Configlet:

The screenshot shows the ARISTA configuration tool interface. The 'Form Design' tab is active, displaying a configuration form for a device named 'switch-2' with IP address '192.168.10.72'. A 'Generate' button is prominently displayed. The 'Main Script' and 'Built Configlet' tabs are also visible, showing the underlying configuration code.

```

Main Script
1 from cvplibrary import CVPGlobalVariables, GlobalVariableNames
2 from cvplibrary import Device
3 import ssl
4 ssl_https_verify_certificates(False)
5
6 device_ip = CVPGlobalVariables.getValue(GlobalVariableNames.CVP_IP)
7 stp = CVPGlobalVariables.getValue(GlobalVariableNames.ZTP_STATE)
8
9 if stp == 'true':
10 user = CVPGlobalVariables.getValue(GlobalVariableNames.ZTP_USERNAME)
11 passwd = CVPGlobalVariables.getValue(GlobalVariableNames.ZTP_PASSWORD)
12 else:
13 user = CVPGlobalVariables.getValue(GlobalVariableNames.CVP_USERNAME)
14 passwd = CVPGlobalVariables.getValue(GlobalVariableNames.CVP_PASSWORD)
15
16 device = Device(device_ip,username=user,password=passwd)
17 cmdList = [ "enable", "show hostname", "show interfaces Management1" ]
18 cmdResponse = device.runCmds(cmdList)
19 hostname = cmdResponse[0]['response']['hostname']
20 intfs = cmdResponse[2]['response']
21 mgmtIp = intfs['interfaces']['Management1']['interfaceAddress']
22 mgmtMask = intfs['interfaces']['Management1']['interfaceAddress']['primaryIp']['masklen']
23 mgmt = "Management1"
24 print "hostname %s" % hostname
25 print "interface Management1"
26 print "ip address %s/%s" % ( mgmtIp, mgmtMask )
27
Built Configlet
1 hostname switch-2
2 interface Management1
3 ip address 192.168.10.72/24
    
```

Changes within edited **static Configlets** can be reviewed by clicking on “View”:

The screenshot shows the ARISTA configuration tool interface. The 'Change History' tab is active, displaying a table of changes for the static configlet 'EX6_BasicStatic'. The 'View' button is visible next to the 'admin' user entry.

User Name	Update On	View
admin	2019-11-15 14:34:43	View

The screenshot shows the ARISTA configuration tool interface. The 'View Configlet' dialog box is open, displaying the 'Old Configlet' and 'Updated Configlet' side-by-side. The 'Old Configlet' shows the original configuration, and the 'Updated Configlet' shows the changes made by the user.

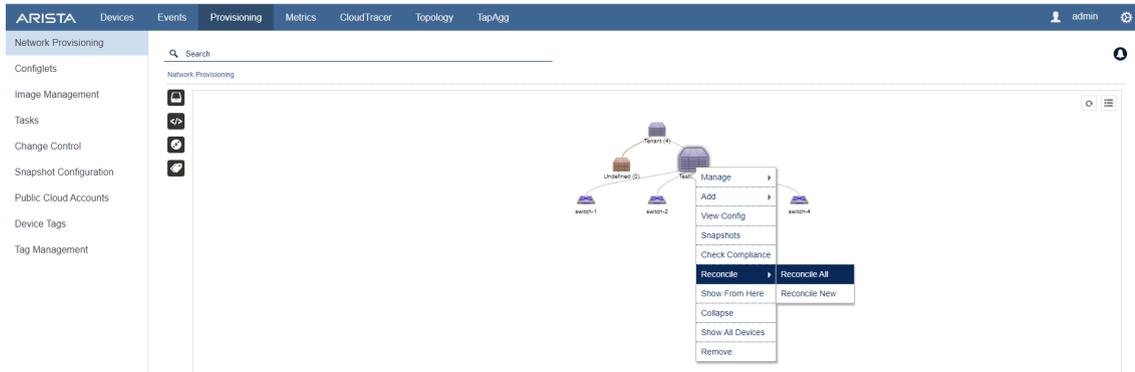
Old Configlet	Updated Configlet
1 load-interval default 5	1 load-interval default 5
2	2
3 no logging console	3 no logging console
4	4
5 logging vrt MGMT host 10.1.16.253	5 logging vrt MGMT host 192.168.10.60
6	6
7 default ip name-server	7 default ip name-server

Created: cypadmin | Date & Time: 2019-11-15 14:30:44
 Updated: admin | Date & Time: 2019-11-15 14:34:43

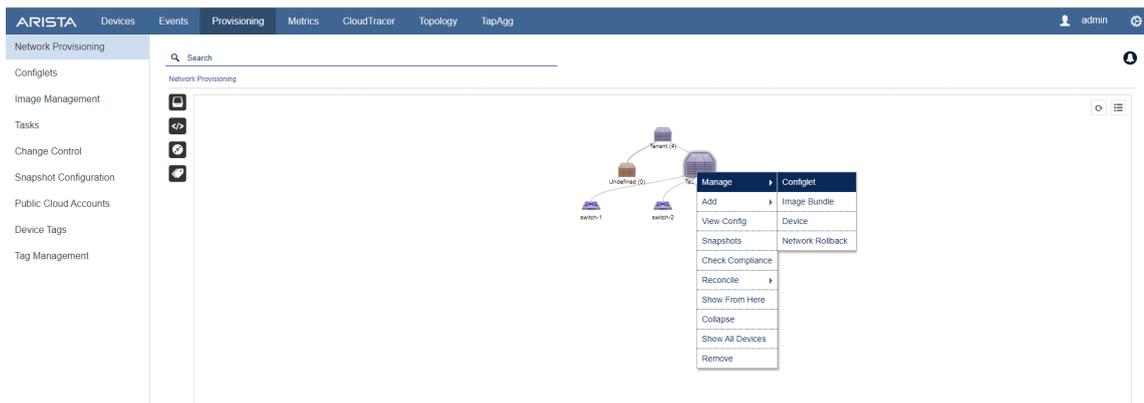
Applying Configlets to a Container:

Configlets can easily applied to several devices within a Container.

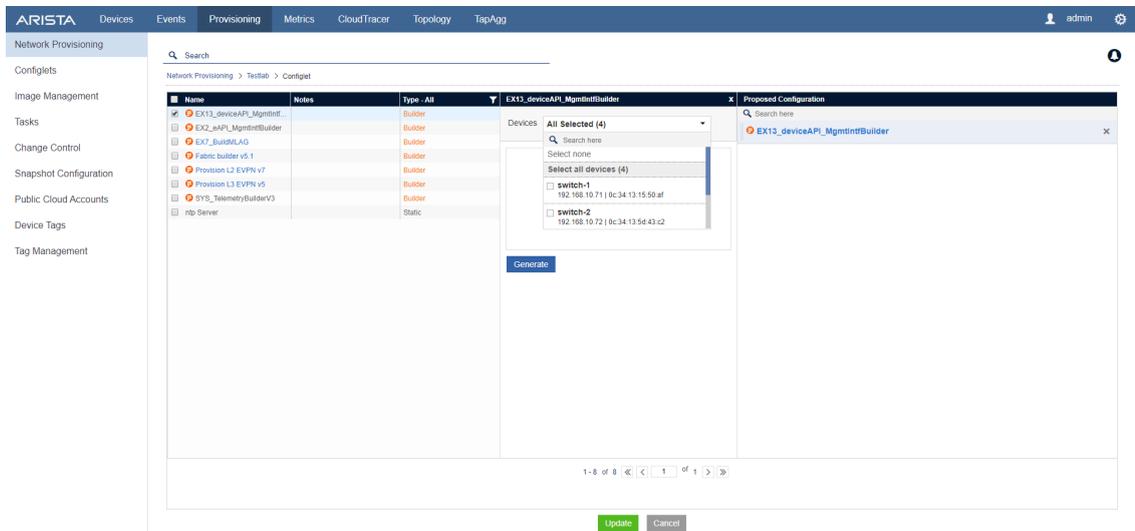
First step would be to initiate a “**Reconcile All**” on the Container to backup the running-config:



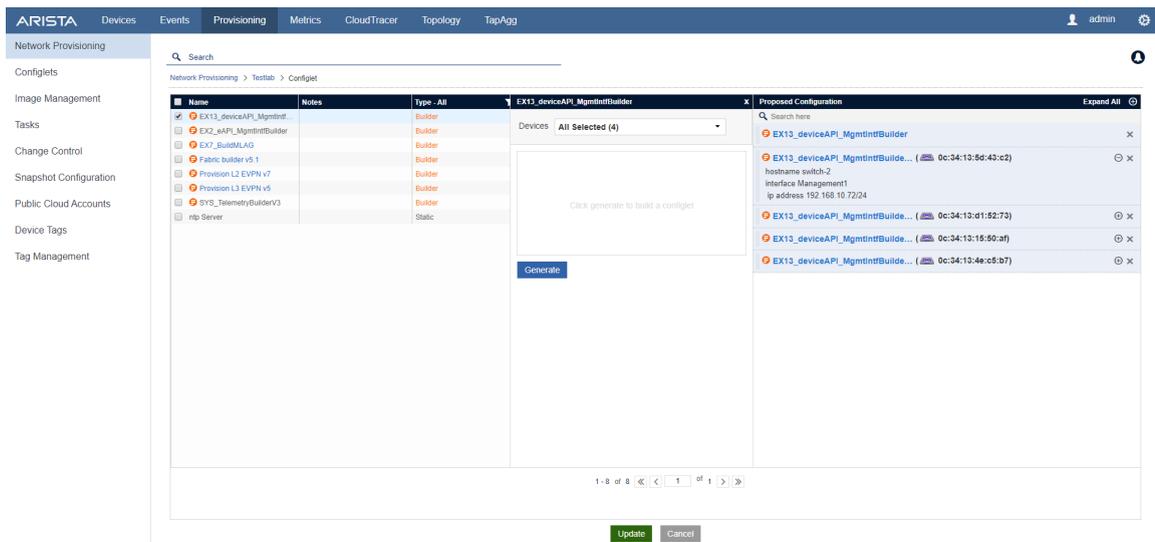
Right click on the Container and select **Manage -> Configlet**:



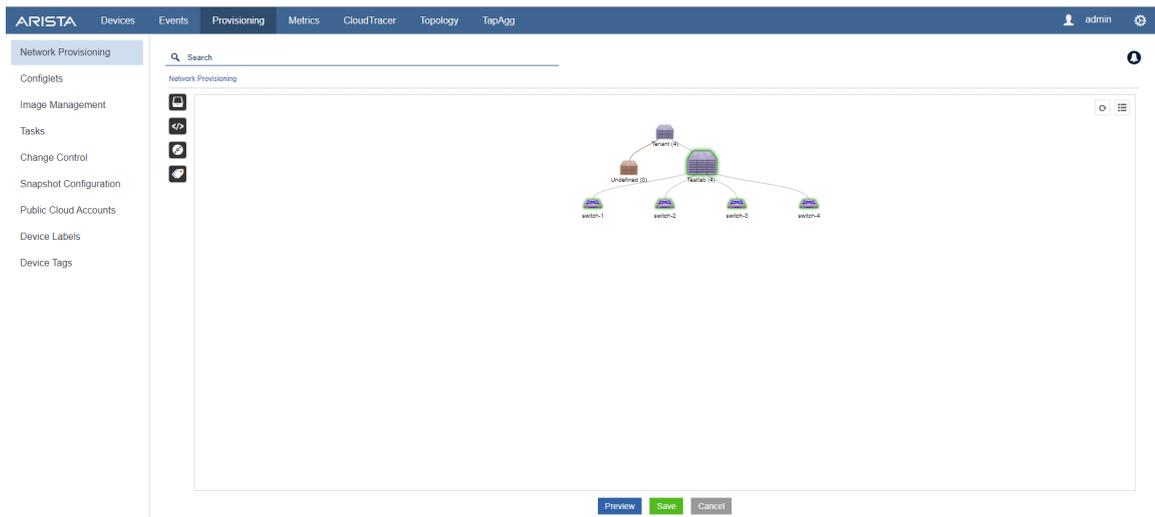
Select one or more Configlets and in case of a dynamic Configlet also the desired target devices within the Container:



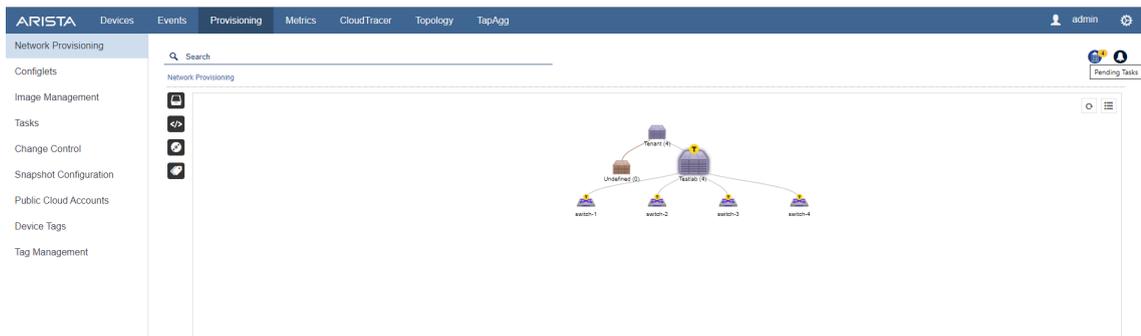
Click **“Generate”** and then **“Update”**:



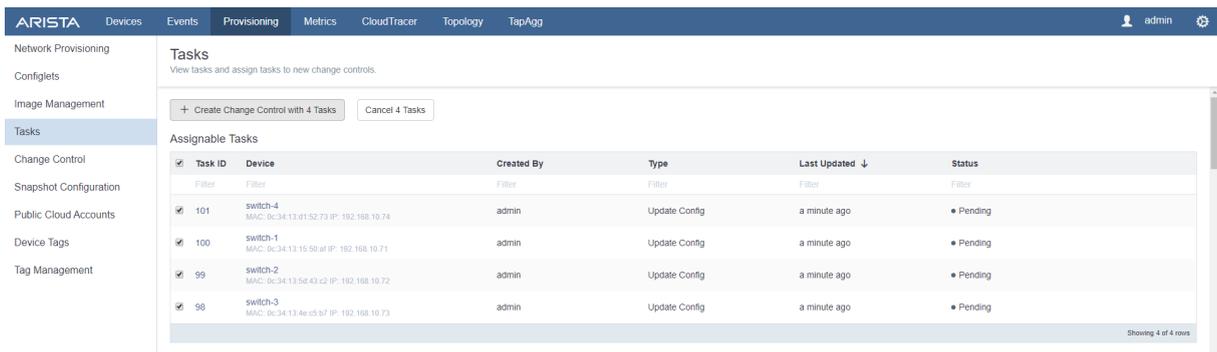
Click **“Save”**:



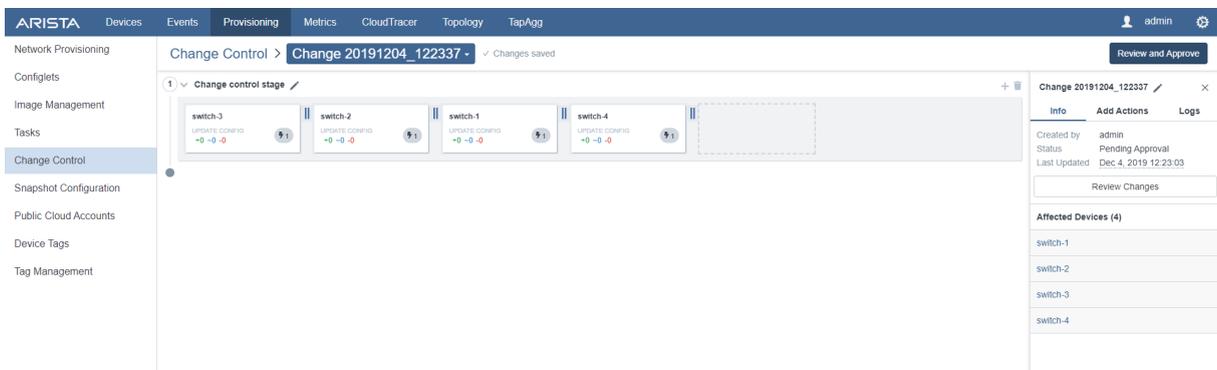
Click on **“Pending Tasks”**:



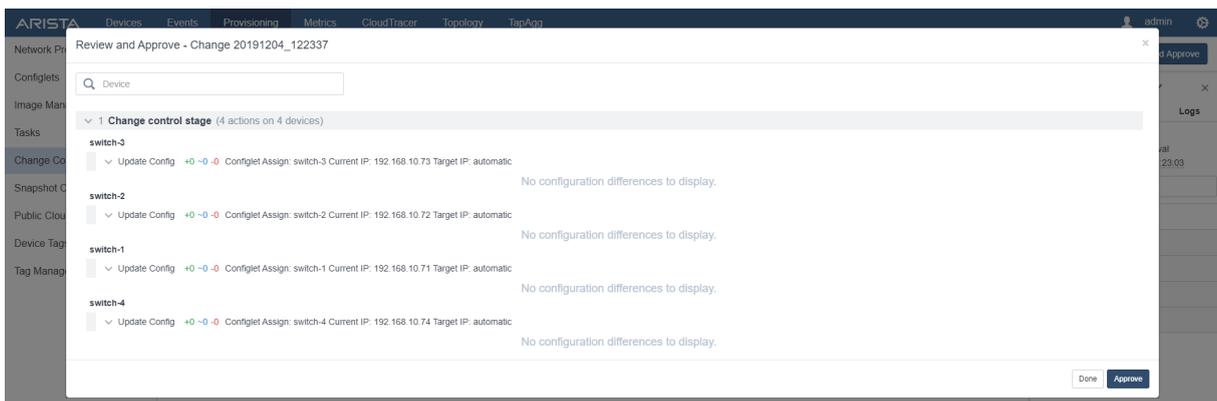
Select the Assignable Tasks and click on **“Create Change Control with x Tasks”**:



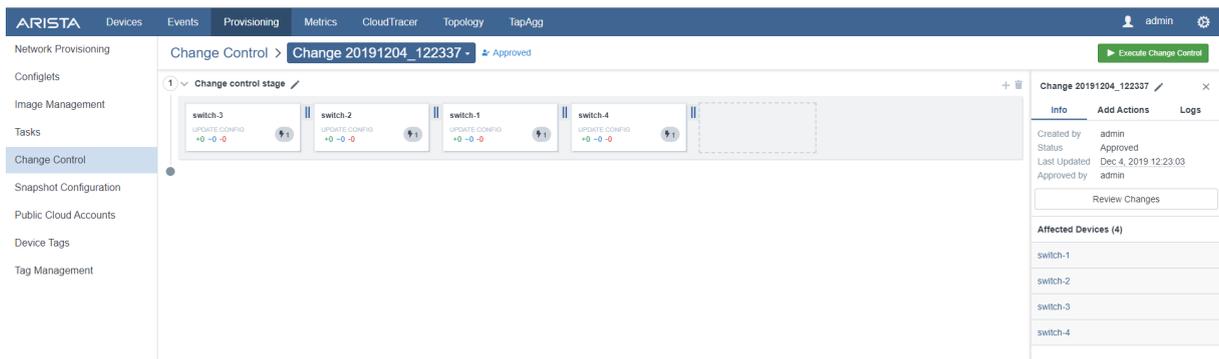
Click on **“Review & Approve”**:



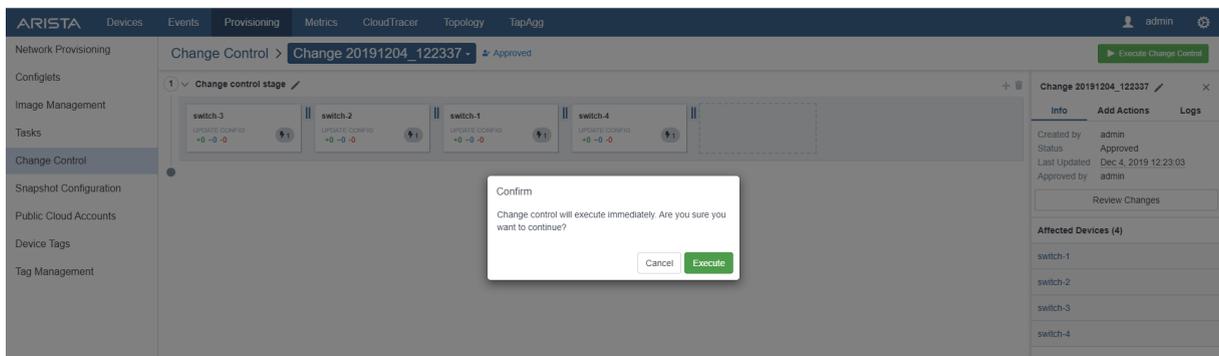
Click on **“Approve”**:



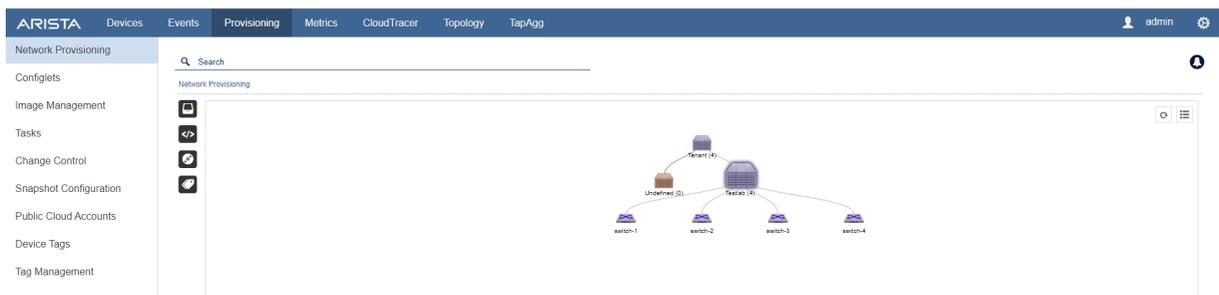
Click on **“Execute Change Control”**:



Click on **“Execute”**:



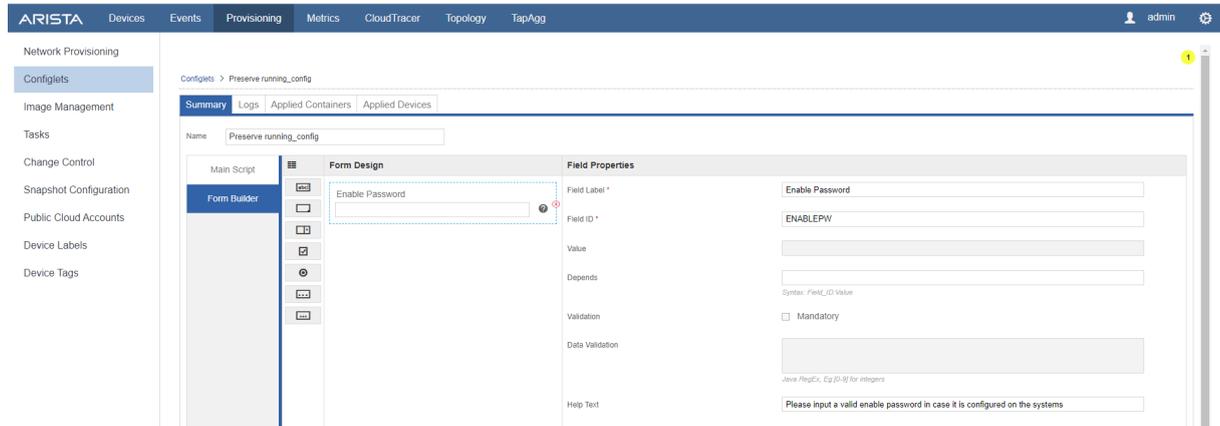
The Configlets will be pushed to the devices within the Container and the tasks will be marked as **“Completed”**:



Adding parts of a configuration without reconciliation:

By adding the attached code into an existing Configlet it is possible to add configuration parts to the existing running-config of a system **without doing a "Reconcile All"**:

Form Builder:



Main Script:

```
# written by Martin Krausch, Systems Engineer Arista Networks
# (c) Arista Networks 2020
# Usage at your own risk!
#*****

from cvplib import Form
from cvplib import CVPGlobalVariables, GlobalVariableNames
from cvplib import Device
from jsonrpclib import Server
import ssl
ssl._https_verify_certificates(False)

enablepw = Form.getFieldById( 'ENABLEPW' ).getValue()
if not enablepw:
    enablepw="None"

device_ip = CVPGlobalVariables.getValue(GlobalVariableNames.CVP_IP)
ztp = CVPGlobalVariables.getValue( GlobalVariableNames.ZTP_STATE );

if ztp == 'true':
    user = CVPGlobalVariables.getValue( GlobalVariableNames.ZTP_USERNAME );
    passwd = CVPGlobalVariables.getValue( GlobalVariableNames.ZTP_PASSWORD );
else:
    user = CVPGlobalVariables.getValue( GlobalVariableNames.CVP_USERNAME );
    passwd = CVPGlobalVariables.getValue( GlobalVariableNames.CVP_PASSWORD );

device = ("https://" + user + ":" + passwd + "@" + device_ip + "/command-api")
switch = Server (device)

cmdResponse = switch.runCmds( 1, [{"cmd": "enable",
                                  "input": enablepw},
                                  "show running-config"],
                              "text")
```

```
response_string = str(cmdResponse[1])

shelllist = (response_string.split("\n"))
for part in shelllist:
    if (not str(part).startswith("{")) and (not str(part).endswith("}")):
        print part

#*****
# Please enter all additional config parts starting from here
#*****
```

Traffic Flows

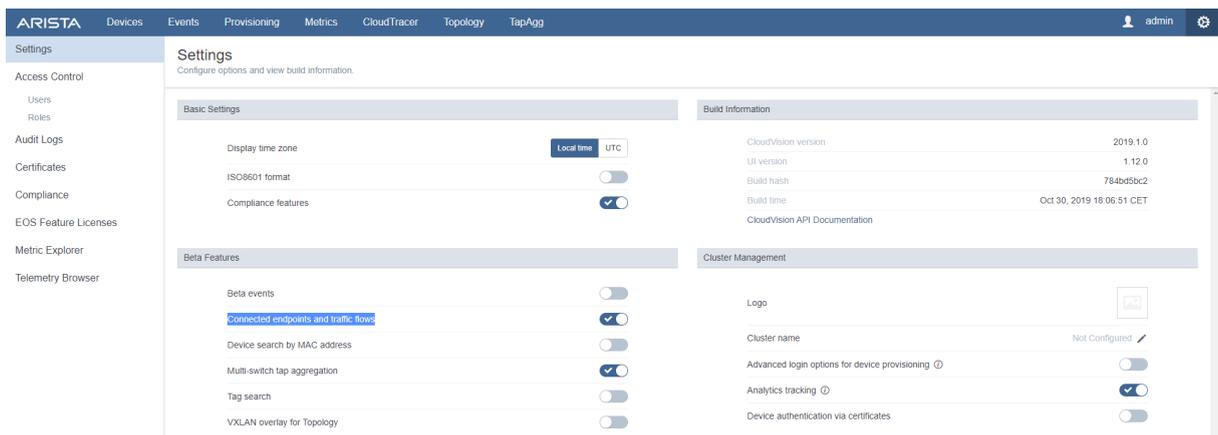
Within **CloudVision v2019-01** Collecting sFlow based traffic flows is still a **Beta Feature** which has to be enabled first:

Click on **“Settings & Tools”**:



Device	Status	Model	Software	Streaming Agent	IP Address	MAC Address	Device ID
switch-2	✓ OK	vEOS	4.22.2F	1.6.1	192.168.10.72	0c:34:13:5d:43:c2	A3AE00208225FE6430...
switch-3	✓ OK	vEOS	4.22.2F	1.6.1	192.168.10.73	0c:34:13:4e:c5:b7	7FA995403E68303594C8...
switch-4	✓ OK	vEOS	4.22.2F	1.6.1	192.168.10.74	0c:34:13:d1:52:73	6C58EDA497E7F2E0254...

Activate **“Connected Endpoints and traffic flows”**:



Settings

Configure options and view build information.

Basic Settings

Display time zone: Local time / UTC

ISO8601 format:

Compliance features:

Beta Features

Beta events:

Connected endpoints and traffic flows:

Device search by MAC address:

Multi-switch tap aggregation:

Tag search:

VXLAN overlay for Topology:

Build Information

CloudVision version: 2019.1.0

UI version: 1.12.0

Build hash: 784bd5bc2

Build time: Oct 30, 2019 18:06:51 CET

CloudVision API Documentation

Cluster Management

Logo:

Cluster name: Not Configured

Advanced login options for device provisioning:

Analytics tracking:

Device authentication via certificates:

EOS Device Configuration:

sFlow Destination has to be set to **127.0.0.1** on the EOS device!

```
vlan 10
  name client-VLAN-10
!
sflow sample 20
sflow polling-interval 10
sflow destination 127.0.0.1
sflow source 192.168.10.73
sflow run
!
interface Ethernet12
  switchport access vlan 10
  sflow enable
!
```

CloudVision Output:

Select a device and the “Traffic Flows”:

The screenshot displays the ARISTA CloudVision interface for a device named 'switch-3'. The navigation bar at the top includes 'ARISTA', 'Devices', 'Events', 'Provisioning', 'Metrics', 'CloudTracer', 'Topology', and 'TapAgg'. The breadcrumb trail shows 'All Devices > switch-3 > Device Overview'. On the left, a sidebar lists various system components like Processes, Storage, Log Messages, Hardware Capacity, Running Config, Snapshots, Compliance, Environment, Tags, Switching, Routing, and Traffic Flows. The main content area is divided into three sections: 'System Details' (hostname: switch-3, model: vEOS, software version: 4.22.2F, uptime: 1 hour, 21 minutes, management IP: 192.168.10.73, device ID: 7FA995403E68303594C8BD7A974EEAEF, MAC address: 0c:34:13:4e:c5:b7), 'System Status' (streaming agent version: 1.6.1, mode: Normal, provisioning status: Ready, streaming status: Active, latency: 169.144 ms, compliance status: 1 bug), and 'Interface Counts' (12 Ethernet Interfaces, 0 VLAN Interfaces, 1 IP Interfaces, 1 Port Channels).

Here you can see different flows and optional traffic filters:

The screenshot shows the 'Traffic Flows' section of the ARISTA CloudVision interface for 'switch-3'. The breadcrumb trail is 'All Devices > switch-3 > Traffic Flows'. The interface features a search bar with fields for 'IP', 'Port', 'To', and 'IP', along with a 'Clear Filters' button and a 'Sort By' dropdown set to 'Bandwidth (bytes)'. Below the search bar, it indicates '2 matching flow records (342.9 KB bandwidth used, 1,604 packets, 2 active hosts, sampling rate: 1/16), refreshing in 4 seconds'. The 'Split By' dropdown is set to 'Destination Host'. The main area displays two horizontal bar charts: 'Top Destination Hosts' and 'Top Source Hosts'. The 'Top Destination Hosts' chart shows two bars for IP addresses 192.168.50.1 and 192.168.50.2. The 'Top Source Hosts' chart shows two bars for IP addresses 192.168.50.2 and 192.168.50.1. To the right of these charts are two line graphs showing bandwidth usage over time. A tooltip for the top destination host shows a peak of 5.3 Mbps on Dec 9, 2019 at 13:08:19. The bottom of the interface includes a 'Top Destination Ports' chart and a time-based bandwidth usage graph.

Integration of 3rd party devices into CloudVision

CloudVision 2019 supports a new **Device SDK** for visibility into legacy platforms that do not support OpenConfig. Using this SDK, CloudVision can pull visibility metrics from **SNMP-capable** platforms.

Configuration of an Extreme Networks XOS based 3rd party device:

In this manual we are including an **X440G2-12P** from the vendor **Extreme Networks** into CloudVision. As Extreme XOS based products **don't support industry standard CLI** attached are two short examples how to configure SNMP.

Configuring SNMPv3 access:

```
X440G2-12p-10G4.13 # configure snmpv3 add user snmpuser authentication md5 privacy des
Authentication password: snmpauthcred
Reenter authentication password: snmpauthcred
Privacy password: snmprivcred
Reenter privacy password: snmprivcred
* X440G2-12p-10G4.14 # configure snmpv3 add group admin user snmpuser sec-model
usm
* X440G2-12p-10G4.15 # enable snmp access
* X440G2-12p-10G4.16 # configure snmp sysContact "Extreme Networks"
* X440G2-12p-10G4.17 # configure snmp sysLocation "Demo"
* X440G2-12p-10G4.18 # configure snmp sysName "X440G2-12p"
```

Configuring SNMPv2c access:

```
* X440G2-12p.22 # configure snmp add community readwrite secretpw
* X440G2-12p.23 # enable snmp access snmp-v1v2c
* X440G2-12p-10G4.24 # configure snmp sysContact "Extreme Networks"
* X440G2-12p-10G4.25 # configure snmp sysLocation "Demo"
* X440G2-12p-10G4.26 # configure snmp sysName "X440G2-12p"
```

Configuration of CloudVision

Connect to CloudVision via SSH or Console using the default credentials:

Username: **root**
Password: **password**

Copy the latest Collector modules (**please contact am Arista SE how to get the latest ones!**) to the folder **"/home/cvpadadmin/collector_upgrade"** which has to be created first:

```
[root@cvp ~]# cd /home/cvpadadmin/
[root@cvp cvpadadmin]# mkdir collector_upgrade
[root@cvp cvpadadmin]# ls
collector_upgrade
[root@cvp cvpadadmin]# cd collector_upgrade/
[root@cvp collector_upgrade]# wget
ftp://mkrausch:password@192.168.10.206/collector-eft-setup-latest.sh
--2019-12-13 10:09:13--
ftp://mkrausch:*password*@192.168.10.206/collector-eft-setup-latest.sh
      => 'collector-eft-setup-latest.sh'
Connecting to 192.168.10.206:21... connected.
Logging in as mkrausch ... Logged in!
==> SYST ... done.      ==> PWD ... done.
==> TYPE I ... done.   ==> CWD not needed.
==> SIZE collector-eft-setup-latest.sh ... 2955
==> PASV ... done.     ==> RETR collector-eft-setup-latest.sh ... done.
Length: 2955 (2.9K) (unauthoritative)

100%[=====]
=====>] 2,955      --.-K/s   in 0.006s

2019-12-13 10:09:13 (460 KB/s) - 'collector-eft-setup-latest.sh' saved [2955]

[root@cvp collector_upgrade]# wget
ftp://mkrausch:password@192.168.10.206/Collector-latest
--2019-12-13 10:10:57-- ftp://mkrausch:*password*@192.168.10.206/Collector-latest
      => 'Collector-latest'
Connecting to 192.168.10.206:21... connected.
Logging in as mkrausch ... Logged in!
==> SYST ... done.      ==> PWD ... done.
==> TYPE I ... done.   ==> CWD not needed.
==> SIZE Collector-latest ... 10694208
==> PASV ... done.     ==> RETR Collector-latest ... done.
Length: 10694208 (10M) (unauthoritative)

100%[=====]
=====>] 10,694,208  15.0MB/s   in 0.7s

2019-12-13 10:10:58 (15.0 MB/s) - 'Collector-latest' saved [10694208]

[root@cvp collector_upgrade]# wget
ftp://mkrausch:password@192.168.10.206/GNMIAdapter-latest
--2019-12-13 10:11:50--
ftp://mkrausch:*password*@192.168.10.206/GNMIAdapter-latest
      => 'GNMIAdapter-latest'
```

```

Connecting to 192.168.10.206:21... connected.
Logging in as mkrausch ... Logged in!
==> SYST ... done.    ==> PWD ... done.
==> TYPE I ... done.  ==> CWD not needed.
==> SIZE GNMIAdapter-latest ... 16189120
==> PASV ... done.    ==> RETR GNMIAdapter-latest ... done.
Length: 16189120 (15M) (unauthoritative)

100%[=====
=====>] 16,189,120  15.8MB/s  in 1.0s

2019-12-13 10:11:51 (15.8 MB/s) - 'GNMIAdapter-latest' saved [16189120]

[root@cvp collector_upgrade]#
[root@cvp collector_upgrade]# wget
ftp://mkrausch:password@192.168.10.206/GNMIAdapter-yang-modules-latest.tar.gz
--2019-12-13 10:12:51--
ftp://mkrausch:*password*@192.168.10.206/GNMIAdapter-yang-modules-latest.tar.gz
=> 'GNMIAdapter-yang-modules-latest.tar.gz'
Connecting to 192.168.10.206:21... connected.
Logging in as mkrausch ... Logged in!
==> SYST ... done.    ==> PWD ... done.
==> TYPE I ... done.  ==> CWD not needed.
==> SIZE GNMIAdapter-yang-modules-latest.tar.gz ... 662847
==> PASV ... done.    ==> RETR GNMIAdapter-yang-modules-latest.tar.gz ... done.
Length: 662847 (647K) (unauthoritative)

100%[=====
=====>] 662,847  --.-K/s  in 0.05s

2019-12-13 10:12:51 (13.3 MB/s) - 'GNMIAdapter-yang-modules-latest.tar.gz' saved
[662847]

[root@cvp collector_upgrade]# ls
collector-eft-setup-latest.sh  Collector-latest  GNMIAdapter-latest
GNMIAdapter-yang-modules-latest.tar.gz
[root@cvp collector_upgrade]#
[root@cvp collector_upgrade]# chmod 755 collector-eft-setup-latest.sh

```

Important hint:

When trying to start the script and you get an error message **“No such file or directory”** maybe you might have to convert it to a correct Linux format!

```

[root@cvp collector_upgrade]# sudo ./collector-eft-setup-latest.sh -u
: No such file or directory
[root@cvp collector_upgrade]# yum install dos2unix
Loaded plugins: fastestmirror
Determining fastest mirrors
 * base: centosmirror.netcup.net
 * extras: centosmirror.netcup.net
 * updates: ftp.hosteurope.de
base
| 3.6 kB 00:00:00
cvpi-local
| 2.9 kB 00:00:00
extras

```

```

| 2.9 kB 00:00:00
updates
| 2.9 kB 00:00:00
(1/5): cvpi-local/primary_db
| 947 kB 00:00:00
(2/5): base/7/x86_64/group_gz
| 165 kB 00:00:00
(3/5): extras/7/x86_64/primary_db
| 153 kB 00:00:00
(4/5): updates/7/x86_64/primary_db
| 5.8 MB 00:00:00
(5/5): base/7/x86_64/primary_db
| 6.0 MB 00:00:01
Resolving Dependencies
--> Running transaction check
---> Package dos2unix.x86_64 0:6.0.3-7.el7 will be installed
--> Finished Dependency Resolution

```

Dependencies Resolved

```

=====
=====
Package Arch Version
Repository Size
=====
=====
Installing:
dos2unix x86_64 6.0.3-7.el7
base 74 k

```

Transaction Summary

```

=====
=====
Install 1 Package

```

```

Total download size: 74 k
Installed size: 190 k
Is this ok [y/d/N]: y
Downloading packages:
warning: /var/cache/yum/x86_64/7/base/packages/dos2unix-6.0.3-7.el7.x86_64.rpm:
Header V3 RSA/SHA256 Signature, key ID f4a80eb5: NOKEY
Public key for dos2unix-6.0.3-7.el7.x86_64.rpm is not installed
dos2unix-6.0.3-7.el7.x86_64.rpm
| 74 kB 00:00:00
Retrieving key from file:///etc/pki/rpm-gpg/RPM-GPG-KEY-CentOS-7
Importing GPG key 0xF4A80EB5:
  Userid      : "CentOS-7 Key (CentOS 7 Official Signing Key) <security@centos.org>"
  Fingerprint: 6341 ab27 53d7 8a78 a7c2 7bb1 24c6 a8a7 f4a8 0eb5
  Package     : centos-release-7-6.1810.2.el7.centos.x86_64 (@cvp-dist-os)
  From        : /etc/pki/rpm-gpg/RPM-GPG-KEY-CentOS-7
Is this ok [y/N]: y
Running transaction check
Running transaction test
Transaction test succeeded
Running transaction
  Installing : dos2unix-6.0.3-7.el7.x86_64
1/1
  Verifying  : dos2unix-6.0.3-7.el7.x86_64

```

1/1

Installed:

dos2unix.x86_64 0:6.0.3-7.e17

Complete!

```
[root@cvp collector_upgrade]# dos2unix collector-eft-setup-latest.sh
```

dos2unix: converting file collector-eft-setup-latest.sh to Unix format ...

Now the script can be started:

```
[root@cvp collector_upgrade]# sudo ./collector-eft-setup-latest.sh -u
```

```
E1213 11:39:19.594915 19689 limits.go:19] WARNING: Component "gnmi-adapter"
doesn't define resource limits.
```

(E) => Enabled

(D) => Disabled

(?) => Zookeeper Down

Action Output

COMPONENT	ACTION	NODE	STATUS
ERROR			
gnmi-adapter	enable	primary	(E) DONE

-

Executing command. This may take a few seconds...

(E) => Enabled

(D) => Disabled

(?) => Zookeeper Down

Action Output

COMPONENT	ACTION	NODE	STATUS
ERROR			
gnmi-adapter	stop	primary	(E) DONE

-

wifimanager-collector	stop	primary	(E) DONE
-----------------------	------	---------	----------

-

Failed to stop gnmi-adapter.service: Unit gnmi-adapter.service not loaded.

Failed to stop collector.service: Unit collector.service not loaded.

GNMIAdapter-yang-modules/

GNMIAdapter-yang-modules/github.com/

GNMIAdapter-yang-modules/github.com/YangModels/

GNMIAdapter-yang-modules/github.com/YangModels/yang/

GNMIAdapter-yang-modules/github.com/YangModels/yang/standard/

GNMIAdapter-yang-modules/github.com/YangModels/yang/standard/ietf/

GNMIAdapter-yang-modules/github.com/YangModels/yang/standard/ietf/RFC/

GNMIAdapter-yang-modules/github.com/YangModels/yang/standard/ietf/RFC/ietf-complex

-types@2011-03-15.yang

GNMIAdapter-yang-modules/github.com/YangModels/yang/standard/ietf/RFC/ietf-netconf

-nmda@2019-01-07.yang

...

GNMIAdapter-yang-modules/github.com/aristanetworks/yang/release/openconfig/models/

lacp/

```
GNMIAadapter-yang-modules/github.com/aristanetworks/yang/release/openconfig/models/  
lACP/arista-lACP-augments.yang  
GNMIAadapter-yang-modules/github.com/aristanetworks/yang/release/openconfig/models/  
lACP/arista-lACP-deviations.yang  
GNMIAadapter-yang-modules/github.com/aristanetworks/yang/release/openconfig/models/  
policy-forwarding/  
GNMIAadapter-yang-modules/github.com/aristanetworks/yang/release/openconfig/models/  
policy-forwarding/arista-srte-deviations.yang  
cp: omitting directory '/cvpi/apps/gnmi-adapter/gnmi-adapter-yang-modules/'  
cp: cannot stat '/usr/local/bin/Collector': No such file or directory  
[root@cvp collector_upgrade]#  
[root@cvp collector_upgrade]# ls  
collector-eft-setup-latest.sh  GNMIAadapter-latest  
GNMIAadapter-yang-modules-latest.tar.gz  
Collector-latest                GNMIAadapter-yang-modules  
[root@cvp collector_upgrade]#
```

The **Collector Config** is located at `/cvpi/conf/collector-config.yaml` and is a **YAML** file with the following example structure:

- **Device: snmp**
Options:
 - address: 192.168.10.10**
 - v: 2c**
 - c: public**
- **Device: snmp**
Options:
 - address: 192.168.10.11**
 - v: 3**
 - l: authPriv**
 - u: snmpuser**
 - a: md5**
 - A: snmpauthcred**
 - x: des**
 - X: snmppriv**
- **Device: openconfig**
Options:
 - address: device3**
 - username: admin**
 - Password: secret**

Configuration of the Collector Config for SNMPv2c:

Creating a file "collector-config.yaml" for SNMPv2c:

```
[root@cvp collector_upgrade]# vi /cvpi/conf/collector-config.yaml
```

```
*****
```

```
- Device: snmp
  Options:
    address: 192.168.10.11
    v: 2c
    c: secretpw
```

```
*****
```

Check Collector Config for SNMPv2c:

```
[root@cvp collector_upgrade]# /usr/local/bin/Collector -mock -configFile
/cvpi/conf/collector-config.yaml -mockFeature
interfaces=/interfaces/interface -mockFeature system=/system/state
-mockFeature lldp=/lldp -mockFeature platform=/components/component
time="2019-12-13T13:46:29Z" level=info msg="Added device 1815N-40191"
caller="*devices.snmp"
time="2019-12-13T13:46:29Z" level=info msg="Mock Collector is running"
All features are supported by all devices:
  interfaces      supported
  system          supported
  lldp            supported
  platform        supported
```

Starting the Collector:

```
root@cvp collector_upgrade]# su cvp
[cvp@cvp collector_upgrade]$ cvpi start gnmi-adapter
```

Executing command. This may take a few seconds...

(E) => Enabled
(D) => Disabled
(?) => Zookeeper Down

Action Output

COMPONENT	ACTION	NODE	STATUS
ERROR			
gnmi-adapter	start	primary	(E) DONE
-			

```
[cvp@cvp collector_upgrade]$ cvpi status gnmi-adapter
```

Executing command. This may take a few seconds...

```
primary components total:1 running:1
```

```
[cvp@cvp collector_upgrade]$ exit
```

```
exit
```

```
[root@cvp collector_upgrade]# systemctl start collector
```

```
[root@cvp collector_upgrade]# systemctl status collector
```

```
? collector.service - Collector Service
```

```
Loaded: loaded (/etc/systemd/system/collector.service; disabled; vendor preset: disabled)
```

```
Active: active (running) since Fri 2019-12-13 13:57:35 UTC; 10s ago
```

```
Main PID: 7549 (Collector)
```

```
Tasks: 15
```

```
Memory: 4.3M
```

```
CGroup: /system.slice/collector.service
```

```
mq7549 /usr/local/bin/Collector -logLevel info -configFile /cvpi/conf/collector-config.yaml
```

```
Dec 13 13:57:35 cvp.demo.com systemd[1]: Started Collector Service.
```

```
Dec 13 13:57:36 cvp.demo.com Collector[7549]: time="2019-12-13T13:57:36Z"
```

```
level=info msg="Added device 1815N-40191" caller="*devices.snmp"
```

```
Dec 13 13:57:36 cvp.demo.com Collector[7549]: time="2019-12-13T13:57:36Z"
```

```
level=info msg="Collector is running"
```

```
[root@cvp collector_upgrade]#
```

The 3rd party system will be shown in the device inventory:

The screenshot shows the ARISTA web interface for device inventory. The 'All Devices > Inventory' page displays a table with the following data:

Device	Status	Model	Software	Streaming Agent	IP Address	MAC Address	Device ID
X440G2-12p	✓	000614-00-14	22.6.1.4	1.6.1-00next-223-g58629d6	Unassigned	00:04:96:a2:9b:21	1815N-40191

The screenshot shows the 'Device Overview' page for the device X440G2-12p. It includes the following details:

- System Details:** Hostname: X440G2-12p, Model: 000614-00-14, Software Version: 22.6.1.4, Uptime: 1 hour, 35 minutes, Management IP: None found, Device ID: 1815N-40191, MAC Address: 00:04:96:a2:9b:21.
- System Status:** Streaming Agent Version: 1.6.1-00next-223-g58629d6, Streaming Agent Mode: N/A, Provisioning Status: Streaming Only, Streaming Status: Active, Streaming Latency: 286 ms, Compliance Status: Compliant.
- Interface Counts:** Ethernet Interfaces: 0, VLAN Interfaces: 0, IP Interfaces: 0, Port Channels: 0.

ARISTA | Dashboards | Explorer | X44002-12p

View Mode: Graphs grouped by device

Metric Type: [Dropdown]

Devices: X44002-12p

Metrics: Clear All

- Environment
 - Ambient Temperature
- Switching
 - MAC Addresses Learned
 - Total VLANs
 - Configured VLANs
- Routing
 - IPv4 Total Route Count
 - IPv6 Total Route Count
 - Multicast Sparse Mode Total Table Size
 - Multicast Static Mode Total Table Size
 - ARP Table Size
 - Hostname
 - Software Version
- CPU
 - CPU Utilization
 - 1-Minute CPU Load Average
 - 5-Minute CPU Load Average
 - 15-Minute CPU Load Average
- Memory
 - Buffers Memory
 - Cached Memory
 - Active Memory
 - Used Memory
 - Available Memory
- Hardware Status
 - Discovery Agent Version

Metric	Dec 14, 2019 19:00	Dec 15, 2019 19:00	Dec 16, 2019 19:00
Ambient Temperature	N/A	N/A	N/A
MAC Addresses Learned	N/A	N/A	N/A
Total VLANs	N/A	N/A	N/A
Configured VLANs	N/A	N/A	N/A
IPv4 Total Route Count	N/A	N/A	N/A
IPv6 Total Route Count	N/A	N/A	N/A
Multicast Sparse Mode Total Table Size	N/A	N/A	N/A
Multicast Static Mode Total Table Size	N/A	N/A	N/A
ARP Table Size	N/A	N/A	N/A
Hostname	X44002-12p		
Software Version	22.6.1-1		
CPU Utilization	N/A	N/A	N/A
1-Minute CPU Load Average	N/A	N/A	N/A
5-Minute CPU Load Average	N/A	N/A	N/A
15-Minute CPU Load Average	N/A	N/A	N/A
VLAN Interfaces	N/A	N/A	N/A
Port Channels	N/A	N/A	N/A
VLAN Interfaces	N/A	N/A	N/A
Port Channels	N/A	N/A	N/A
IP Interfaces	N/A	N/A	N/A
Total Power Applied	N/A	N/A	N/A
Total Power Drawn	N/A	N/A	N/A
Total Power Output	N/A	N/A	N/A

Dec 13, 2019 19:00:00 - Now

Show Legend: 1h, 2h, 3h, 6h, 12h, 1d, 3d, 7d, 14d, 30d, 90d, 180d, 360d, 720d, 1440d, 2880d, 5760d, 11520d, 23040d, 46080d, 92160d, 184320d, 368640d, 737280d, 1474560d, 2949120d, 5898240d, 11796480d, 23592960d, 47185920d, 94371840d, 188743680d, 377487360d, 754974720d, 1509949440d, 3019898880d, 6039797760d, 12079595520d, 24159191040d, 48318382080d, 96636764160d, 193273528320d, 386547056640d, 773094113280d, 1546188226560d, 3092376453120d, 6184752906240d, 12369505812480d, 24739011624960d, 49478023249920d, 98956046499840d, 197912092999680d, 395824185999360d, 791648371998720d, 1583296743997440d, 3166593487994880d, 6333186975989760d, 12666373951979520d, 25332747903959040d, 50665495807918080d, 101330991615836160d, 202661983231672320d, 405323966463344640d, 810647932926689280d, 1621295865853378560d, 3242591731706757120d, 6485183463413514240d, 12970366926827028480d, 25940733853654056960d, 51881467707308113920d, 103762935414616227840d, 207525870829232455680d, 415051741658464911360d, 830103483316929822720d, 1660206966633859645440d, 3320413933267719290880d, 6640827866535438581760d, 13281655733070877163520d, 26563311466141754327040d, 53126622932283508654080d, 106253245864567017308160d, 212506491729134034616320d, 425012983458268069232640d, 850025966916536138465280d, 1700051933833072276930560d, 3400103867666144553861120d, 6800207735332289107722240d, 13600415470664578215444480d, 27200830941329156430888960d, 54401661882658312861777920d, 108803323765316625723555840d, 217606647530633251447111680d, 435213295061266502894223360d, 870426590122533005788446720d, 1740853180245066011576893440d, 3481706360490132023153786880d, 6963412720980264046307573760d, 13926825441960528092615147520d, 27853650883921056185230295040d, 55707301767842112370460590080d, 111414603535684224740921180160d, 222829207071368449481842360320d, 445658414142736898963684720640d, 891316828285473797927369441280d, 1782633656570947595854738822560d, 3565267313141895191709477645120d, 7130534626283790383418955290240d, 14261069252567580766837910580480d, 28522138505135161533675821160960d, 57044277010270323067351642321920d, 114088554020540646134703284643840d, 228177108041081292269406569287680d, 456354216082162584538813138575360d, 912708432164325169077626277150720d, 1825416864328650338155252554301440d, 3650833728657300676310505108602880d, 7301667457314601352621010217205760d, 14603334914629202705242020434411520d, 29206669829258405410484040868823040d, 58413339658516810820968081737646080d, 116826679317033621641936163475292160d, 233653358634067243283872326950584320d, 467306717268134486567744653901168640d, 934613434536268973135489307802337280d, 1869226869072537946270978615604674560d, 3738453738145075892541957231209349120d, 7476907476290151785083914462418698240d, 14953814952580303570167828924837396480d, 29907629905160607140335657849674792960d, 59815259810321214280671315699349585280d, 119630519620642428561342631398691171520d, 239261039241284857122685262797382343040d, 478522078482569714245370525594764686080d, 957044156965139428490741051189529372160d, 1914088313930278856981482102379058744320d, 382817662786055771396296420475811748640d, 765635325572111542792592840951623497280d, 1531270651144223085585185681903246994560d, 3062541302288446171170371363806493989120d, 6125082604576892342340742727612987978240d, 12250165209153784684681485455225975956480d, 24500330418307569369362970910451951912960d, 49000660836615138738725941820903903825920d, 98001321673230277477451883641807807651840d, 196002643346460554954903767283615615303680d, 39200528669292110990980753456723122607040d, 78401057338584221981961506913446245214080d, 15680211467716844396392301382689248428160d, 31360422935433688792784602765378496856320d, 62720845870867377585569205530756993712640d, 125441691741734755171138411061513987425280d, 25088338348346951034227682212302797445120d, 50176676696693902068455364424605594890240d, 10035335339338780413691072884921118980480d, 20070670678677560827382145769842237960960d, 40141341357355121654764291539684475921920d, 80282682714710243309528583079368951843840d, 160565365429420486619057166158737903687680d, 321130730858840973238114332317475807375360d, 64226146171768194647622866463495161474720d, 128452292343536389295245732926990322949440d, 256904584687072778590491465853980645898880d, 513809169374145557180982931707961291797760d, 1027618338748291114361965863415922583595520d, 2055236677496582228723931726831845167191040d, 4110473354993164457447863453663690334382080d, 8220946709986328914895726907327380668744320d, 16441893419972657829791453814654761337488640d, 32883786839945315659582907629309522674977280d, 65767573679890631319165815258619045349954560d, 131535147359781262638331630517238090699909120d, 263070294719562525276663261034476181399818240d, 526140589439125050553326522068952362799636480d, 1052281178878250101106653044137904725599272960d, 2104562357756500202213306088275809451198545920d, 4209124715513000404426612176551618902397091840d, 8418249431026000808853224353103237804794183680d, 16836498862052001617706448706206475609588367360d, 33672997724104003235412897412412951219176734720d, 67345995448208006470825794824825902438353469440d, 134691990896416012941651589649651804868706938880d, 269383981792832025883303179299303609737413877760d, 538767963585664051766606358598607219474827755520d, 1077535927171328103533212717197214438949655511040d, 2155071854342656207066425434394428877899311022080d, 4310143708685312414132850868788857757988622044160d, 8620287417370624828265701737577715155977244088320d, 17240574834741249656531403475155431111954488176640d, 3448114966948249931306280695031086222390897633280d, 6896229933896499862612561390062172444817952666560d, 13792459867792999725225122780124344889639153333280d, 27584919735585999450450245560248689779278266666560d, 5516983947117199890090049112049377557555653333280d, 11033967894234399780180098224098751115111106666560d, 22067935788468799560360196448197502230222213333120d, 4413587157693759912072039289639500446044442666240d, 8827174315387519824144078579279000892088885332480d, 17654348630775039648288157558558001784177770664960d, 35308697261550079296576315117116003568355541329920d, 70617394523100158593152630234232007136711088659840d, 141234789046200317186305260468464014274221777319680d, 282469578092400634372610520936928028548443554639360d, 564939156184801268745221041873856057096887109278720d, 1129878312369602537490442083747712114193774215557440d, 2259756624739205074980884167495424228387544431114880d, 451951324947841014996176833499084845677508886223360d, 903902649895682029992353666998169691355017772446720d, 1807805299791364059984707333996339382710035544893440d, 3615610599582728119969414667992678765420071099786880d, 7231221199165456239938829335985357530840142199573760d, 14462443992330912479877658671970715061680284399147520d, 289248879846618249597553173439414301223656879923520d, 578497759693236499195106346878828602447313759847040d, 1156995519386472998390212693777652048894675197694080d, 2313991038772945996780425387555304097789350395388160d, 4627982077545891993560850775110608195578700790776320d, 925596415509178398712170155022121139115774017815520d, 185119283101835679742434031004424227823154435631040d, 370238566203671359484868062008848455646308871262080d, 740477132407342718969736124017696911292617742524160d, 148095426481468537793947224803539382558335485048320d, 296190852962937075587894449607078765116670970096640d, 59238170592587415117578889921415753023334118113280d, 118476341185174830235157779842831506666668222226560d, 236952682370349660470315559685663013333336444453120d, 473905364740699320940631119371326026666672888906240d, 947810729481398641881262238742652053333345777812480d, 18956214589627972837625247774853040666669155556480d, 37912429179255945675250495549706081333338311112960d, 75824858358511891350500991099412162666666622225920d, 151649716717023782701001982198824325333332444451840d, 30329943343404756540200396439764865066664888903360d, 60659886686809513080400792879529730133339777806720d, 121319773373619026160801585759059460266679555613440d, 242639546747238052321603171518118920533359111226880d, 48527909349447610464320634303623780666782224455360d, 97055818698895220928641268607247561333464448910720d, 194111637397790441857282537214495122668888888821440d, 388223274795580883714565074428982253337777777642880d, 776446549591161767429130148857964506675555555285760d, 1552893099182323534858260297715929013345111111571520d, 3105786198364647069716520595431858026682222223443840d, 621157239672929413943304119086371605336444446887680d, 1242314479345858827886608238172743210668888893775360d, 2484628958691717655773216476345462421337777787550720d, 4969257917383435311546432952690924826675555751101440d, 9938515834766870623092865905381849653351111502202880d, 19877031669533741246185731810763709306702223004405760d, 39754063339067482492371463621527418613404446008811520d, 7950812667813496498474292724305483722680889201763840d, 15901625335626992996948585448610967445361778403527680d, 31803250671253985993897170897221934890723556807055360d, 6360650134250797198779434179544386978144711136141120d, 1272130026850159439755886835908877576289422272282240d, 2544260053700318879511773671817751544578844544564480d, 5088520107400637759023547343635503089157788889112960d, 10177040214801275518047094687271006178355577778257920d, 203540804296025510360941893745420123767111555555151520d, 407081608592051020721883787490840247534223111110303040d, 814163217184102041443767574981604451068444222220606080d, 16283264343682040828875351499632089021368888444441212160d, 32566528687364081657750702999264178273777788888882424320d, 65133057374728163315501405998523565547555577777764448640d, 1302661147494563266310028119970471310951111555555512897280d, 260532229498912653262005623994084221800222311111125794560d, 521064458997825306524011247988164443600444222222515589120d, 10421289179956506130480224959763288872008884444451117171840d, 208425783599130122609604499195265776160177788888922343360d, 416851567198260245219208998390531543203555777777844686720d, 833703134396520490438417996781063086407111555555519373440d, 166740626879304098087683599356212617281422311111138746880d, 333481253758608196175367198712425234562844422222277493760d, 666962507517216392350734397424850471136888844444554977280d, 133392501503443278470146879484970094273777888889109954560d, 266785003006886556940293759369940188547555777777821911040d, 533570006013773113880587518739880377095111555555543822080d, 106714001202754622776117503747976154190222311111187644160d, 213428002405509245552235007495952308380444222222375288320d, 426856004811018491104470014991904616676888444445549776640d, 8537120096220369822089400299838123333377788889109954560d, 17074240192440739644178800599676466667555777777821911040d, 34148480384881479288357601199332933335111555555543822080d, 682969607697629585767152023986656667022311111187644160d, 1365939215395259171534304047973313334223111555555519373440d, 2731878430790518343068608095946666844445549776640d, 5463756861581036686137216191893333688889109954560d, 10927513723162073372274432383786667777788889109954560d, 218550274463241467445488647675733355555777777821911040d, 43710054892648

Configuration of the Collector Config for SNMPv3

Creating a file "collector-config.yaml" for SNMPv3:

```
[root@cvp collector_upgrade]# vi /cvpi/conf/collector-config.yaml
```

```
*****
```

```
- Device: snmp
  Options:
    address: 192.168.10.11
    v: 3
    l: authPriv
    u: snmpuser
    a: md5
    A: snmpauthcred
    x: des
    X: snmprivcred
```

```
*****
```

Check Collector Config for SNMPv3:

```
[root@cvp collector_upgrade]# /usr/local/bin/Collector -mock -configFile
/cvpi/conf/collector-config.yaml -mockFeature interfaces=/interfaces/interface
-mockFeature system=/system/state -mockFeature lldp=/lldp
dp -mockFeature platform=/components/component
time="2019-12-13T15:13:44Z" level=info msg="Added device 1815N-40191"
caller="*devices.snmp"
time="2019-12-13T15:13:44Z" level=info msg="Mock Collector is running"
All features are supported by all devices:
  interfaces      supported
  system          supported
  lldp            supported
  platform        supported
```

Starting the Collector:

```
root@cvp collector_upgrade]# su cvp
[cvp@cvp collector_upgrade]$ cvpi start gnmi-adapter
```

Executing command. This may take a few seconds...

(E) => Enabled
(D) => Disabled
(?) => Zookeeper Down

Action Output

COMPONENT	ACTION	NODE	STATUS
ERROR			
gnmi-adapter	start	primary	(E) DONE
-			

```
[cvp@cvp collector_upgrade]$
```

```
[cvp@cvp collector_upgrade]$ cvpi status gnmi-adapter
```

Executing command. This may take a few seconds...

```
primary          components total:1 running:1
```

```
[cvp@cvp collector_upgrade]$ exit
```

```
exit
```

```
[root@cvp collector_upgrade]# systemctl start collector
```

```
[root@cvp collector_upgrade]# systemctl status collector
```

```
? collector.service - Collector Service
```

```
Loaded: loaded (/etc/systemd/system/collector.service; disabled; vendor preset: disabled)
```

```
Active: active (running) since Fri 2019-12-13 15:18:44 UTC; 12s ago
```

```
Main PID: 22823 (Collector)
```

```
Tasks: 14
```

```
Memory: 5.0M
```

```
CGroup: /system.slice/collector.service
```

```
mq22823 /usr/local/bin/Collector -logLevel info -configFile /cvpi/conf/collector-config.yaml
```

```
Dec 13 15:18:44 cvp.demo.com systemd[1]: Started Collector Service.
```

```
Dec 13 15:18:44 cvp.demo.com Collector[22823]: time="2019-12-13T15:18:44Z"
```

```
level=info msg="Added device 1815N-40191" caller="*devices.snmp"
```

```
Dec 13 15:18:44 cvp.demo.com Collector[22823]: time="2019-12-13T15:18:44Z"
```

```
level=info msg="Collector is running"
```

```
[root@cvp collector_upgrade]#
```

The 3rd party system will be shown in the device inventory:

The screenshot shows the ARISTA web interface for the 'Devices' section. The 'Inventory' tab is active, displaying a table with one device entry. The table columns are: Device, Status, Model, Software, Streaming Agent, IP Address, MAC Address, and Device ID. The device entry is X440G2-12p, with a status of 'Active', model 80614-00-14, software version 22.6.1.4, streaming agent 1.6.1-00ne-x223-g56c2968, IP address 'Unassigned', MAC address 00:04:96:a2:9b:21, and device ID 1815N-40191.

Device	Status	Model	Software	Streaming Agent	IP Address	MAC Address	Device ID
X440G2-12p	✓	80614-00-14	22.6.1.4	1.6.1-00ne-x223-g56c2968	Unassigned	00:04:96:a2:9b:21	1815N-40191

The screenshot shows the ARISTA web interface for the 'Devices' section, specifically the 'Device Overview' page for the device X440G2-12p. The page is divided into three sections: System Details, System Status, and Interface Counts. The System Details section shows a device icon and the following information: Hostname: X440G2-12p, Model: 80614-00-14, Software Version: 22.6.1.4, Uptime: 22 minutes, Management IP: None found, Device ID: 1815N-40191, MAC Address: 00:04:96:a2:9b:21. The System Status section shows: Streaming Agent Version: 1.6.1-00ne-x223-g56c2968, Streaming Agent Mode: N/A, Provisioning Status: Streaming Only, Streaming Status: Active, Streaming Latency: 375 ms, Compliance Status: Compliant. The Interface Counts section shows: Ethernet Interfaces: 0, VLAN Interfaces: 0, IP Interfaces: 0, Port Channels: 0.

ARISTA | Devices | Events | Provisioning | Metrics | CloudTracer | Topology | Tap-Agg | admin

Dashboard Explorer

View Mode: X44002-12p

Metric Type: X44002-12p

Devices: X44002-12p

Metrics: Clear All

- Environment
 - Ambient Temperature
- Switching
 - MAC Addresses Learned
 - Total VLANs
 - Configured VLANs
- Routing
 - Pv4 Total Route Count
 - Pv6 Total Route Count
 - Multicast Sparse Mode Total Table Size
 - Multicast Static Mode Total Table Size
 - ARP Table Size
- Device Details
 - Hostname
 - Software Version
- CPU
 - CPU Utilization
 - 1-Minute CPU Load Average
 - 5-Minute CPU Load Average
 - 15-Minute CPU Load Average
- Memory
 - Buffer Memory
 - Cached Memory
 - Active Memory
 - Used Memory
 - Available Memory
- Telemetry Status
 - Streaming Agent Version

Metric	Dec 14, 2019 12:00	Dec 15, 2019 12:00	Dec 16, 2019 12:00
Ambient Temperature			
MAC Addresses Learned			
Total VLANs			
Configured VLANs			
Pv4 Total Route Count			
Pv6 Total Route Count			
Multicast Sparse Mode Total Table Size			
Multicast Static Mode Total Table Size			
ARP Table Size			
Hostname			
Software Version			
CPU Utilization	22.61%		
1-Minute CPU Load Average			
5-Minute CPU Load Average			
15-Minute CPU Load Average			
VLAN Interfaces			
Port Channels			
VLAN Interfaces			
Port Channels			
Total Power Approved			
Total Power Granted			
Total Power Output			

Dec 13, 2019 17:01:01 | New | Show Last: 1h 20m 60s

Add View | Save Dashboard

CloudVision ZTP based automatic onboarding

Enabling ZTP Process on EOS devices:

By default ZTP is **enabled** on all Arista systems!

In case it has been disabled you can re-enable it:

```
switch-1#dir
Directory of flash:/

-rw-          7854          Dec 5 08:27  AsuFastPktTransmit.log
drwx         4096          Oct 4 10:56  Fossil
-rw-         2125          Dec 5 11:35  MLAG
-rw-         4828          Dec 5 08:27  SsuRestore.log
-rw-         4828          Dec 5 08:27  SsuRestoreLegacy.log
-rw-          743          Nov 29 10:07  ansible.pub
-rw-          24          Sep 26 22:56  boot-config
-rw-       6136506          Dec 4 10:55  boot-extensions
drwx         4096          Nov 29 16:12  debug
drwx         4096          Oct 4 10:56  fastpkttx.backup
drwx        16384          Sep 26 22:56  lost+found
drwx         4096          Dec 5 11:38  persist
drwx         4096          Oct 4 11:01  schedule
-rw-         2194          Dec 4 11:58  startup-config
-rw-    418919380          Sep 26 22:56  vEOS-lab.swi
-rw-          13          Oct 4 12:37  zerotouch-config
```

4093313024 bytes total (3211153408 bytes free)

```
switch-1#more zerotouch-config
```

```
DISABLE=True
```

```
switch-1#bash
```

Arista Networks EOS shell

```
[admin@switch-1 ~]$ ls /mnt/flash
AsuFastPktTransmit.log  boot-config          schedule
Fossil                  boot-extensions     startup-config
MLAG                    debug                vEOS-lab.swi
SsuRestore.log          fastpkttx.backup    zerotouch-config
SsuRestoreLegacy.log   lost+found
ansible.pub             persist
[admin@switch-1 ~]$ vi /mnt/flash/zerotouch-config
```

```
*****
```

```
DISABLE=False
```

```
~
```

```
~
```

```
*****
```

```
[admin@switch-1 ~]$ exit
logout
switch-1#more zerotouch-config
DISABLE=False
switch-1#write erase
```

```
Proceed with erasing startup configuration? [confirm]y  
cleanup None None  
switch-1#reload now
```

Setting up an DHCP Server

The ZTP process relies on a **DHCP server** which we install separately within this example on an **Ubuntu Server 18.04.3**

Please make sure that a DNS Server is configured on the Ubuntu Server (e.g. 8.8.8.8 from Google)!

For older Ubuntu versions:

```
root@ubuntu18:/etc/network# nano interfaces
```

```
*****
GNU nano 2.9.3                                     interfaces
Modified

# This file describes the network interfaces available on your system
# and how to activate them. For more information, see interfaces(5).

source /etc/network/interfaces.d/*

# The loopback network interface
auto lo
iface lo inet loopback

# The primary network interface
auto enp0s8
iface enp0s8 inet static
    address 192.168.10.180
    netmask 255.255.255.0
    gateway 192.168.10.1
    dns-nameservers 8.8.8.8

*****
```

For newer Ubuntu Server Versions you will have to edit a YAML File:

```
root@ubuntu18:/etc/netplan# ls
50-cloud-init.yaml
root@ubuntu18:/etc/netplan# nano 50-cloud-init.yaml
```

```
*****

# This file is generated from information provided by
# the datasource. Changes to it will not persist across an instance.
# To disable cloud-init's network configuration capabilities, write a file
# /etc/cloud/cloud.cfg.d/99-disable-network-config.cfg with the following:
# network: {config: disabled}
network:
  ethernets:
    ens160:
      dhcp4: no
      addresses:
        - 192.168.10.180/24
      gateway4: 192.168.10.1
```

```
nameservers:
    addresses: [8.8.8.8, 10.83.29.222]
version: 2
```

Installing an DHCP Server:

```
mkrausch@ubuntu18:~$ sudo su
[sudo] password for mkrausch:password
root@ubuntu18:/home/mkrausch# apt-get install isc-dhcp-server
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following packages were automatically installed and are no longer required:
  libatkmm-1.6-1v5 libcairomm-1.0-1v5 libgtkmm-3.0-1v5 libnih-dbus1
  libpangomm-1.4-1v5
Use 'sudo apt autoremove' to remove them.
The following additional packages will be installed:
  libirs-export160 libiscfg-export160
Suggested packages:
  isc-dhcp-server-ldap policycoreutils
The following NEW packages will be installed:
  isc-dhcp-server libirs-export160 libiscfg-export160
0 upgraded, 3 newly installed, 0 to remove and 103 not upgraded.
Need to get 509 kB of archives.
After this operation, 1,791 kB of additional disk space will be used.
Do you want to continue? [Y/n] y
Get:1 http://de.archive.ubuntu.com/ubuntu bionic-updates/main amd64
libiscfg-export160 amd64 1:9.11.3+dfsg-1ubuntu1.11 [45.4 kB]
Get:2 http://de.archive.ubuntu.com/ubuntu bionic-updates/main amd64
libirs-export160 amd64 1:9.11.3+dfsg-1ubuntu1.11 [18.4 kB]
Get:3 http://de.archive.ubuntu.com/ubuntu bionic-updates/main amd64
isc-dhcp-server amd64 4.3.5-3ubuntu7.1 [446 kB]
Fetched 509 kB in 0s (1,897 kB/s)
Preconfiguring packages ...
Selecting previously unselected package libiscfg-export160.
(Reading database ... 147440 files and directories currently installed.)
Preparing to unpack .../libiscfg-export160_1%3a9.11.3+dfsg-1ubuntu1.11_amd64.deb
...
Unpacking libiscfg-export160 (1:9.11.3+dfsg-1ubuntu1.11) ...
Selecting previously unselected package libirs-export160.
Preparing to unpack .../libirs-export160_1%3a9.11.3+dfsg-1ubuntu1.11_amd64.deb ...
Unpacking libirs-export160 (1:9.11.3+dfsg-1ubuntu1.11) ...
Selecting previously unselected package isc-dhcp-server.
Preparing to unpack .../isc-dhcp-server_4.3.5-3ubuntu7.1_amd64.deb ...
Unpacking isc-dhcp-server (4.3.5-3ubuntu7.1) ...
Setting up libiscfg-export160 (1:9.11.3+dfsg-1ubuntu1.11) ...
Setting up libirs-export160 (1:9.11.3+dfsg-1ubuntu1.11) ...
Setting up isc-dhcp-server (4.3.5-3ubuntu7.1) ...
Generating /etc/default/isc-dhcp-server...
Created symlink
/etc/systemd/system/multi-user.target.wants/isc-dhcp-server.service ?
/lib/systemd/system/isc-dhcp-server.service.
Created symlink
/etc/systemd/system/multi-user.target.wants/isc-dhcp-server6.service ?
```

```

/lib/systemd/system/isc-dhcp-server6.service.
Processing triggers for libc-bin (2.27-3ubuntu1) ...
Processing triggers for systemd (237-3ubuntu10.31) ...
Processing triggers for man-db (2.8.3-2ubuntu0.1) ...
Processing triggers for ureadahead (0.100.0-21) ...
root@ubuntu18:/home/mkrausch#

```

Configuring the DHCP Server:

Please review first on what interface the DHCP Server will listen for DHCP requests!

```

root@ubuntu18:/home/mkrausch# ifconfig
enp0s8: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 192.168.10.180 netmask 255.255.255.0 broadcast 192.168.10.255
    inet6 fe80::a00:27ff:fe9:10b2 prefixlen 64 scopeid 0x20<link>
    ether 08:00:27:f9:10:b2 txqueuelen 1000 (Ethernet)
    RX packets 8506 bytes 7996523 (7.9 MB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 2487 bytes 292320 (292.3 KB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
    inet 127.0.0.1 netmask 255.0.0.0
    inet6 ::1 prefixlen 128 scopeid 0x10<host>
    loop txqueuelen 1000 (Local Loopback)
    RX packets 128 bytes 9731 (9.7 KB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 128 bytes 9731 (9.7 KB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
root@ubuntu18:/home/mkrausch# nano /etc/default/isc-dhcp-server

```

```

*****

GNU nano 2.9.3 /etc/default/isc-dhcp-server
Modified

# Defaults for isc-dhcp-server (sourced by /etc/init.d/isc-dhcp-server)

# Path to dhcpd's config file (default: /etc/dhcp/dhcpd.conf).
DHCPDv4_CONF=/etc/dhcp/dhcpd.conf
#DHCPDv6_CONF=/etc/dhcp/dhcpd6.conf

# Path to dhcpd's PID file (default: /var/run/dhcpd.pid).
DHCPDv4_PID=/var/run/dhcpd.pid
#DHCPDv6_PID=/var/run/dhcpd6.pid

# Additional options to start dhcpd with.
# Don't use options -cf or -pf here; use DHCPD_CONF/ DHCPD_PID instead
#OPTIONS=""

# On what interfaces should the DHCP server (dhcpd) serve DHCP requests?
# Separate multiple interfaces with spaces, e.g. "eth0 eth1".
INTERFACESv4="enp0s8"
INTERFACESv6=""

*****

```

The option "bootfile-name" must include the IP address of CloudVision:

```
root@ubuntu18:/home/mkrausch# nano /etc/dhcp/dhcpd.conf
```

```
*****

# dhcpd.conf
#
# Sample configuration file for ISC dhcpd
#
# Attention: If /etc/ltsp/dhcpd.conf exists, that will be used as
# configuration file instead of this file.
#

# option definitions common to all supported networks...
# option domain-name "example.org";
# option domain-name-servers ns1.example.org, ns2.example.org;

default-lease-time 600;
max-lease-time 7200;

# The ddns-updates-style parameter controls whether or not the server will
# attempt to do a DNS update when a lease is confirmed. We default to the
# behavior of the version 2 packages ('none', since DHCP v2 didn't
# have support for DDNS.)
ddns-update-style none;

# If this DHCP server is the official DHCP server for the local
# network, the authoritative directive should be uncommented.
authoritative;

# Use this to send dhcp log messages to a different log file (you also
# have to hack syslog.conf to complete the redirection).
#log-facility local7;

# No service will be given on this subnet, but declaring it helps the
# DHCP server to understand the network topology.

#subnet 10.152.187.0 netmask 255.255.255.0 {
#}

# This is a very basic subnet declaration.

subnet 192.168.10.0 netmask 255.255.255.0 {
    range 192.168.10.70 192.168.10.90;
    interface enp0s8;
    option routers 192.168.10.1;
    option bootfile-name "http://192.168.10.60/ztp/bootstrap";
}

# This declaration allows BOOTP clients to get dynamic addresses,
# which we don't really recommend.
...

*****
```

Starting & Stopping the DHCP Server

Starting the DHCP Server:

```
root@ubuntu18:/home/mkrausch# service isc-dhcp-server start
root@ubuntu18:/home/mkrausch# service isc-dhcp-server status
? isc-dhcp-server.service - ISC DHCP IPv4 server
   Loaded: loaded (/lib/systemd/system/isc-dhcp-server.service; enabled; vendor
   preset: enabled)
   Active: active (running) since Fri 2019-12-06 11:07:09 CET; 3s ago
     Docs: man:dhcpd(8)
  Main PID: 3185 (dhcpd)
    Tasks: 1 (limit: 4915)
   CGroup: /system.slice/isc-dhcp-server.service
           mq3185 dhcpd -user dhcpd -group dhcpd -f -4 -pf
           /run/dhcp-server/dhcpd.pid -cf /etc/dhcp/dhcpd.conf
```

```
Dec 06 11:07:09 ubuntu18 sh[3185]: PID file: /run/dhcp-server/dhcpd.pid
Dec 06 11:07:09 ubuntu18 dhcpd[3185]: Wrote 1 leases to leases file.
Dec 06 11:07:09 ubuntu18 sh[3185]: Wrote 1 leases to leases file.
Dec 06 11:07:09 ubuntu18 dhcpd[3185]: Listening on
LPF/enp0s8/08:00:27:f9:10:b2/192.168.10.0/24
Dec 06 11:07:09 ubuntu18 sh[3185]: Listening on
LPF/enp0s8/08:00:27:f9:10:b2/192.168.10.0/24
Dec 06 11:07:09 ubuntu18 sh[3185]: Sending on
LPF/enp0s8/08:00:27:f9:10:b2/192.168.10.0/24
Dec 06 11:07:09 ubuntu18 sh[3185]: Sending on Socket/fallback/fallback-net
Dec 06 11:07:09 ubuntu18 dhcpd[3185]: Sending on
LPF/enp0s8/08:00:27:f9:10:b2/192.168.10.0/24
Dec 06 11:07:09 ubuntu18 dhcpd[3185]: Sending on Socket/fallback/fallback-net
Dec 06 11:07:09 ubuntu18 dhcpd[3185]: Server starting service.
```

Stopping the DHCP Server:

```
root@ubuntu18:/home/mkrausch# service isc-dhcp-server stop
```

EOS & CloudVision ZTP Outputs

EOS Output while booting the system:

...

No startup-config was found.

The device is in Zero Touch Provisioning mode and is attempting to download the startup-config from a remote system. The device will not be fully functional until either a valid startup-config is downloaded from a remote system or Zero Touch Provisioning is cancelled.

To cancel Zero Touch Provisioning, login as admin and type 'zerotouch cancel' at the CLI. Alternatively, to disable Zero Touch Provisioning permanently, type 'zerotouch disable' at the CLI.

Note: The device will reload when these commands are issued.

```
localhost login: Dec 6 10:41:57 localhost ZeroTouch: %ZTP-6-DHCPv4_QUERY: Sending DHCPv4 request on [ Ethernet1, Ethernet2, Ethernet3, Ethernet4, Ethernet5, Ethernet6, Ethernet7, Ethernet8, Ethernet9, Ethernet10, Ethernet11, Ethernet12, Management1 ]
```

```
Dec 6 10:42:00 localhost ZeroTouch: %ZTP-6-DHCPv4_SUCCESS: DHCPv4 response received on Management1 [ Ip Address: 192.168.10.70/24/24; Gateway: 192.168.10.1; Boot File: http://192.168.10.60/ztp/bootstrap ]
```

```
Dec 6 10:42:05 localhost ZeroTouch: %ZTP-6-CONFIG_DOWNLOAD: Attempting to download the startup-config from http://192.168.10.60/ztp/bootstrap
```

```
Dec 6 10:42:05 localhost ZeroTouch: %ZTP-6-CONFIG_DOWNLOAD_SUCCESS: Successfully downloaded config script from http://192.168.10.60/ztp/bootstrap
```

```
Dec 6 10:42:05 localhost ZeroTouch: %ZTP-6-EXEC_SCRIPT: Executing the downloaded config script
```

```
Dec 6 10:42:05 localhost cvpNotifyIntvl = 60
```

```
Dec 6 10:42:05 localhost configPollIntvl = 2
```

```
Dec 6 10:42:05 localhost cvpUrl =
```

```
https://192.168.10.60/cvpservice/services/ztp/config
```

```
Dec 6 10:42:05 localhost cvpUser = cvptemp
```

```
Dec 6 10:42:05 localhost Removing temporary files
```

```
Dec 6 10:42:06 localhost ['192.168.10.70']
```

```
Dec 6 10:42:06 localhost Could not resolve ip 192.168.10.70 to a host name
```

```
Dec 6 10:42:06 localhost hostname sw-192.168.10.70 ipAdrrs ['192.168.10.70']
```

```
Dec 6 10:42:06 localhost Applying Base EOS configuration
```

```
Dec 6 10:44:53 sw-192 upgrading TerminAttr
```

```
Dec 6 10:44:53 sw-192 check space available
```

```
Dec 6 10:44:53 sw-192 freeSz : 3063.000000 imageSz :5.900000
```

```
Dec 6 10:44:53 sw-192 free space available..nothing to do
```

```
Dec 6 10:44:53 sw-192 running commands enable#012show module | json errMsg Command 'echo -e 'enable#012show module | json' | /usr/bin/FastCli' returned non-zero exit status 1
```

```
Dec 6 10:44:53 sw-192 error getting module information '"Command \'echo -e \'enable\\nshow module | json\' | /usr/bin/FastCli\' returned non-zero exit status 1"'
```

```
Dec 6 10:45:24 sw-192 status code=8 msg=None
```

```
Dec 6 10:45:24 sw-192 waiting for Eos configuration
```

```
Dec 6 10:45:24 sw-192 Sending request to
```

```
https://192.168.10.60/cvpservice/services/ztp/config [0]
```

```
Dec 6 10:45:27 sw-192 status code=11 msg=Successfully requested CVP for config
```

No startup-config was found.

The device is in Zero Touch Provisioning mode and is attempting to download the startup-config from a remote system. The device will not be fully functional until either a valid startup-config is downloaded from a remote system or Zero Touch Provisioning is cancelled.

To cancel Zero Touch Provisioning, login as admin and type 'zerotouch cancel' at the CLI. Alternatively, to disable Zero Touch Provisioning permanently, type 'zerotouch disable' at the CLI.

Note: The device will reload when these commands are issued.

```
localhost login: admin
sw-192.168.10.70>en
sw-192.168.10.70#show run
! Command: show running-config
! device: sw-192.168.10.70 (vEOS, EOS-4.22.2F)
!
! boot system flash:/vEOS-lab.swi
!
service configuration session max pending 10
!
daemon TerminAttr
  exec /usr/bin/TerminAttr -ingestgrpcurl=192.168.10.60:9910 -cvcompression=gzip
  -taillogs -ingestauth=key,secretkey
  -smashexcludes=ale,flexCounter,hardware,kni,pulse,strata
  -ingestexclude=/Sysdb/cell/1/agent,/Sysdb/cell/2/agent
  no shutdown
!
switchport default mode routed
!
transceiver qsfp default-mode 4x10G
!
logging console informational
!
logging level AAA errors
logging level ACCOUNTING errors
logging level ACL errors
logging level AGENT errors
logging level ALE errors
logging level ARP errors
logging level BFD errors
logging level BGP errors
logging level BMP errors
logging level CAPACITY errors
logging level CAPI errors
logging level CLEAR errors
logging level CVX errors
logging level DATAPLANE errors
logging level DHCP errors
logging level DOT1X errors
logging level DSCP errors
logging level ENVMON errors
logging level ETH errors
logging level EVENTMON errors
logging level EXTENSION errors
logging level FHRP errors
logging level FLOW errors
```

logging level FORWARDING errors
logging level FRU errors
logging level FWK errors
logging level GMP errors
logging level HARDWARE errors
logging level HEALTH errors
logging level HTTPSERVICE errors

logging level IGMP errors
logging level IGMP Snooping errors
logging level INT errors
logging level INTF errors
logging level IP6ROUTING errors
logging level IPRIB errors
logging level IRA errors
logging level ISIS errors
logging level KERNELFIB errors
logging level LACP errors
logging level LAG errors
logging level LAUNCHER errors
logging level LDP errors
logging level LICENSE errors
logging level LINEPROTO errors
logging level LLDP errors
logging level LOGMGR errors
logging level LOOPBACK errors
logging level LOOPPROTECT errors
logging level MAPREDUCEMONITOR errors
logging level MIRRORING errors
logging level MKA errors
logging level MLAG errors
logging level MMODE errors
logging level MROUTE errors
logging level MRP errors
logging level MSDP errors
logging level MSRP errors
logging level MSSPOLICYMONITOR errors
logging level MVRP errors
logging level NAT errors
logging level OPENCONFIG errors
logging level OPENFLOW errors
logging level OSPF errors
logging level OSPF3 errors
logging level PACKAGE errors
logging level PFC errors
logging level PIMBSR errors
logging level PORTSECURITY errors
logging level PTP errors
logging level PWRMGMT errors
logging level QOS errors
logging level QUEUEMONITOR errors
logging level REDUNDANCY errors
logging level RIB errors
logging level ROUTING errors
logging level SECURITY errors
logging level SERVERMONITOR errors
logging level SPANTREE errors
logging level SSO errors

```

logging level STAGEMGR errors
logging level SYS errors
logging level SYSDB errors
logging level TAPAGG errors
logging level TCP errors
logging level TRANSCEIVER errors
logging level TUNNEL errors
logging level TUNNELINTF errors
logging level VMTRACERSESS errors
logging level VMWAREVI errors
logging level VMWAREVS errors
logging level VRF errors
logging level VRRP errors
logging level VXLAN errors
logging level XMPP errors
logging level ZTP informational
!
hostname sw-192.168.10.70
!
spanning-tree mode mstp
!
no aaa root
!
username cvptemp privilege 15 secret sha512
$6$oFUro2SrkJzjSH1$uyooeSEczwVfhrjgQYQ6Af5ZkBlNK0gkHmdVXLxVChuPfb7P/veb/fdnJoCaY9
hydPsv6fIzr3FmlWMB6DG300
!
interface Ethernet1
    no switchport
    ipv6 enable
    ipv6 address auto-config
    ipv6 nd ra rx accept default-route
!
interface Ethernet2
    no switchport
    ipv6 enable
    ipv6 address auto-config
    ipv6 nd ra rx accept default-route
!
interface Ethernet3
    no switchport
    ipv6 enable
    ipv6 address auto-config
    ipv6 nd ra rx accept default-route
!
interface Ethernet4
    no switchport
    ipv6 enable
    ipv6 address auto-config
    ipv6 nd ra rx accept default-route
!
interface Ethernet5
    no switchport
    ipv6 enable
    ipv6 address auto-config
    ipv6 nd ra rx accept default-route
!
interface Ethernet6

```

```

    no switchport
    ipv6 enable
    ipv6 address auto-config
    ipv6 nd ra rx accept default-route
!
interface Ethernet7
    no switchport
    ipv6 enable
    ipv6 address auto-config
    ipv6 nd ra rx accept default-route
!
interface Ethernet8
    no switchport
    ipv6 enable
    ipv6 address auto-config
    ipv6 nd ra rx accept default-route
!
interface Ethernet9
    no switchport
    ipv6 enable
    ipv6 address auto-config
    ipv6 nd ra rx accept default-route
!
interface Ethernet10
    no switchport
    ipv6 enable
    ipv6 address auto-config
    ipv6 nd ra rx accept default-route
!
interface Ethernet11
    no switchport
    ipv6 enable
    ipv6 address auto-config
    ipv6 nd ra rx accept default-route
!
interface Ethernet12
    no switchport
    ipv6 enable
    ipv6 address auto-config
    ipv6 nd ra rx accept default-route
!
interface Management1
    ip address 192.168.10.70/24
    ipv6 enable
    ipv6 address auto-config
    ipv6 nd ra rx accept default-route
!
ip route 0.0.0.0/0 192.168.10.1
!
no ip routing
!
control-plane
    no service-policy input copp-system-policy
!
banner login
No startup-config was found.
The device is in Zero Touch Provisioning mode and is attempting to
download the startup-config from a remote system. The device will not

```

be fully functional until either a valid startup-config is downloaded from a remote system or Zero Touch Provisioning is cancelled. To cancel Zero Touch Provisioning, login as admin and type 'zerotouch cancel' at the CLI. Alternatively, to disable Zero Touch Provisioning permanently, type 'zerotouch disable' at the CLI. Note: The device will reload when these commands are issued.

EOF

!

```
management api http-commands
  no shutdown
```

!

```
end
```

```
sw-192.168.10.70#
```

CloudVision automatically onboarded ztp enabled devices:

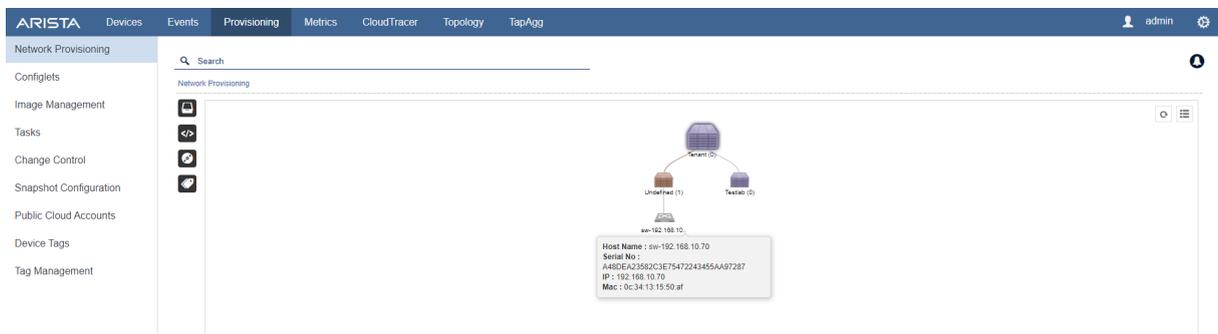


The screenshot shows the ARISTA CloudVision interface. The top navigation bar includes 'ARISTA', 'Devices', 'Events', 'Provisioning', 'Metrics', 'CloudTracer', 'Topology', and 'TapAgg'. The user is logged in as 'admin'. The main content area is titled 'All Devices > Inventory' and shows a search bar with the text 'Showing all 1 device'. Below the search bar is a table with the following data:

Device	Status	Model	Software	Streaming Agent	IP Address	MAC Address	Device ID
sw-192.168.10.70	✓ 🚩	vEOS	4.22.2F	1.6.1	192.168.10.70	0c:34:13:15:50:af	A48DEA23582C3E75472...

Below the table, there is an 'Export to CSV' link and a note 'Showing 1 to 1 of 1 row'.

The new device will automatically be assigned to the Container “**Undefined**” from where you can reassign it to other Containers with predefined Configlets



CloudVision Portal RESTful API Client (CVPRAC)

CloudVision includes a **full-featured RESTful API** that provides all of the same functionality available via the GUI which can be used to automate workflows and integrate with other tools

Prerequisites:

Please make sure that you are meeting the **same prerequisites** as in the chapter “**Remote Python scripting**”!

Installing CVPRAC on Ubuntu Linux

Within this Manual we will be using an external **Ubuntu Server 20.04.3**

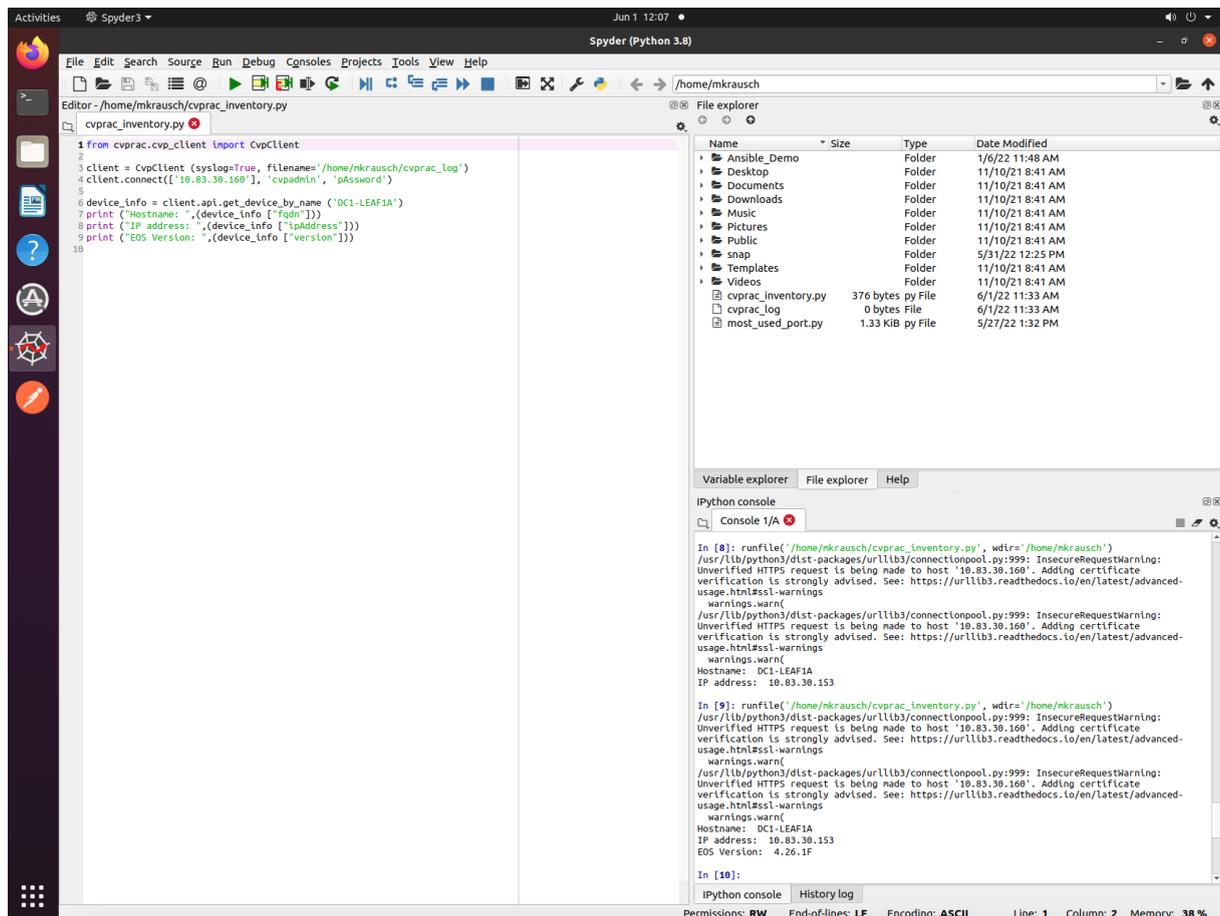
Please make sure that a DNS Server is configured on the Ubuntu Server (e.g. 8.8.8.8 from Google) and that you can reach GitHub!

```
mkrausch@ubuntu:~$ sudo su
[sudo] password for mkrausch:
root@ubuntu:/home/mkrausch# ping www.github.com
PING github.com (192.30.255.113) 56(84) bytes of data.
64 bytes from lb-192-30-255-113-sea.github.com (192.30.255.113): icmp_seq=1
ttl=128 time=157 ms
64 bytes from lb-192-30-255-113-sea.github.com (192.30.255.113): icmp_seq=2
ttl=128 time=157 ms
64 bytes from lb-192-30-255-113-sea.github.com (192.30.255.113): icmp_seq=3
ttl=128 time=157 ms
^C
--- github.com ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2004ms
rtt min/avg/max/mdev = 156.840/157.020/157.290/0.194 ms
root@ubuntu:/home/mkrausch# pip install cvprac
Collecting cvprac
  Downloading cvprac-1.2.0.tar.gz (4.3 MB)
    |████████████████████████████████████████| 4.3 MB 22.4 MB/s
Requirement already satisfied: requests>=1.0.0 in /usr/lib/python3/dist-packages
(from cvprac) (2.22.0)
Building wheels for collected packages: cvprac
  Building wheel for cvprac (setup.py) ... done
  Created wheel for cvprac: filename=cvprac-1.2.0-py3-none-any.whl size=46548
sha256=c64f66e3416e6b52296e4945fba6b1c68a986371c0b8c603be0b5d3840103f85
  Stored in directory:
/root/.cache/pip/wheels/6c/96/53/6f61f628f52c188595f1b0b572b0df8f7ee5797cbd078ff42
b
Successfully built cvprac
Installing collected packages: cvprac
Successfully installed cvprac-1.2.0
```

Example Python Script for reading information

On the installed Ubuntu Desktop you can start e.g. the **Spyder Editor** to create a Python script, edit an existing one or even run the script and watch the outputs in a Console Window.

This quite simple Python script can be run on any remote machine connected to the network of the **CloudVision Management Interface** and displays the Hostname, IP Address and SW Version of an EOS system identified by its hostname without directly connecting to the system:



The screenshot shows the Spyder Python IDE interface. The editor window displays a Python script named `cvprac_inventory.py` with the following code:

```
1 from cvprac.cvp_client import CvpClient
2
3 client = CvpClient (syslog=True, filename='/home/mkrausch/cvprac_log')
4 client.connect(['10.83.30.160'], 'cvpadmin', 'pAssword')
5
6 device_info = client.api.get_device_by_name ('DC1-LEAF1A')
7 print ("Hostname: ",(device_info ["fqdn"]))
8 print ("IP address: ",(device_info ["ipAddress"]))
9 print ("EOS Version: ",(device_info ["version"]))
10
```

The File explorer on the right shows the file structure of the project, including `cvprac_inventory.py`, `cvprac_log`, and `most_used_port.py`. The IPython console at the bottom shows the execution output:

```
In [8]: runfile('/home/mkrausch/cvprac_inventory.py', wdir='/home/mkrausch')
/usr/lib/python3/dist-packages/urllib3/connectionpool.py:999: InsecureRequestWarning:
Unverified HTTPS request is being made to host '10.83.30.160'. Adding certificate
verification is strongly advised. See: https://urllib3.readthedocs.io/en/latest/advanced-
usage.html#ssl-warnings
warnings.warn(
/usr/lib/python3/dist-packages/urllib3/connectionpool.py:999: InsecureRequestWarning:
Unverified HTTPS request is being made to host '10.83.30.160'. Adding certificate
verification is strongly advised. See: https://urllib3.readthedocs.io/en/latest/advanced-
usage.html#ssl-warnings
warnings.warn(
Hostname: DC1-LEAF1A
IP address: 10.83.30.153

In [9]: runfile('/home/mkrausch/cvprac_inventory.py', wdir='/home/mkrausch')
/usr/lib/python3/dist-packages/urllib3/connectionpool.py:999: InsecureRequestWarning:
Unverified HTTPS request is being made to host '10.83.30.160'. Adding certificate
verification is strongly advised. See: https://urllib3.readthedocs.io/en/latest/advanced-
usage.html#ssl-warnings
warnings.warn(
/usr/lib/python3/dist-packages/urllib3/connectionpool.py:999: InsecureRequestWarning:
Unverified HTTPS request is being made to host '10.83.30.160'. Adding certificate
verification is strongly advised. See: https://urllib3.readthedocs.io/en/latest/advanced-
usage.html#ssl-warnings
warnings.warn(
Hostname: DC1-LEAF1A
IP address: 10.83.30.153
EOS Version: 4.26.1F

In [10]:
```

Python Script:

```
from cvprac.cvp_client import CvpClient

client = CvpClient (syslog=True, filename='/home/mkrausch/cvprac_log')
client.connect(['10.83.30.160'], 'cvpadmin', 'pAssword')

device_info = client.api.get_device_by_name ('DC1-LEAF1A')
print ("Hostname: ",(device_info ["fqdn"]))
print ("IP address: ",(device_info ["ipAddress"]))
print ("EOS Version: ",(device_info ["version"]))
```

Output:

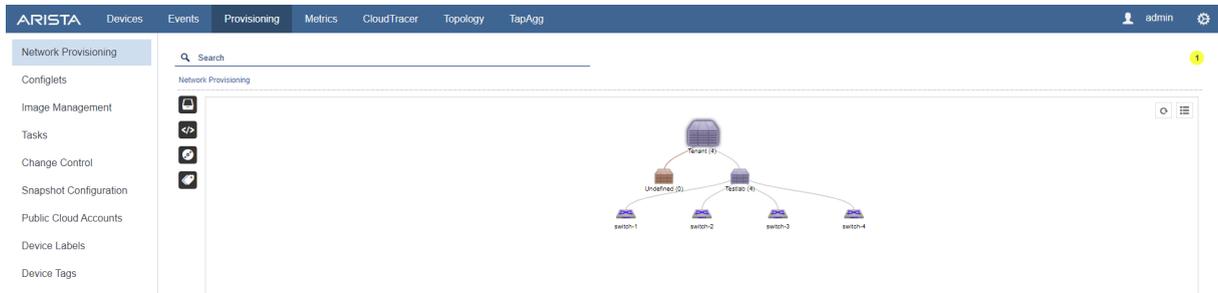
Please ignore the Warnings if no certificate verification is configured

```
root@ubuntu:/home/mkrausch# python3 cvprac_inventory.py
/usr/lib/python3/dist-packages/urllib3/connectionpool.py:999:
InsecureRequestWarning: Unverified HTTPS request is being made to host
'10.83.30.160'. Adding certificate verification is strongly advised. See:
https://urllib3.readthedocs.io/en/latest/advanced-usage.html#ssl-warnings
  warnings.warn(
/usr/lib/python3/dist-packages/urllib3/connectionpool.py:999:
InsecureRequestWarning: Unverified HTTPS request is being made to host
'10.83.30.160'. Adding certificate verification is strongly advised. See:
https://urllib3.readthedocs.io/en/latest/advanced-usage.html#ssl-warnings
  warnings.warn(
Hostname: DC1-LEAF1A
IP address: 10.83.30.153
EOS Version: 4.26.1F
root@ubuntu:/home/mkrausch#
```

Example Python Script creating a new Container within CloudVision

This quite simple Python script can be run on any remote machine connected to the network of the **CloudVision Management Interface** and creates a new Container named “TOR” attached to the already existing Container “Testlab”

Before creating the new Container:



Python Script:

```
from cvprac.cvp_client import CvpClient

client = CvpClient (syslog=True, filename='/home/mkrausch/cvprac_log')
client.connect(['10.83.30.160'], 'cvpadmin', 'pAssword')

parent = client.api.search_topology('Testlab')
client.api.add_container('TOR', 'Testlab', parent['containerList'][0]['key'])
```

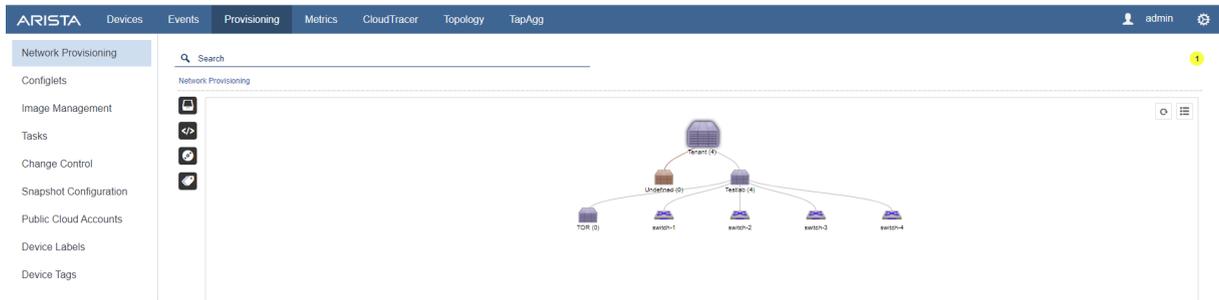
Running the script:

Please ignore the Warnings if no certificate verification is configured

```
root@ubuntu:/home/mkrausch# python3 cvprac_create_container.py
/usr/lib/python3/dist-packages/urllib3/connectionpool.py:999:
InsecureRequestWarning: Unverified HTTPS request is being made to host
'10.83.30.160'. Adding certificate verification is strongly advised. See:
https://urllib3.readthedocs.io/en/latest/advanced-usage.html#ssl-warnings
warnings.warn(
/usr/lib/python3/dist-packages/urllib3/connectionpool.py:999:
InsecureRequestWarning: Unverified HTTPS request is being made to host
'10.83.30.160'. Adding certificate verification is strongly advised. See:
https://urllib3.readthedocs.io/en/latest/advanced-usage.html#ssl-warnings
warnings.warn(
/usr/lib/python3/dist-packages/urllib3/connectionpool.py:999:
InsecureRequestWarning: Unverified HTTPS request is being made to host
'10.83.30.160'. Adding certificate verification is strongly advised. See:
https://urllib3.readthedocs.io/en/latest/advanced-usage.html#ssl-warnings
warnings.warn(
/usr/lib/python3/dist-packages/urllib3/connectionpool.py:999:
InsecureRequestWarning: Unverified HTTPS request is being made to host
'10.83.30.160'. Adding certificate verification is strongly advised. See:
https://urllib3.readthedocs.io/en/latest/advanced-usage.html#ssl-warnings
warnings.warn(
```

root@ubuntu:/home/mkrausch#

Now the new Container “TOR” has been configured and attached to the Container (Testlab):



Configuring MAC based Client Authentication

Arista systems support MAC based authentication an 802.1x based authentication with dynamic VLAN assignments.

Setting up a RADIUS Server

In this example we are using an Ubuntu Server 18.04 to setup **Freeradius**:

```
mkrausch@ubuntu18:~$ sudo su
[sudo] password for mkrausch:
root@ubuntu18:/home/mkrausch# apt-get update
Hit:1 http://de.archive.ubuntu.com/ubuntu bionic InRelease
Hit:2 http://security.ubuntu.com/ubuntu bionic-security InRelease
Hit:3 http://de.archive.ubuntu.com/ubuntu bionic-updates InRelease
Hit:4 http://de.archive.ubuntu.com/ubuntu bionic-backports InRelease
Reading package lists... Done
root@ubuntu18:/home/mkrausch# apt-get install freeradius
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following packages were automatically installed and are no longer required:
  libatkmm-1.6-1v5 libcairomm-1.0-1v5 libgtkmm-3.0-1v5 libnih-dbus1
  libpangomm-1.4-1v5
Use 'sudo apt autoremove' to remove them.
The following additional packages will be installed:
  freeradius-common freeradius-config freeradius-utils freetds-common libct4
  libdbi-perl libfreeradius3 make
Suggested packages:
  freeradius-ldap freeradius-postgresql freeradius-mysql freeradius-krb5 snmp
  libmldb-perl libnet-daemon-perl libsql-statement-perl
  make-doc
The following NEW packages will be installed:
  freeradius freeradius-common freeradius-config freeradius-utils freetds-common
  libct4 libdbi-perl libfreeradius3 make
0 upgraded, 9 newly installed, 0 to remove and 165 not upgraded.
Need to get 2,185 kB of archives.
After this operation, 7,925 kB of additional disk space will be used.
Do you want to continue? [Y/n] y
Get:1 http://de.archive.ubuntu.com/ubuntu bionic-updates/main amd64 freetds-common
all 1.00.82-2ubuntu0.1 [24.7 kB]
Get:2 http://de.archive.ubuntu.com/ubuntu bionic-updates/main amd64 libct4 amd64
1.00.82-2ubuntu0.1 [147 kB]
Get:3 http://de.archive.ubuntu.com/ubuntu bionic-updates/main amd64
freeradius-common all 3.0.16+dfsg-1ubuntu3.1 [214 kB]
Get:4 http://de.archive.ubuntu.com/ubuntu bionic/main amd64 make amd64
4.1-9.1ubuntu1 [154 kB]
Get:5 http://de.archive.ubuntu.com/ubuntu bionic-updates/main amd64
freeradius-config amd64 3.0.16+dfsg-1ubuntu3.1 [150 kB]
Get:6 http://de.archive.ubuntu.com/ubuntu bionic-updates/main amd64 libfreeradius3
amd64 3.0.16+dfsg-1ubuntu3.1 [180 kB]
Get:7 http://de.archive.ubuntu.com/ubuntu bionic-updates/main amd64 freeradius
amd64 3.0.16+dfsg-1ubuntu3.1 [502 kB]
Get:8 http://de.archive.ubuntu.com/ubuntu bionic-updates/main amd64
freeradius-utils amd64 3.0.16+dfsg-1ubuntu3.1 [88.3 kB]
```

```
Get:9 http://de.archive.ubuntu.com/ubuntu bionic/main amd64 libdbi-perl amd64
1.640-1 [724 kB]
Fetched 2,185 kB in 44s (49.5 kB/s)
```

```
...
.....+.....
.....+.....
.....+*+*+*+*+*+*+*
Setting up libdbi-perl (1.640-1) ...
Setting up libct4:amd64 (1.00.82-2ubuntu0.1) ...
Setting up freeradius-utils (3.0.16+dfsg-1ubuntu3.1) ...
Setting up freeradius (3.0.16+dfsg-1ubuntu3.1) ...
update-rc.d: warning: start and stop actions are no longer supported; falling back
to defaults
Processing triggers for systemd (237-3ubuntu10.31) ...
Processing triggers for man-db (2.8.3-2ubuntu0.1) ...
Processing triggers for ureadahead (0.100.0-21) ...
Processing triggers for libc-bin (2.27-3ubuntu1) ...
root@ubuntu18:/home/mkrausch#
```

Configuring RADIUS Clients (Arista systems):

```
root@ubuntu18:/home/mkrausch# cd /etc/freeradius/3.0/
root@ubuntu18:/etc/freeradius/3.0# ls
certs                huntgroups          policy.d            sites-enabled
clients.conf         mods-available      proxy.conf          templates.conf
dictionary           mods-config         radiusd.conf        trigger.conf
experimental.conf    mods-enabled        README.rst          users
hints                panic.gdb           sites-available
root@ubuntu18:/etc/freeradius/3.0# nano clients.conf
```

```
*****
```

```
...
client Arista-1 {
ipaddr = 10.83.28.225
secret = secretpw1
}

client Arista-2 {
ipaddr = 192.168.10.202
secret = password2
}

client Arista-3 {
ipaddr = 192.168.10.5
secret = password3
}
...
*****
```

Configuring Users:

```
root@ubuntu18:/etc/freeradius/3.0# ls
certs                huntgroups          policy.d            sites-enabled
clients.conf         mods-available     proxy.conf         templates.conf
dictionary           mods-config        radiusd.conf       trigger.conf
experimental.conf   mods-enabled       README.rst         users
hints                panic.gdb          sites-available
root@ubuntu18:/etc/freeradius/3.0# nano users
```

```
*****
```

```
...
```

```
00:1c:73:85:ea:3d Auth-Type := Accept
                  Cleartext-Password := "00:1c:73:85:ea:3d",
                  Tunnel-Type:0 := VLAN,
                  Tunnel-Medium-Type := IEEE-802,
                  Tunnel-Private-Group-ID := 50
```

```
00:1c:73:86:56:85 Auth-Type := Accept
                  Cleartext-Password := "00:1c:73:86:56:85",
                  Tunnel-Type:0 := VLAN,
                  Tunnel-Medium-Type := IEEE-802,
                  Tunnel-Private-Group-ID := vlan50
```

```
mkrausch Cleartext-Password := "password"
```

```
...
```

```
*****
```

Restarting the Freeradius Service:

```
root@ubuntu18:/etc/freeradius/3.0# service freeradius restart
```

Testing local Authentication Service on Freeradius:

```
root@ubuntu18:/etc/freeradius/3.0# radtest mkrausch password localhost 0
testing123
Sent Access-Request Id 147 from 0.0.0.0:43369 to 127.0.0.1:1812 length 78
  User-Name = "mkrausch"
  User-Password = "password"
  NAS-IP-Address = 10.83.28.122
  NAS-Port = 0
  Message-Authenticator = 0x00
  Cleartext-Password = "password"
Received Access-Accept Id 147 from 127.0.0.1:1812 to 0.0.0.0:0 length 20
root@ubuntu18:/etc/freeradius/3.0#
```

Setting up Arista EOS systems

Configuring an Arista device for MAC based Authentication & bypassing LLDP based IP Phones:

```
switchport default phone cos 3
!
switchport default phone trunk untagged
!
switchport default phone vlan 75
!
hostname Arista-1
!
radius-server host 10.83.28.244 key secretpw1
radius-server host 10.83.28.123 key secretpw2
!
aaa group server radius MyRadiusServers
    server 10.83.28.244
    server 10.83.28.123
!
aaa authentication dot1x default group MyRadiusServers
!
vlan 10
    name guest
!
vlan 40
    name vlan40
!
vlan 50
    name vlan50
!
vlan 75
    name phone
!
!
interface Ethernet54/1
    speed forced 10000full
    switchport trunk native vlan 10
    switchport mode trunk phone
    spanning-tree portfast
    dot1x pae authenticator
    dot1x authentication failure action traffic allow vlan 10
    dot1x reauthentication
    dot1x port-control auto
    dot1x host-mode multi-host authenticated
    dot1x mac based authentication
    dot1x timeout reauth-period 28800
!
interface Ethernet54/2
    speed forced 10000full
    switchport trunk native vlan 10
    switchport mode trunk phone
    spanning-tree portfast
    dot1x pae authenticator
    dot1x authentication failure action traffic allow vlan 10
    dot1x reauthentication
    dot1x port-control auto
    dot1x host-mode multi-host authenticated
```

```

dot1x mac based authentication
dot1x timeout reauth-period 28800
!
interface Management1
  ip address 10.83.28.225/22
!
ip routing
!
ip route 0.0.0.0/0 10.83.28.1
!
ip radius source-interface Management1
!
dot1x system-auth-control
dot1x protocol lldp bypass
!
end

```

Validating the RADIUS Authentication:

Arista-1#show radius

RADIUS server : 10.83.28.244/1812/1813
Dynamic authorization UDP port: 3799

```

    Messages sent:          2
    Messages received:     2
    Requests accepted:     2
    Requests rejected:     0
    Requests timeout:      0
Requests retransmitted:   0
    Bad responses:         0
    DNS errors:            0
    CoA request received:  0
    DM request received:   0
    CoA ack sent:          0
    DM ack sent:           0
    CoA Nak sent:          0
    DM Nak sent:           0

```

RADIUS server : 10.83.28.123/1812/1813
Dynamic authorization UDP port: 3799

```

    Messages sent:          0
    Messages received:     0
    Requests accepted:     0
    Requests rejected:     0
    Requests timeout:      0
Requests retransmitted:   0
    Bad responses:         0
    DNS errors:            0
    CoA request received:  0
    DM request received:   0
    CoA ack sent:          0
    DM ack sent:           0
    CoA Nak sent:          0
    DM Nak sent:           0

```

```
RADIUS server-group: MyRadiusServers
  0: 10.83.28.244/1812
  1: 10.83.28.123/1812
```

Last time counters were cleared: never

```
Arista-1#show dot1x hosts
```

```
Interface: Ethernet54/1
Supplicant MAC      Auth Method      State      VLAN Id
-----
001c.7385.ea3d     MAC-BASED-AUTH  SUCCESS   50
```

```
Interface: Ethernet54/2
Supplicant MAC      Auth Method      State      VLAN Id
-----
001c.7386.5685     MAC-BASED-AUTH  SUCCESS   50
```

```
Arista-1#show vlan 50
```

```
VLAN Name              Status      Ports
-----
50   vlan50              active      Et54/1, Et54/2
```

Both RADIUS Servers not available:

```
Arista-1(config)#show dot1x hosts
```

```
Interface: Ethernet54/1
Supplicant MAC      Auth Method      State      VLAN Id
-----
001c.7385.ea3d     MAC-BASED-AUTH  FAILED    10
```

```
Interface: Ethernet54/2
Supplicant MAC      Auth Method      State      VLAN Id
-----
001c.7386.5685     MAC-BASED-AUTH  FAILED    10
```

```
Arista-1(config)#show vlan 10
```

```
VLAN Name              Status      Ports
-----
10   guest              active      Et54/1, Et54/2
```

Changing MAC based authentication format:

Sometimes it may be necessary to change the format of the MAC addresses send to the RADIUS server (e.g. without any colons used by default):

```
Arista-1(config)#dot1x
```

```
Arista-1(config-dot1x)#mac-based-auth radius av-pair user-name delimiter ?
```

```
colon  Set colon as delimiter in MAC address string
hyphen Set hyphen as delimiter in MAC address string
none   No delimiter in MAC address string
```

period Set period as delimiter in MAC address string
Arista-1(config-dot1x)#**mac-based-auth radius av-pair user-name delimiter none ?**
lowercase MAC address string in lowercase
uppercase MAC address string in uppercase

Arista-1(config-dot1x)#**mac-based-auth radius av-pair user-name delimiter none**
lowercase

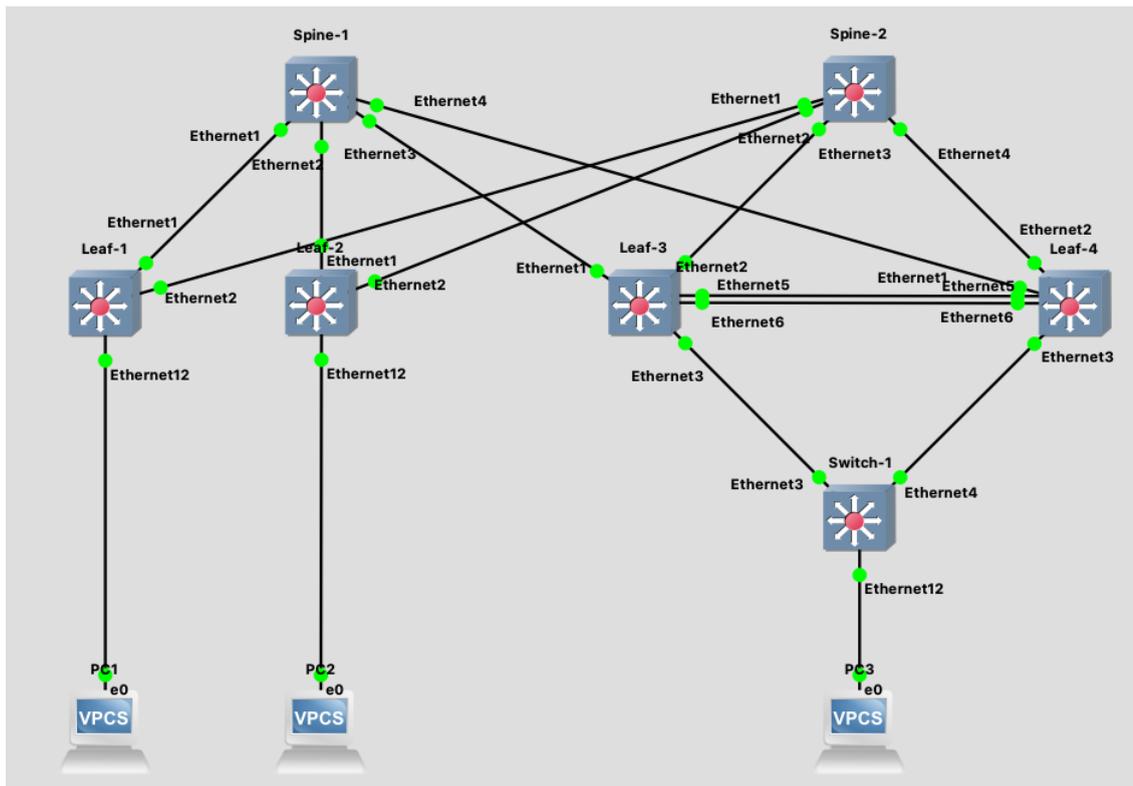
IP Fabric

With Arista IP Fabrics support is provided for a Layer 3 Clos deployment for data center sites where all links in the Clos topology are Layer 3 links.

To enable Layer 2 domain extension across the Layer 3 infrastructure, VXLAN encapsulation is leveraged. IP Fabrics provide multitenancy at Layers 2 and 3, enabling traffic isolation and segmentation across the fabric.

Additional information can be found at:

https://www.arista.com/custom_data/downloads/?f=/support/download/DesignGuides/EVPN_Deployment_Guide.pdf



Loopback Interfaces for BGP Peering and VXLAN Tunnel Sources

Node:	Loopback0/ Router-ID:	Loopback1:
Spine-1	1.1.1.201	n.a.
Spine-2	1.1.1.202	n.a.
Leaf-1	1.1.1.101	2.2.2.101
Leaf-2	1.1.1.102	2.2.2.102
Leaf-3	1.1.1.103	2.2.2.113
Leaf-4	1.1.1.104	2.2.2.113

Point-to-Point L3 IP Connections:

Node 1	Interface IP Address	Node 2	Interface IP Address
Spine-1	Eth1 10.101.201.201/24	Leaf-1	Eth1 10.101.201.101/24
Spine-1	Eth2 10.102.201.201/24	Leaf-2	Eth1 10.102.201.102/24
Spine-1	Eth3 10.103.201.201/24	Leaf-3	Eth1 10.103.201.103/24
Spine-1	Eth4 10.104.201.201/24	Leaf-4	Eth1 10.104.201.104/24
Spine-2	Eth1 10.101.202.202/24	Leaf-1	Eth2 10.101.202.101/24
Spine-2	Eth2 10.102.202.202/24	Leaf-2	Eth2 10.102.202.102/24
Spine-2	Eth3 10.103.202.202/24	Leaf-3	Eth2 10.103.202.103/24
Spine-2	Eth4 10.104.202.202/24	Leaf-4	Eth2 10.104.202.104/24

BGP AS Numbers:

Node:	AS:
Spine-1	66001
Spine-2	66001
Leaf-1	65001
Leaf-2	65002
Leaf-3	65013
Leaf-4	65013

IP Fabric BGP Underlay Spine Layer

All Spine nodes have the **same BGP AS number**.

In order for EOS to support the use of the EVPN address-family, a **multi-agent routing model** must be used for BGP (**reload required!**)

```
Spine-1(config)#service routing protocols model multi-agent
! Change will take effect only after switch reboot
Spine-1(config)#write mem
Copy completed successfully.
Spine-1(config)#reload
! Image does not support next image compatibility checks. Running only
checks from the current image.
Proceed with reload? [confirm]y
```

To prevent traffic from being dropped in situations where routes are advertised, but they aren't programmed yet in hardware please configure "**update wait-install**".

Important Note: Enabling this feature in vEOS will cause routes not to be advertised at all!

Spine-1:

```
service routing protocols model
multi-agent
!
hostname Spine-1
!
spanning-tree mode mstp
!
no aaa root
!
username admin privilege 15 role
network-admin secret arista
!
vrf instance MGMT
!
interface Ethernet1
  description Leaf-1
  no switchport
  ip address 10.101.201.201/24
!
interface Ethernet2
  description Leaf-2
  no switchport
  ip address 10.102.201.201/24
!
interface Ethernet3
  description Leaf-3
  no switchport
  ip address 10.103.201.201/24
!
interface Ethernet4
  description Leaf-4
  no switchport
  ip address 10.104.201.201/24
!
interface Management1
  vrf MGMT
  ip address 192.168.10.71/24
  no lldp receive
!
ip route vrf MGMT 0.0.0.0/0
192.168.10.1
!
ip routing
no ip routing vrf MGMT
!
```

Spine-2:

```
service routing protocols model
multi-agent
!
hostname Spine-2
!
spanning-tree mode mstp
!
no aaa root
!
username admin privilege 15 role
network-admin secret arista
!
vrf instance MGMT
!
interface Ethernet1
  description Leaf-1
  no switchport
  ip address 10.101.202.202/24
!
interface Ethernet2
  description Leaf-2
  no switchport
  ip address 10.102.202.202/24
!
interface Ethernet3
  description Leaf-3
  no switchport
  ip address 10.103.202.202/24
!
interface Ethernet4
  description Leaf-4
  no switchport
  ip address 10.104.202.202/24
!
interface Management1
  vrf MGMT
  ip address 192.168.10.72/24
  no lldp receive
!
ip route vrf MGMT 0.0.0.0/0
192.168.10.1
!
ip routing
no ip routing vrf MGMT
!
```

Spine1 :

```
router bgp 66001
  router-id 1.1.1.201
  update wait-install
  neighbor IPv4-UNDERLAY-LEAFS peer
  group
    neighbor IPv4-UNDERLAY-LEAFS
  password secretpw
  neighbor IPv4-UNDERLAY-LEAFS
  send-community
  neighbor IPv4-UNDERLAY-LEAFS
  maximum-routes 12000
  neighbor 10.101.201.101 peer group
  IPv4-UNDERLAY-LEAFS
  neighbor 10.101.201.101 remote-as
  65001
  neighbor 10.102.201.102 peer group
  IPv4-UNDERLAY-LEAFS
  neighbor 10.102.201.102 remote-as
  65002
  neighbor 10.103.201.103 peer group
  IPv4-UNDERLAY-LEAFS
  neighbor 10.103.201.103 remote-as
  65013
  neighbor 10.104.201.104 peer group
  IPv4-UNDERLAY-LEAFS
  neighbor 10.104.201.104 remote-as
  65013
  redistribute connected
!
management api http-commands
  no shutdown
!
  vrf MGMT
    no shutdown
!
end
```

Spine2 :

```
router bgp 66001
  router-id 1.1.1.202
  update wait-install
  neighbor IPv4-UNDERLAY-LEAFS peer
  group
    neighbor IPv4-UNDERLAY-LEAFS
  password secretpw
  neighbor IPv4-UNDERLAY-LEAFS
  send-community
  neighbor IPv4-UNDERLAY-LEAFS
  maximum-routes 12000
  neighbor 10.101.202.101 peer group
  IPv4-UNDERLAY-LEAFS
  neighbor 10.101.202.101 remote-as
  65001
  neighbor 10.102.202.102 peer group
  IPv4-UNDERLAY-LEAFS
  neighbor 10.102.202.102 remote-as
  65002
  neighbor 10.103.202.103 peer group
  IPv4-UNDERLAY-LEAFS
  neighbor 10.103.202.103 remote-as
  65013
  neighbor 10.104.202.104 peer group
  IPv4-UNDERLAY-LEAFS
  neighbor 10.104.202.104 remote-as
  65013
  redistribute connected
!
management api http-commands
  no shutdown
!
  vrf MGMT
    no shutdown
!
end
```

IP Fabric BGP Underlay Leaf-Range Peer-Filter

For simplification you may use **Leaf-Range Peer-Filters** at the Spine Layer to avoid configuring remote AS numbers each time when adding Leafs to the fabric.

The default peering in the IPv4 Unicast address-family where the BGP process in EOS will automatically attempt to peer with all configured neighbors is disabled in this example.

Spine1 :

```
!  
peer-filter leaf-range  
    10 match as-range 65001-65100  
result accept  
!  
router bgp 66001  
    router-id 1.1.1.201  
    update wait-install  
    no bgp default ipv4-unicast  
    bgp listen range 10.0.0.0/8  
peer-group IPv4-UNDERLAY-LEAFS  
peer-filter leaf-range  
    neighbor IPv4-UNDERLAY-LEAFS peer  
group  
    neighbor IPv4-UNDERLAY-LEAFS  
password secretpw  
    neighbor IPv4-UNDERLAY-LEAFS  
send-community  
    neighbor IPv4-UNDERLAY-LEAFS  
maximum-routes 12000  
    redistribute connected  
    !  
    address-family ipv4  
        neighbor IPv4-UNDERLAY-LEAFS  
activate  
!
```

Spine2 :

```
!  
peer-filter leaf-range  
    10 match as-range 65001-65100  
result accept  
!  
router bgp 66001  
    router-id 1.1.1.202  
    update wait-install  
    no bgp default ipv4-unicast  
    bgp listen range 10.0.0.0/8  
peer-group IPv4-UNDERLAY-LEAFS  
peer-filter leaf-range  
    neighbor IPv4-UNDERLAY-LEAFS peer  
group  
    neighbor IPv4-UNDERLAY-LEAFS  
password secretpw  
    neighbor IPv4-UNDERLAY-LEAFS  
send-community  
    neighbor IPv4-UNDERLAY-LEAFS  
maximum-routes 12000  
    redistribute connected  
    !  
    address-family ipv4  
        neighbor IPv4-UNDERLAY-LEAFS  
activate  
!
```

IP Fabric BGP Underlay Leaf Layer

In order for EOS to support the use of the EVPN address-family, a **multi-agent routing model** must be used for BGP (**reload required!**)

```
Leaf-1(config)#service routing protocols model multi-agent
! Change will take effect only after switch reboot
Leaf-1(config)#write mem
Copy completed successfully.
Leaf-1(config)#reload
! Image does not support next image compatibility checks. Running only
checks from the current image.
Proceed with reload? [confirm]y
```

By default, BGP will choose a single best path to reach a particular prefix.

Multi-pathing can be enabled for a prefix in BGP with the command "**maximum-paths**" where this number should match the **number of Spines** deployed in the topology

Leaf-1:

```
service routing protocols model
multi-agent
!
hostname Leaf-1
!
spanning-tree mode mstp
!
no aaa root
!
username admin privilege 15 role
network-admin secret arista
!
vrf instance MGMT
!
interface Ethernet1
    description Spine-1
    no switchport
    ip address 10.101.201.101/24
!
interface Ethernet2
    description Spine-2
    no switchport
    ip address 10.101.202.101/24
!
interface Management1
    vrf MGMT
    ip address 192.168.10.73/24
    no lldp receive
!
ip route vrf MGMT 0.0.0.0/0
192.168.10.1
!
ip routing
no ip routing vrf MGMT
!
router bgp 65001
    router-id 1.1.1.101
    update wait-install
    maximum-paths 2
    neighbor IPv4-UNDERLAY-SPINES peer
group
    neighbor IPv4-UNDERLAY-SPINES
remote-as 66001
    neighbor IPv4-UNDERLAY-SPINES
password secretpw
    neighbor IPv4-UNDERLAY-SPINES
send-community
    neighbor IPv4-UNDERLAY-SPINES
maximum-routes 12000
    neighbor 10.101.201.201 peer group
```

Leaf-2:

```
service routing protocols model
multi-agent
!
hostname Leaf-2
!
spanning-tree mode mstp
!
no aaa root
!
username admin privilege 15 role
network-admin secret arista
!
vrf instance MGMT
!
interface Ethernet1
    description Spine-1
    no switchport
    ip address 10.102.201.102/24
!
interface Ethernet2
    description Spine-2
    no switchport
    ip address 10.102.202.102/24
!
interface Management1
    vrf MGMT
    ip address 192.168.10.74/24
    no lldp receive
!
ip route vrf MGMT 0.0.0.0/0
192.168.10.1
!
ip routing
no ip routing vrf MGMT
!
router bgp 65002
    router-id 1.1.1.102
    update wait-install
    maximum-paths 2
    neighbor IPv4-UNDERLAY-SPINES peer
group
    neighbor IPv4-UNDERLAY-SPINES
remote-as 66001
    neighbor IPv4-UNDERLAY-SPINES
password secretpw
    neighbor IPv4-UNDERLAY-SPINES
send-community
    neighbor IPv4-UNDERLAY-SPINES
maximum-routes 12000
    neighbor 10.102.201.201 peer group
```

Leaf-1:

```
IPv4-UNDERLAY-SPINES
  neighbor 10.101.202.202 peer group
IPv4-UNDERLAY-SPINES
  redistribute connected
!
management api http-commands
  no shutdown
!
vrf MGMT
  no shutdown
!
end
```

Leaf-2:

```
IPv4-UNDERLAY-SPINES
  neighbor 10.102.202.202 peer group
IPv4-UNDERLAY-SPINES
  redistribute connected
!
management api http-commands
  no shutdown
!
vrf MGMT
  no shutdown
!
end
```

IP Fabric BGP Underlay MLAG Leaf Layer

All Leaf nodes of an MLAG Pair have the **same BGP AS number**.

All MLAG pairs will have a **dedicated VLAN interface for iBGP peering** between each other in the IPv4 Unicast address-family.

The “**next-hop-self**” policy will have to be applied which will dictate that any prefix advertised to the MLAG peer via the iBGP peering session will have the next-hop attribute modified to whatever is defined as the local “update-source” to avoid scenarios of “next-hop inaccessible” occurring for prefixes sent to the MLAG peer

Leaf-3:

```
service routing protocols model
multi-agent
!
hostname Leaf-3
!
spanning-tree mode mstp
no spanning-tree vlan 4094
!
no aaa root
!
username admin privilege 15 role
network-admin secret arista
!
vlan 10
    name client-VLAN-10
!
vlan 4093
    name MLAG_iBGP
!
vlan 4094
    name MLAG-Interlink-VLAN
    trunk group MLAG-Peer
!
vrf instance MGMT
!
interface Port-Channel1
    switchport trunk allowed vlan 10
    switchport mode trunk
    mlag 1
!
interface Port-Channel1000
    switchport mode trunk
    switchport trunk group MLAG-Peer
!
interface Ethernet1
    description Spine-1
    no switchport
    ip address 10.103.201.103/24
!
```

Leaf-4:

```
service routing protocols model
multi-agent
!
hostname Leaf-4
!
spanning-tree mode mstp
no spanning-tree vlan 4094
!
no aaa root
!
username admin privilege 15 role
network-admin secret arista
!
vlan 10
    name client-VLAN-10
!
vlan 4093
    name MLAG_iBGP
!
vlan 4094
    name MLAG-Interlink-VLAN
    trunk group MLAG-Peer
!
vrf instance MGMT
!
interface Port-Channel1
    switchport trunk allowed vlan 10
    switchport mode trunk
    mlag 1
!
interface Port-Channel1000
    switchport mode trunk
    switchport trunk group MLAG-Peer
!
interface Ethernet1
    description Spine-1
    no switchport
    ip address 10.104.201.104/24
!
```

Leaf-3:

```
interface Ethernet2
  description Spine-2
  no switchport
  ip address 10.103.202.103/24
!
interface Ethernet3
  channel-group 1 mode active
!
interface Ethernet5
  description MLAG-Interface
  channel-group 1000 mode active
!
interface Ethernet6
  description MLAG-Interface
  channel-group 1000 mode active
!
interface Management1
  vrf MGMT
  ip address 192.168.10.83/24
!
interface Vlan4093
  description MLAG iBGP Peering
  ip address 192.0.0.1/24
!
interface Vlan4094
  description MLAG-Interlink-IP
  no autostate
  ip address 10.10.10.1/24
!
ip route vrf MGMT 0.0.0.0/0
192.168.10.1
!
ip routing
no ip routing vrf MGMT
!
mlag configuration
  domain-id Arista-MLAG-1
  local-interface Vlan4094
  peer-address 10.10.10.2
  peer-link Port-Channel1000
```

Leaf-4:

```
interface Ethernet2
  description Spine-2
  no switchport
  ip address 10.104.202.104/24
!
interface Ethernet3
  channel-group 1 mode active
!
interface Ethernet5
  description MLAG-Interface
  channel-group 1000 mode active
!
interface Ethernet6
  description MLAG-Interface
  channel-group 1000 mode active
!
interface Management1
  vrf MGMT
  ip address 192.168.10.84/24
!
interface Vlan4093
  description MLAG iBGP Peering
  ip address 192.0.0.2/24
!
interface Vlan4094
  description MLAG-Interlink-IP
  no autostate
  ip address 10.10.10.2/24
!
ip route vrf MGMT 0.0.0.0/0
192.168.10.1
!
ip routing
no ip routing vrf MGMT
!
mlag configuration
  domain-id Arista-MLAG-1
  local-interface Vlan4094
  peer-address 10.10.10.1
  peer-link Port-Channel1000
```

Leaf-3:

```

router bgp 65013
  router-id 1.1.1.103
  update wait-install
  maximum-paths 2
  neighbor IPv4-UNDERLAY-SPINES peer
group
  neighbor IPv4-UNDERLAY-SPINES
remote-as 66001
  neighbor IPv4-UNDERLAY-SPINES
password secretpw
  neighbor IPv4-UNDERLAY-SPINES
send-community
  neighbor IPv4-UNDERLAY-SPINES
maximum-routes 12000
  neighbor MLAG-IPv4-UNDERLAY-PEER
peer group
  neighbor MLAG-IPv4-UNDERLAY-PEER
remote-as 65013
  neighbor MLAG-IPv4-UNDERLAY-PEER
next-hop-self
  neighbor MLAG-IPv4-UNDERLAY-PEER
password secretpw
  neighbor MLAG-IPv4-UNDERLAY-PEER
send-community
  neighbor MLAG-IPv4-UNDERLAY-PEER
maximum-routes 12000
  neighbor 10.103.201.201 peer group
IPv4-UNDERLAY-SPINES
  neighbor 10.103.202.202 peer group
IPv4-UNDERLAY-SPINES
  neighbor 192.0.0.2 peer group
MLAG-IPv4-UNDERLAY-PEER
  redistribute connected
!
management api http-commands
  no shutdown
!
  vrf MGMT
  no shutdown
!
end

```

Leaf-4:

```

router bgp 65013
  router-id 1.1.1.104
  update wait-install
  maximum-paths 2
  neighbor IPv4-UNDERLAY-SPINES peer
group
  neighbor IPv4-UNDERLAY-SPINES
remote-as 66001
  neighbor IPv4-UNDERLAY-SPINES
password secretpw
  neighbor IPv4-UNDERLAY-SPINES
send-community
  neighbor IPv4-UNDERLAY-SPINES
maximum-routes 12000
  neighbor MLAG-IPv4-UNDERLAY-PEER
peer group
  neighbor MLAG-IPv4-UNDERLAY-PEER
remote-as 65013
  neighbor MLAG-IPv4-UNDERLAY-PEER
next-hop-self
  neighbor MLAG-IPv4-UNDERLAY-PEER
password secretpw
  neighbor MLAG-IPv4-UNDERLAY-PEER
send-community
  neighbor MLAG-IPv4-UNDERLAY-PEER
maximum-routes 12000
  neighbor 10.104.201.201 peer group
IPv4-UNDERLAY-SPINES
  neighbor 10.104.202.202 peer group
IPv4-UNDERLAY-SPINES
  neighbor 192.0.0.1 peer group
MLAG-IPv4-UNDERLAY-PEER
  redistribute connected
!
management api http-commands
  no shutdown
!
  vrf MGMT
  no shutdown
!
end

```

Checking Underlay BGP Connectivity:

Spine Layer:

```
Spine-1#show ip bgp summary
```

```
BGP summary information for VRF default
```

```
Router identifier 1.1.1.201, local AS number 66001
```

```
Neighbor Status Codes: m - Under maintenance
```

Neighbor	V	AS	MsgRcvd	MsgSent	InQ	OutQ	Up/Down	State	
10.101.201.101	4	65001	86	86	0	0	01:09:19	Estab	4
10.102.201.102	4	65002	72	72	0	0	00:57:58	Estab	4
10.103.201.103	4	65013	72	74	0	0	00:43:46	Estab	5
10.104.201.104	4	65013	75	77	0	0	00:37:26	Estab	6

```
6
```

```
Spine-1#
```

```
Spine-1#show ip route bgp
```

```
VRF: default
```

```
Codes: C - connected, S - static, K - kernel,
```

```
0 - OSPF, IA - OSPF inter area, E1 - OSPF external type 1,
```

```
E2 - OSPF external type 2, N1 - OSPF NSSA external type 1,
```

```
N2 - OSPF NSSA external type2, B - BGP, B I - iBGP, B E - eBGP,
```

```
R - RIP, I L1 - IS-IS level 1, I L2 - IS-IS level 2,
```

```
O3 - OSPFv3, A B - BGP Aggregate, A O - OSPF Summary,
```

```
NG - Nexthop Group Static Route, V - VXLAN Control Service,
```

```
DH - DHCP client installed default route, M - Martian,
```

```
DP - Dynamic Policy Route, L - VRF Leaked
```

```
B E 1.1.1.101/32 [200/0] via 10.101.201.101, Ethernet1
B E 1.1.1.102/32 [200/0] via 10.102.201.102, Ethernet2
B E 1.1.1.103/32 [200/0] via 10.103.201.103, Ethernet3
B E 1.1.1.104/32 [200/0] via 10.104.201.104, Ethernet4
B E 2.2.2.101/32 [200/0] via 10.101.201.101, Ethernet1
B E 2.2.2.102/32 [200/0] via 10.102.201.102, Ethernet2
B E 2.2.2.113/32 [200/0] via 10.103.201.103, Ethernet3
B E 10.10.10.0/24 [200/0] via 10.103.201.103, Ethernet3
B E 10.101.202.0/24 [200/0] via 10.101.201.101, Ethernet1
B E 10.102.202.0/24 [200/0] via 10.102.201.102, Ethernet2
B E 10.103.202.0/24 [200/0] via 10.103.201.103, Ethernet3
B E 10.104.202.0/24 [200/0] via 10.104.201.104, Ethernet4
B E 192.0.0.0/24 [200/0] via 10.104.201.104, Ethernet4
```

```
Spine-1#
```

Leaf-Layer

Leaf-1#show ip bgp summary

BGP summary information for VRF default

Router identifier 1.1.1.101, local AS number 65001

Neighbor Status Codes: m - Under maintenance

Neighbor	V	AS	MsgRcvd	MsgSent	InQ	OutQ	Up/Down	State	
PfxRcd PfxAcc									
10.101.201.201	4	66001	89	88	0	0	01:11:18	Estab	5
5									
10.101.202.202	4	66001	89	89	0	0	01:11:18	Estab	5
5									

Leaf-1#

Leaf-1#show ip route bgp

VRF: default

Codes: C - connected, S - static, K - kernel,

O - OSPF, IA - OSPF inter area, E1 - OSPF external type 1,

E2 - OSPF external type 2, N1 - OSPF NSSA external type 1,

N2 - OSPF NSSA external type2, B - BGP, B I - iBGP, B E - eBGP,

R - RIP, I L1 - IS-IS level 1, I L2 - IS-IS level 2,

O3 - OSPFv3, A B - BGP Aggregate, A O - OSPF Summary,

NG - Nexthop Group Static Route, V - VXLAN Control Service,

DH - DHCP client installed default route, M - Martian,

DP - Dynamic Policy Route, L - VRF Leaked

B E 1.1.1.201/32 [200/0] via 10.101.201.201, Ethernet1
B E 1.1.1.202/32 [200/0] via 10.101.202.202, Ethernet2
B E 10.102.201.0/24 [200/0] via 10.101.201.201, Ethernet1
B E 10.102.202.0/24 [200/0] via 10.101.202.202, Ethernet2
B E 10.103.201.0/24 [200/0] via 10.101.201.201, Ethernet1
B E 10.103.202.0/24 [200/0] via 10.101.202.202, Ethernet2
B E 10.104.201.0/24 [200/0] via 10.101.201.201, Ethernet1
B E 10.104.202.0/24 [200/0] via 10.101.202.202, Ethernet2

Leaf-1#

MLAG Leaf Layer:

Leaf-3#show mlag

MLAG Configuration:

```
domain-id      : Arista-MLAG-1
local-interface : Vlan4094
peer-address   : 10.10.10.2
peer-link      : Port-Channel1000
peer-config    : consistent
```

MLAG Status:

```
state          : Active
negotiation status : Connected
peer-link status : Up
local-int status : Up
system-id      : 0e:7e:d1:2c:c9:a1
dual-primary detection : Disabled
```

MLAG Ports:

```
Disabled      : 0
Configured    : 0
Inactive      : 0
Active-partial : 0
Active-full   : 1
```

Leaf-3#show ip bgp summary

BGP summary information for VRF default

Router identifier 1.1.1.103, local AS number 65013

Neighbor Status Codes: m - Under maintenance

Neighbor	V	AS	MsgRcvd	MsgSent	InQ	OutQ	Up/Down
10.103.201.201	4	66001	191	190	0	0	02:39:42
Estab 3 3							
10.103.202.202	4	66001	195	191	0	0	02:39:43
Estab 3 3							
192.0.0.2		4 65013		184	188	0	0
02:34:14 Estab	5	5					

Leaf-3#

Leaf-3#show ip route bgp

VRF: default

Codes: C - connected, S - static, K - kernel,
O - OSPF, IA - OSPF inter area, E1 - OSPF external type 1,
E2 - OSPF external type 2, N1 - OSPF NSSA external type 1,
N2 - OSPF NSSA external type2, B - BGP, B I - iBGP, B E - eBGP,
R - RIP, I L1 - IS-IS level 1, I L2 - IS-IS level 2,
O3 - OSPFv3, A B - BGP Aggregate, A O - OSPF Summary,
NG - Nexthop Group Static Route, V - VXLAN Control Service,
DH - DHCP client installed default route, M - Martian,
DP - Dynamic Policy Route, L - VRF Leaked

```
B I    1.1.1.104/32 [200/0] via 192.0.0.2, Vlan4093  
B E    1.1.1.201/32 [200/0] via 10.103.201.201, Ethernet1  
B E    1.1.1.202/32 [200/0] via 10.103.202.202, Ethernet2  
B E    10.101.201.0/24 [200/0] via 10.103.201.201, Ethernet1  
B E    10.101.202.0/24 [200/0] via 10.103.202.202, Ethernet2  
B E    10.102.201.0/24 [200/0] via 10.103.201.201, Ethernet1  
B E    10.102.202.0/24 [200/0] via 10.103.202.202, Ethernet2  
B I    10.104.201.0/24 [200/0] via 192.0.0.2, Vlan4093  
B I    10.104.202.0/24 [200/0] via 192.0.0.2, Vlan4093
```

Leaf-3#

IP Fabric EVPN Overlay Spine Layer:

BGP next-hop-unchanged has to be configured on the **spine switches** for each EVPN neighbor if the leaf switch is an eBGP peer. This means that eBGP BGP Spine speakers will not change the next hop while sending updates to eBGP neighbors in an IP Fabric (the next hop is the VTEP ID which remains the same in the IP Fabric).

Otherwise the VXLAN tunnel will not be created!

By default the TTL value is set to a value of 1 for eBGP sessions.

If left at the default value, the TTL would expire before reaching the Loopback address of the remote peer within an IP Fabric. **This value should consistently set to 3 or higher on all Leaf and Spine switches** with the command “**ebgp-multihop 3**”.

The reason for this is that Spine and Leaf switches would only require a TTL of 2 but if an MLAG peer lost it's connectivity to the Spines, it would need to maintain it's BGP EVPN adjacencies with the Spines by sending it's BGP Keepalives through it's MLAG peer (due to the iBGP IPv4 peering between the MLAG peers).

This would result in an additional decrement of the TTL, hence the need for 3 to cover this failure scenario.

When configuring L2 or L3 EVPN services you have to configure the Spines in a way that within the EVPN address-family, adjacencies will be established with BGP speakers sourcing their BGP session from within the 1.0.0.0/24 address range.

This maps to the **Loopback0** IP addresses configured on the systems.

Spine-1:

```
interface Loopback0
  description EVPN Peering Source
  ip address 1.1.1.201/32
!
router bgp 66001
  router-id 1.1.1.201
  update wait-install
  neighbor EVPN-OVERLAY-PEERS peer
  group
  neighbor EVPN-OVERLAY-PEERS
  next-hop-unchanged
  neighbor EVPN-OVERLAY-PEERS
  update-source Loopback0
  neighbor EVPN-OVERLAY-PEERS bfd
  neighbor EVPN-OVERLAY-PEERS
  ebgp-multihop 3
  neighbor EVPN-OVERLAY-PEERS
  password secretpw
  neighbor EVPN-OVERLAY-PEERS
  send-community
  neighbor EVPN-OVERLAY-PEERS
  maximum-routes 0
```

Spine-2:

```
interface Loopback0
  description EVPN Peering Source
  ip address 1.1.1.202/32
!
router bgp 66001
  router-id 1.1.1.202
  update wait-install
  neighbor EVPN-OVERLAY-PEERS peer
  group
  neighbor EVPN-OVERLAY-PEERS
  next-hop-unchanged
  neighbor EVPN-OVERLAY-PEERS
  update-source Loopback0
  neighbor EVPN-OVERLAY-PEERS bfd
  neighbor EVPN-OVERLAY-PEERS
  ebgp-multihop 3
  neighbor EVPN-OVERLAY-PEERS
  password secretpw
  neighbor EVPN-OVERLAY-PEERS
  send-community
  neighbor EVPN-OVERLAY-PEERS
  maximum-routes 0
```

Spine-1:

```
neighbor IPv4-UNDERLAY-LEAFS peer
group
  neighbor IPv4-UNDERLAY-LEAFS
password secretpw
  neighbor IPv4-UNDERLAY-LEAFS
send-community
  neighbor IPv4-UNDERLAY-LEAFS
maximum-routes 12000
  neighbor 1.1.1.101 peer group
EVPN-OVERLAY-PEERS
  neighbor 1.1.1.101 remote-as 65001
  neighbor 1.1.1.102 peer group
EVPN-OVERLAY-PEERS
  neighbor 1.1.1.102 remote-as 65002
  neighbor 1.1.1.103 peer group
EVPN-OVERLAY-PEERS
  neighbor 1.1.1.103 remote-as 65013
  neighbor 1.1.1.104 peer group
EVPN-OVERLAY-PEERS
  neighbor 1.1.1.104 remote-as 65013
  neighbor 10.101.201.101 peer group
IPv4-UNDERLAY-LEAFS
  neighbor 10.101.201.101 remote-as
65001
  neighbor 10.102.201.102 peer group
IPv4-UNDERLAY-LEAFS
  neighbor 10.102.201.102 remote-as
65002
  neighbor 10.103.201.103 peer group
IPv4-UNDERLAY-LEAFS
  neighbor 10.103.201.103 remote-as
65013
  neighbor 10.104.201.104 peer group
IPv4-UNDERLAY-LEAFS
  neighbor 10.104.201.104 remote-as
65013
  redistribute connected
  !
  address-family evpn
    neighbor EVPN-OVERLAY-PEERS
activate
  !
  address-family ipv4
    no neighbor EVPN-OVERLAY-PEERS
activate
  !
```

Spine-2:

```
neighbor IPv4-UNDERLAY-LEAFS peer
group
  neighbor IPv4-UNDERLAY-LEAFS
password secretpw
  neighbor IPv4-UNDERLAY-LEAFS
send-community
  neighbor IPv4-UNDERLAY-LEAFS
maximum-routes 12000
  neighbor 1.1.1.101 peer group
EVPN-OVERLAY-PEERS
  neighbor 1.1.1.101 remote-as 65001
  neighbor 1.1.1.102 peer group
EVPN-OVERLAY-PEERS
  neighbor 1.1.1.102 remote-as 65002
  neighbor 1.1.1.103 peer group
EVPN-OVERLAY-PEERS
  neighbor 1.1.1.103 remote-as 65013
  neighbor 1.1.1.104 peer group
EVPN-OVERLAY-PEERS
  neighbor 1.1.1.104 remote-as 65013
  neighbor 10.101.202.101 peer group
IPv4-UNDERLAY-LEAFS
  neighbor 10.101.202.101 remote-as
65001
  neighbor 10.102.202.102 peer group
IPv4-UNDERLAY-LEAFS
  neighbor 10.102.202.102 remote-as
65002
  neighbor 10.103.202.103 peer group
IPv4-UNDERLAY-LEAFS
  neighbor 10.103.202.103 remote-as
65013
  neighbor 10.104.202.104 peer group
IPv4-UNDERLAY-LEAFS
  neighbor 10.104.202.104 remote-as
65013
  redistribute connected
  !
  address-family evpn
    neighbor EVPN-OVERLAY-PEERS
activate
  !
  address-family ipv4
    no neighbor EVPN-OVERLAY-PEERS
activate
  !
```

IP Fabric BGP Overlay Leaf-Range Peer-Filter

For simplification you may use **Leaf-Range Peer-Filters** at the Spine Layer to avoid configuring remote AS numbers each time when adding Leafs to the fabric.

The default peering in the IPv4 Unicast address-family where the BGP process in EOS will automatically attempt to peer with all configured neighbors is disabled in this example.

Spine1 :

```
interface Loopback0
  description EVPN Peering Source
  ip address 1.1.1.201/32
!
peer-filter leaf-range
  10 match as-range 65001-65100
result accept
!
router bgp 66001
  router-id 1.1.1.201
  update wait-install
  no bgp default ipv4-unicast
  bgp listen range 1.1.1.0/24
peer-group EVPN-OVERLAY-PEERS
peer-filter leaf-range
  bgp listen range 10.0.0.0/8
peer-group IPv4-UNDERLAY-LEAFS
peer-filter leaf-range
  neighbor EVPN-OVERLAY-PEERS peer
group
  neighbor EVPN-OVERLAY-PEERS
next-hop-unchanged
  neighbor EVPN-OVERLAY-PEERS
update-source Loopback0
  neighbor EVPN-OVERLAY-PEERS bfd
  neighbor EVPN-OVERLAY-PEERS
ebgp-multihop 3
  neighbor EVPN-OVERLAY-PEERS
password secretpw
  neighbor EVPN-OVERLAY-PEERS
send-community
  neighbor EVPN-OVERLAY-PEERS
maximum-routes 0
  neighbor IPv4-UNDERLAY-LEAFS peer
group
  neighbor IPv4-UNDERLAY-LEAFS
password secretpw
  neighbor IPv4-UNDERLAY-LEAFS
send-community
  neighbor IPv4-UNDERLAY-LEAFS
maximum-routes 12000
  redistribute connected
!
```

Spine2 :

```
interface Loopback0
  description EVPN Peering Source
  ip address 1.1.1.202/32
!
peer-filter leaf-range
  10 match as-range 65001-65100
result accept
!
router bgp 66001
  router-id 1.1.1.202
  update wait-install
  no bgp default ipv4-unicast
  bgp listen range 1.1.1.0/24
peer-group EVPN-OVERLAY-PEERS
peer-filter leaf-range
  bgp listen range 10.0.0.0/8
peer-group IPv4-UNDERLAY-LEAFS
peer-filter leaf-range
  neighbor EVPN-OVERLAY-PEERS peer
group
  neighbor EVPN-OVERLAY-PEERS
next-hop-unchanged
  neighbor EVPN-OVERLAY-PEERS
update-source Loopback0
  neighbor EVPN-OVERLAY-PEERS bfd
  neighbor EVPN-OVERLAY-PEERS
ebgp-multihop 3
  neighbor EVPN-OVERLAY-PEERS
password secretpw
  neighbor EVPN-OVERLAY-PEERS
send-community
  neighbor EVPN-OVERLAY-PEERS
maximum-routes 0
  neighbor IPv4-UNDERLAY-LEAFS peer
group
  neighbor IPv4-UNDERLAY-LEAFS
password secretpw
  neighbor IPv4-UNDERLAY-LEAFS
send-community
  neighbor IPv4-UNDERLAY-LEAFS
maximum-routes 12000
  redistribute connected
!
```

Spine1 :

```
    address-family evpn
      neighbor EVPN-OVERLAY-PEERS
activate
!
    address-family ipv4
      neighbor IPv4-UNDERLAY-LEAFS
activate
!
```

Spine2 :

```
    address-family evpn
      neighbor EVPN-OVERLAY-PEERS
activate
!
    address-family ipv4
      neighbor IPv4-UNDERLAY-LEAFS
activate
!
```

IP Fabric EVPN Overlay Leaf Layer

“**hardware tcam profile vxlan-routing**” (R-Series Platforms only!) should be configured for optimized VXLAN processing.

By default the TTL value is set to a value of 1 for eBGP sessions.

If left at the default value, the TTL would expire before reaching the Loopback address of the remote peer within an IP Fabric. **This value should consistently set to 3 or higher on all Leaf and Spine switches** with the command “**ebgp-multihop 3**”.

The reason for this is that Spine and Leaf switches would only require a TTL of 2 but if an MLAG peer lost it’s connectivity to the Spines, it would need to maintain it’s BGP EVPN adjacencies with the Spines by sending it’s BGP Keepalives through it’s MLAG peer (due to the iBGP IPv4 peering between the MLAG peers).

This would result in an additional decrement of the TTL, hence the need for 3 to cover this failure scenario.

When configuring a L2 or L3 EVPN services you have to configure the Leafs in a way that within the EVPN address-family, adjacencies will be established with BGP speakers sourcing their BGP session from within the 1.0.0.0/24 address range.

This maps to the **Loopback0** IP addresses configured on the systems.

When configuring **MLAG Pairs** both **Loopback 1 addresses are identical** on the MLAG leaves.

Leaf-1:

```
service routing protocols model
multi-agent
!
hardware tcam profile vxlan-routing
!
interface Loopback0
  description EVPN Peering Source
  ip address 1.1.1.101/32
!
interface Loopback1
  description VXLAN Tunnel Source
  ip address 2.2.2.101/32
!
router bgp 65001
  router-id 1.1.1.101
  update wait-install
  maximum-paths 2
  neighbor EVPN-OVERLAY-PEERS peer
group
  neighbor EVPN-OVERLAY-PEERS
remote-as 66001
  neighbor EVPN-OVERLAY-PEERS
update-source Loopback0
  neighbor EVPN-OVERLAY-PEERS bfd
  neighbor EVPN-OVERLAY-PEERS
```

Leaf-2:

```
service routing protocols model
multi-agent
!
hardware tcam profile vxlan-routing
!
interface Loopback0
  description EVPN Peering Source
  ip address 1.1.1.102/32
!
interface Loopback1
  description VXLAN Tunnel Source
  ip address 2.2.2.102/32
!
router bgp 65002
  router-id 1.1.1.102
  update wait-install
  maximum-paths 2
  neighbor EVPN-OVERLAY-PEERS peer
group
  neighbor EVPN-OVERLAY-PEERS
remote-as 66001
  neighbor EVPN-OVERLAY-PEERS
update-source Loopback0
  neighbor EVPN-OVERLAY-PEERS bfd
  neighbor EVPN-OVERLAY-PEERS
```

Leaf-1:

```

ebgp-multihop 3
  neighbor EVPN-OVERLAY-PEERS
password secretpw
  neighbor EVPN-OVERLAY-PEERS
send-community
neighbor EVPN-OVERLAY-PEERS
maximum-routes 12000
  neighbor IPv4-UNDERLAY-SPINES peer
group
  neighbor IPv4-UNDERLAY-SPINES
remote-as 66001
  neighbor IPv4-UNDERLAY-SPINES
password secretpw
  neighbor IPv4-UNDERLAY-SPINES
send-community
  neighbor IPv4-UNDERLAY-SPINES
maximum-routes 12000
  neighbor 1.1.1.201 peer group
EVPN-OVERLAY-PEERS
  neighbor 1.1.1.202 peer group
EVPN-OVERLAY-PEERS
  neighbor 10.101.201.201 peer group
IPv4-UNDERLAY-SPINES
  neighbor 10.101.202.202 peer group
IPv4-UNDERLAY-SPINES
  redistribute connected
  !
  address-family evpn
    neighbor EVPN-OVERLAY-PEERS
activate
  !
  address-family ipv4
    no neighbor EVPN-OVERLAY-PEERS
activate
  !

```

Leaf-2:

```

ebgp-multihop 3
  neighbor EVPN-OVERLAY-PEERS
password secretpw
  neighbor EVPN-OVERLAY-PEERS
send-community
neighbor EVPN-OVERLAY-PEERS
maximum-routes 12000
  neighbor IPv4-UNDERLAY-SPINES peer
group
  neighbor IPv4-UNDERLAY-SPINES
remote-as 66001
  neighbor IPv4-UNDERLAY-SPINES
password secretpw
  neighbor IPv4-UNDERLAY-SPINES
send-community
  neighbor IPv4-UNDERLAY-SPINES
maximum-routes 12000
  neighbor 1.1.1.201 peer group
EVPN-OVERLAY-PEERS
  neighbor 1.1.1.202 peer group
EVPN-OVERLAY-PEERS
  neighbor 10.102.201.201 peer group
IPv4-UNDERLAY-SPINES
  neighbor 10.102.202.202 peer group
IPv4-UNDERLAY-SPINES
  redistribute connected
  !
  address-family evpn
    neighbor EVPN-OVERLAY-PEERS
activate
  !
  address-family ipv4
    no neighbor EVPN-OVERLAY-PEERS
activate
  !

```

Leaf-3:

```
service routing protocols model
multi-agent
!
hardware tcam profile vxlan-routing
!
interface Loopback0
  description EVPN Peering Source
  ip address 1.1.1.103/32
!
interface Loopback1
  description VXLAN Tunnel Source
  ip address 2.2.2.113/32
!
router bgp 65013
  router-id 1.1.1.103
  update wait-install
  maximum-paths 2
  neighbor EVPN-OVERLAY-PEERS peer
  group
    neighbor EVPN-OVERLAY-PEERS
  remote-as 66001
  neighbor EVPN-OVERLAY-PEERS
  update-source Loopback0
  neighbor EVPN-OVERLAY-PEERS bfd
  neighbor EVPN-OVERLAY-PEERS
  ebgp-multihop 3
  neighbor EVPN-OVERLAY-PEERS
  password secretpw
  neighbor EVPN-OVERLAY-PEERS
  send-community
  neighbor EVPN-OVERLAY-PEERS
  maximum-routes 12000
  neighbor IPv4-UNDERLAY-SPINES peer
  group
    neighbor IPv4-UNDERLAY-SPINES
  remote-as 66001
  neighbor IPv4-UNDERLAY-SPINES
  password secretpw
  neighbor IPv4-UNDERLAY-SPINES
  send-community
  neighbor IPv4-UNDERLAY-SPINES
  maximum-routes 12000
  neighbor MLAG-IPv4-UNDERLAY-PEER
  peer group
  neighbor MLAG-IPv4-UNDERLAY-PEER
  remote-as 65013
  neighbor MLAG-IPv4-UNDERLAY-PEER
  next-hop-self
  neighbor MLAG-IPv4-UNDERLAY-PEER
  password secretpw
  neighbor MLAG-IPv4-UNDERLAY-PEER
  send-community
```

Leaf-4:

```
service routing protocols model
multi-agent
!
hardware tcam profile vxlan-routing
!
interface Loopback0
  description EVPN Peering Source
  ip address 1.1.1.104/32
!
interface Loopback1
  description VXLAN Tunnel Source
  ip address 2.2.2.113/32
!
router bgp 65013
  router-id 1.1.1.104
  update wait-install
  maximum-paths 2
  neighbor EVPN-OVERLAY-PEERS peer
  group
    neighbor EVPN-OVERLAY-PEERS
  remote-as 66001
  neighbor EVPN-OVERLAY-PEERS
  update-source Loopback0
  neighbor EVPN-OVERLAY-PEERS bfd
  neighbor EVPN-OVERLAY-PEERS
  ebgp-multihop 3
  neighbor EVPN-OVERLAY-PEERS
  password secretpw
  neighbor EVPN-OVERLAY-PEERS
  send-community
  neighbor EVPN-OVERLAY-PEERS
  maximum-routes 12000
  neighbor IPv4-UNDERLAY-SPINES peer
  group
    neighbor IPv4-UNDERLAY-SPINES
  remote-as 66001
  neighbor IPv4-UNDERLAY-SPINES
  password secretpw
  neighbor IPv4-UNDERLAY-SPINES
  send-community
  neighbor IPv4-UNDERLAY-SPINES
  maximum-routes 12000
  neighbor MLAG-IPv4-UNDERLAY-PEER
  peer group
  neighbor MLAG-IPv4-UNDERLAY-PEER
  remote-as 65013
  neighbor MLAG-IPv4-UNDERLAY-PEER
  next-hop-self
  neighbor MLAG-IPv4-UNDERLAY-PEER
  password secretpw
  neighbor MLAG-IPv4-UNDERLAY-PEER
  send-community
```

Leaf-3:

```
neighbor MLAG-IPv4-UNDERLAY-PEER
maximum-routes 12000
neighbor 1.1.1.201 peer group
EVPN-OVERLAY-PEERS
neighbor 1.1.1.202 peer group
EVPN-OVERLAY-PEERS
neighbor 10.103.201.201 peer group
IPv4-UNDERLAY-SPINES
neighbor 10.103.202.202 peer group
IPv4-UNDERLAY-SPINES
neighbor 192.0.0.2 peer group
MLAG-IPv4-UNDERLAY-PEER
redistribute connected
!
address-family evpn
neighbor EVPN-OVERLAY-PEERS
activate
!
address-family ipv4
no neighbor EVPN-OVERLAY-PEERS
activate
!
```

Leaf-4:

```
neighbor MLAG-IPv4-UNDERLAY-PEER
maximum-routes 12000
neighbor 1.1.1.201 peer group
EVPN-OVERLAY-PEERS
neighbor 1.1.1.202 peer group
EVPN-OVERLAY-PEERS
neighbor 10.104.201.201 peer group
IPv4-UNDERLAY-SPINES
neighbor 10.104.202.202 peer group
IPv4-UNDERLAY-SPINES
neighbor 192.0.0.1 peer group
MLAG-IPv4-UNDERLAY-PEER
redistribute connected
!
address-family evpn
neighbor EVPN-OVERLAY-PEERS
activate
!
address-family ipv4
no neighbor EVPN-OVERLAY-PEERS
activate
!
```

Checking Overlay BGP Connectivity

Spine Layer:

```
Spine-1#show bgp evpn summary
```

```
BGP summary information for VRF default
```

```
Router identifier 1.1.1.201, local AS number 66001
```

```
Neighbor Status Codes: m - Under maintenance
```

Neighbor	V	AS	MsgRcvd	MsgSent	InQ	OutQ	Up/Down	State	
PfxRcd PfxAcc									
1.1.1.101	4	65001	57	59	0	0	00:46:16	Estab	0
0									
1.1.1.102	4	65002	56	55	0	0	00:39:49	Estab	0
0									
1.1.1.103	4	65013	23	23	0	0	00:17:23	Estab	0
0									
1.1.1.104	4	65013	22	23	0	0	00:16:26	Estab	0
0									

```
Spine-1#
```

IP Fabric L2 EVPN Overlay Leaf Layer

When configuring a L2 EVPN you have to configure the L2 VLAN, Port assignments and add the VLAN to an EVPN service.

Please be aware that the source of a VXLAN tunnel must be a Loopback address!

Leaf-1:

```
vlan 10
  name VLAN-10
!
vlan 20
  name VLAN-20
!
interface Ethernet12
  description TENANT-A-VLAN-10
  switchport access vlan 10
!
interface Loopback1
  description VXLAN Tunnel Source
  ip address 2.2.2.101/32
!
interface Vxlan1
  vxlan source-interface Loopback1
  vxlan udp-port 4789
  vxlan vlan 10-50 vni 10010-10050
!
!
router bgp 65001
  !
  vlan-aware-bundle TENANT-A
    rd 1.1.1.101:1
    route-target both 1:1
    redistribute learned
    vlan 10-50
  !
```

Leaf-2:

```
vlan 10
  name VLAN-10
!
vlan 20
  name VLAN-20
!
interface Ethernet12
  description TENANT-A-VLAN-10
  switchport access vlan 10
!
interface Loopback1
  description VXLAN Tunnel Source
  ip address 2.2.2.102/32
!
interface Vxlan1
  vxlan source-interface Loopback1
  vxlan udp-port 4789
  vxlan vlan 10-50 vni 10010-10050
!
!
router bgp 65002
  !
  vlan-aware-bundle TENANT-A
    rd 1.1.1.102:1
    route-target both 1:1
    redistribute learned
    vlan 10-50
  !
```

Leaf-3:

```

vlan 10
  name VLAN-10
!
vlan 20
  name VLAN-20
!
interface Loopback1
  description VXLAN Tunnel Source
  ip address 2.2.2.113/32
!
interface Vlan4093
  description MLSG iBGP Peering
  ip address 192.0.0.1/24
!
interface Vlan4094
  description MLAG-Interlink-IP
  no autostate
  ip address 10.10.10.1/24
!
interface Port-Channel1
  switchport trunk allowed vlan 10
  switchport mode trunk
  mlag 1
!
interface Ethernet3
  channel-group 1 mode active
!
interface Vxlan1
  vxlan source-interface Loopback1
  vxlan udp-port 4789
  vxlan vlan 10-50 vni 10010-10050
!
router bgp 65013
!
  vlan-aware-bundle TENANT-A
    rd 1.1.1.103:1
    route-target both 1:1
    redistribute learned
    vlan 10-50
!

```

Leaf-4:

```

vlan 10
  name VLAN-10
!
vlan 20
  name VLAN-20
!
interface Loopback1
  description VXLAN Tunnel Source
  ip address 2.2.2.113/32
!
interface Vlan4093
  description MLAG iBGP Peering
  ip address 192.0.0.2/24
!
interface Vlan4094
  description MLAG-Interlink-IP
  no autostate
  ip address 10.10.10.2/24
!
interface Port-Channel1
  switchport trunk allowed vlan 10
  switchport mode trunk
  mlag 1
!
interface Ethernet3
  channel-group 1 mode active
!
interface Vxlan1
  vxlan source-interface Loopback1
  vxlan udp-port 4789
  vxlan vlan 10-50 vni 10010-10050
!
router bgp 65013
!
  vlan-aware-bundle TENANT-A
    rd 1.1.1.104:1
    route-target both 1:1
    redistribute learned
    vlan 10-50
!

```

Switch-1:

```
hostname Switch-1
!
spanning-tree mode mstp
!
vlan 10
  name VLAN-10
!
interface Port-Channel10
  switchport trunk allowed vlan 10
  switchport mode trunk
!
interface Ethernet3
  channel-group 10 mode active
!
interface Ethernet4
  channel-group 10 mode active
!
interface Ethernet12
  switchport access vlan 10
!
```

IP Fabric L2 EVPN Overlay Leaf Layer verification

Leaf-1#show interfaces vxlan 1

Vxlan1 is up, line protocol is up (connected)

Hardware is Vxlan

Source interface is Loopback1 and is active with 2.2.2.101

Replication/Flood Mode is headend with Flood List Source: EVPN

Remote MAC learning via EVPN

VNI mapping to VLANs

Static VLAN to VNI mapping is

[10, 10010]	[11, 10011]	[12, 10012]	[13, 10013]
[14, 10014]	[15, 10015]	[16, 10016]	[17, 10017]
[18, 10018]	[19, 10019]	[20, 10020]	[21, 10021]
[22, 10022]	[23, 10023]	[24, 10024]	[25, 10025]
[26, 10026]	[27, 10027]	[28, 10028]	[29, 10029]
[30, 10030]	[31, 10031]	[32, 10032]	[33, 10033]
[34, 10034]	[35, 10035]	[36, 10036]	[37, 10037]
[38, 10038]	[39, 10039]	[40, 10040]	[41, 10041]
[42, 10042]	[43, 10043]	[44, 10044]	[45, 10045]
[46, 10046]	[47, 10047]	[48, 10048]	[49, 10049]
[50, 10050]			

Note: All Dynamic VLANs used by VCS are internal VLANs.

Use 'show vxlan vni' for details.

Static VRF to VNI mapping is not configured

Headend replication flood vtep list is:

10 2.2.2.113 2.2.2.102

20 2.2.2.113 2.2.2.102

MLAG Shared Router MAC is 0000.0000.0000

VTEP address mask is None

Leaf-1#show mac address-table vlan 10

Mac Address Table

```
-----
```

Vlan	Mac Address	Type	Ports	Moves	Last Move
10	0050.7966.6800	DYNAMIC	Et12	1	0:04:56 ago
10	0050.7966.6801	DYNAMIC	Vx1	1	0:04:56 ago
10	0050.7966.6802	DYNAMIC	Vx1	1	0:02:52 ago

Total Mac Addresses for this criterion: 3

Multicast Mac Address Table

```
-----
```

Vlan	Mac Address	Type	Ports
------	-------------	------	-------

Total Mac Addresses for this criterion: 0

Leaf-1#show vxlan flood vtep vlan 10

VXLAN Flood VTEP Table

```
-----
```

VLANS	Ip Address
10	2.2.2.102 2.2.2.113

Leaf-1#show bgp evpn

BGP routing table information for VRF default

Router identifier 1.1.1.101, local AS number 65001

Route status codes: s - suppressed, * - valid, > - active, # - not installed, E - ECMP head, e - ECMP

S - Stale, c - Contributing to ECMP, b - backup

% - Pending BGP convergence

Origin codes: i - IGP, e - EGP, ? - incomplete

AS Path Attributes: Or-ID - Originator ID, C-LST - Cluster List, LL Nexthop - Link Local Nexthop

	Network	Next Hop	Metric	LocPref	Weight	Path
* >	RD: 1.1.1.101:1 mac-ip	10010 0050.7966.6800	-	-	0	i
* >Ec	RD: 1.1.1.102:1 mac-ip	10010 0050.7966.6801 2.2.2.102	-	100	0	
66001 65002 i						
* ec	RD: 1.1.1.102:1 mac-ip	10010 0050.7966.6801 2.2.2.102	-	100	0	
66001 65002 i						
* >Ec	RD: 1.1.1.103:1 mac-ip	10010 0050.7966.6802 2.2.2.113	-	100	0	
66001 65013 i						
* ec	RD: 1.1.1.103:1 mac-ip	10010 0050.7966.6802 2.2.2.113	-	100	0	
66001 65013 i						
* >Ec	RD: 1.1.1.104:1 mac-ip	10010 0050.7966.6802 2.2.2.113	-	100	0	
66001 65013 i						
* ec	RD: 1.1.1.104:1 mac-ip	10010 0050.7966.6802 2.2.2.113	-	100	0	
66001 65013 i						
* >	RD: 1.1.1.101:1 imet	10010 2.2.2.101	-	-	0	i
* >Ec	RD: 1.1.1.102:1 imet	10010 2.2.2.102 2.2.2.102	-	100	0	
66001 65002 i						
* ec	RD: 1.1.1.102:1 imet	10010 2.2.2.102 2.2.2.102	-	100	0	
66001 65002 i						
* >Ec	RD: 1.1.1.103:1 imet	10010 2.2.2.113 2.2.2.113	-	100	0	
66001 65013 i						
* ec	RD: 1.1.1.103:1 imet	10010 2.2.2.113 2.2.2.113	-	100	0	
66001 65013 i						
* >Ec	RD: 1.1.1.104:1 imet	10010 2.2.2.113 2.2.2.113	-	100	0	
66001 65013 i						
* ec	RD: 1.1.1.104:1 imet	10010 2.2.2.113 2.2.2.113	-	100	0	
66001 65013 i						
* >	RD: 1.1.1.101:1 imet	10020 2.2.2.101	-	-	0	i
* >Ec	RD: 1.1.1.102:1 imet	10020 2.2.2.102 2.2.2.102	-	100	0	
66001 65002 i						
* ec	RD: 1.1.1.102:1 imet	10020 2.2.2.102				

		2.2.2.102	-	100	0
66001	65002	i			
	* >Ec	RD: 1.1.1.103:1 imet 10020			
		2.2.2.113	-	100	0
66001	65013	i			
	* ec	RD: 1.1.1.103:1 imet 10020			
		2.2.2.113	-	100	0
66001	65013	i			
	* >Ec	RD: 1.1.1.104:1 imet 10020			
		2.2.2.113	-	100	0
66001	65013	i			
	* ec	RD: 1.1.1.104:1 imet 10020			
		2.2.2.113	-	100	0
66001	65013	i			
	Leaf-1#				

IP Fabric L3 EVPN Overlay Leaf Layer for symmetric Routing

Symmetric IRB is recommended, as it does not require that all VLANs and VRFs exist everywhere to guarantee host reachability.

A **common VNI** has to be configured for each VRF instance on the participating Leafs.

It is important to configure **unique Route Distinguishers (RD)** and suitable Export/ Import Route Targets (**same RTs for all participating leaf nodes**).

A dynamic VLAN will be created automatically upon the association of a VRF to a L3VNI (here VLAN 4092 to L3VNI 50001)

When configuring IP Anycast addresses the **IP addresses** and **MAC addresses** have to be **identical** for all **identical subnets** on the participating leaf nodes.

As VTEPs share the same IP address and MAC address for each respective Default Gateways, pings destined to hosts behind remote VTEPs, or even local hosts in the case of MLAG VTEPs, may not work! In order to ensure that VTEPs are able to ping remote hosts you may **optionally** configure a Loopback address that can be automatically used as the source address for pings to hosts within a respective VRF.

Leaf-1:

```
vlan 30
  name VLAN-30
!
vrf instance VRF-TENANT-A
!
interface Ethernet12
  description TENANT-A-VLAN-30
  switchport access vlan 30
!
interface Loopback0
  description EVPN Peering Source
  ip address 1.1.1.101/32
!
interface Loopback1
  description VXLAN Tunnel Source
  ip address 2.2.2.101/32
!
interface Loopback10
  vrf VRF-TENANT-A
  ip address 201.0.0.101/32
!
interface Vlan30
  vrf VRF-TENANT-A
  ip address virtual
192.168.30.250/24
!
interface Vxlan1
  vxlan source-interface Loopback1
  vxlan udp-port 4789
  vxlan vlan 10-50 vni 10010-10050
  vxlan vrf VRF-TENANT-A vni 50001
!
ip virtual-router mac-address
00:1c:73:00:00:99
ip address virtual source-nat vrf
VRF-TENANT-A address 201.0.0.101
```

Leaf-2:

```
vlan 40
  name VLAN-40
!
vrf instance VRF-TENANT-A
!
interface Ethernet12
  description TENANT-A-VLAN-40
  switchport access vlan 40
!
interface Loopback0
  description EVPN Peering Source
  ip address 1.1.1.102/32
!
interface Loopback1
  description VXLAN Tunnel Source
  ip address 2.2.2.102/32
!
interface Loopback10
  vrf VRF-TENANT-A
  ip address 201.0.0.102/32
!
interface Vlan40
  vrf VRF-TENANT-A
  ip address 192.168.40.250/24
!
interface Vxlan1
  vxlan source-interface Loopback1
  vxlan udp-port 4789
  vxlan vlan 10-50 vni 10010-10050
  vxlan vrf VRF-TENANT-A vni 50001
!
ip virtual-router mac-address
00:1c:73:00:00:99
ip address virtual source-nat vrf
VRF-TENANT-A address 201.0.0.102
```

Leaf-1:

```
!  
ip routing vrf VRF-TENANT-A  
!  
router bgp 65001  
!  
  vlan-aware-bundle TENANT-A  
    rd 1.1.1.101:1  
    route-target both 1:1  
    redistribute learned  
    vlan 10-50  
  !  
  vrf VRF-TENANT-A  
    rd 1.1.1.101:1  
    route-target import evpn 2:2  
    route-target export evpn 2:2  
    redistribute connected  
!
```

Leaf-2:

```
!  
ip routing vrf VRF-TENANT-A  
!  
router bgp 65002  
!  
  vlan-aware-bundle TENANT-A  
    rd 1.1.1.102:1  
    route-target both 1:1  
    redistribute learned  
    vlan 10-50  
  !  
  vrf VRF-TENANT-A  
    rd 1.1.1.102:1  
    route-target import evpn 2:2  
    route-target export evpn 2:2  
    redistribute connected  
!
```

Leaf-3:

```

vlan 30
  name VLAN-30
!
vlan 50
  name VLAN-50
!
vlan 4000
  name MLAG_ibGP_VRF-TENANT-A
!
vlan 4093
  name MLAG_ibGP
!
vlan 4094
  name MLAG-Interlink-VLAN
  trunk group MLAG-Peer
!
vrf instance VRF-TENANT-A
!
interface Port-Channel1
  switchport trunk allowed vlan 30,50
  switchport mode trunk
  mlag 1
!
interface Port-Channel1000
  switchport mode trunk
  switchport trunk group MLAG-Peer
!
interface Ethernet3
  channel-group 1 mode active
!
interface Ethernet5
  description MLAG-Interface
  channel-group 1000 mode active
!
interface Ethernet6
  description MLAG-Interface
  channel-group 1000 mode active
!
interface Loopback0
  description EVPN Peering Source
  ip address 1.1.1.103/32
!
interface Loopback1
  description VXLAN Tunnel Source
  ip address 2.2.2.113/32
!
interface Loopback10
  vrf VRF-TENANT-A
  ip address 201.0.0.103/32
!

```

Leaf-4:

```

vlan 30
  name VLAN-30
!
vlan 50
  name VLAN-50
!
vlan 4000
  name MLAG_ibGP_VRF-TENANT-A
!
vlan 4093
  name MLAG_ibGP
!
vlan 4094
  name MLAG-Interlink-VLAN
  trunk group MLAG-Peer
!
vrf instance VRF-TENANT-A
!
interface Port-Channel1
  switchport trunk allowed vlan 30,50
  switchport mode trunk
  mlag 1
!
interface Port-Channel1000
  switchport mode trunk
  switchport trunk group MLAG-Peer
!
interface Ethernet3
  channel-group 1 mode active
!
interface Ethernet5
  description MLAG-Interface
  channel-group 1000 mode active
!
interface Ethernet6
  description MLAG-Interface
  channel-group 1000 mode active
!
interface Loopback0
  description EVPN Peering Source
  ip address 1.1.1.104/32
!
interface Loopback1
  description VXLAN Tunnel Source
  ip address 2.2.2.113/32
!
interface Loopback10
  vrf VRF-TENANT-A
  ip address 201.0.0.104/32
!

```

Leaf-3:

```

interface Vlan30
  vrf VRF-TENANT-A
  ip address 192.168.30.250/24
!
interface Vlan50
  vrf VRF-TENANT-A
  ip address 192.168.50.250/24
!
interface Vlan4000
  description MLAG iBGP Peering
  VRF-TENANT-A
  vrf VRF-TENANT-A
  ip address 192.0.0.1/24
!
interface Vlan4093
  description MLAG iBGP Peering
  ip address 192.0.0.1/24
!
interface Vlan4094
  description MLAG-Interlink-IP
  no autostate
  ip address 10.10.10.1/24
!
interface Vxlan1
  vxlan source-interface Loopback1
  vxlan virtual-router encapsulation
  mac-address mlag-system-id
  vxlan udp-port 4789
  vxlan vlan 10-50 vni 10010-10050
  vxlan vrf VRF-TENANT-A vni 50001
!
ip virtual-router mac-address
00:1c:73:00:00:99
ip address virtual source-nat vrf
VRF-TENANT-A address 201.0.0.103
!
ip routing vrf VRF-TENANT-A
!
router bgp 65013
!
  vlan-aware-bundle TENANT-A
    rd 1.1.1.103:1
    route-target both 1:1
    redistribute learned
    vlan 10-50
  !
vrf VRF-TENANT-A
  rd 1.1.1.103:1
  route-target import evpn 2:2
  route-target export evpn 2:2
  neighbor 192.0.0.2 peer group

```

Leaf-4:

```

interface Vlan30
  vrf VRF-TENANT-A
  ip address 192.168.30.250/24
!
interface Vlan50
  vrf VRF-TENANT-A
  ip address 192.168.50.250/24
!
interface Vlan4000
  description MLAG iBGP Peering
  VRF-TENANT-A
  vrf VRF-TENANT-A
  ip address 192.0.0.2/24
!
interface Vlan4093
  description MLAG iBGP Peering
  ip address 192.0.0.2/24
!
interface Vlan4094
  description MLAG-Interlink-IP
  no autostate
  ip address 10.10.10.2/24
!
interface Vxlan1
  vxlan source-interface Loopback1
  vxlan virtual-router encapsulation
  mac-address mlag-system-id
  vxlan udp-port 4789
  vxlan vlan 10-50 vni 10010-10050
  vxlan vrf VRF-TENANT-A vni 50001
!
ip virtual-router mac-address
00:1c:73:00:00:99
ip address virtual source-nat vrf
VRF-TENANT-A address 201.0.0.104
!
ip routing vrf VRF-TENANT-A
!
router bgp 65013
!
  vlan-aware-bundle TENANT-A
    rd 1.1.1.104:1
    route-target both 1:1
    redistribute learned
    vlan 10-50
  !
vrf VRF-TENANT-A
  rd 1.1.1.104:1
  route-target import evpn 2:2
  route-target export evpn 2:2
  neighbor 192.0.0.1 peer group

```

Leaf-3:

```
MLAG-IPv4-UNDERLAY-PEER
    redistribute connected
!
```

Leaf-4:

```
MLAG-IPv4-UNDERLAY-PEER
    redistribute connected
!
```

Switch-1:

```
hostname Switch-1
!
spanning-tree mode mstp
!
vlan 30
    name VLAN-30
!
vlan 50
    name VLAN-50
!
interface Port-Channel10
    switchport trunk allowed vlan 30,50
    switchport mode trunk
!
interface Ethernet3
    channel-group 10 mode active
!
interface Ethernet4
    channel-group 10 mode active
!
interface Ethernet11
    switchport access vlan 30
!
interface Ethernet12
    switchport access vlan 50
!
```

IP Fabric L3 EVPN Overlay Leaf Layer verification:

Leaf-1#show vrf

Maximum number of vrfs allowed: 1024

Vrf	RD	Protocols	State	Interfaces
VRF-TENANT-A	1.1.1.101:1	ipv4,ipv6	v4:routing,	Loopback10, Vlan30, Vlan4094 v6:no routing
MGMT	<not set>	ipv4,ipv6	v4:no routing,	Management1 v6:no routing

Leaf-3#show vrf

Maximum number of vrfs allowed: 1024

Vrf	RD	Protocols	State	Interfaces
VRF-TENANT-A	1.1.1.103:1	ipv4,ipv6	v4:routing,	Loopback10, Vlan30, Vlan4000, Vlan4092, Vlan50 v6:no routing
MGMT	<not set>	ipv4,ipv6	v4:no routing,	Management1 v6:no routing

Leaf-1#show interfaces vxlan 1

Vxlan1 is up, line protocol is up (connected)

Hardware is Vxlan

Source interface is Loopback1 and is active with 2.2.2.101

Replication/Flood Mode is headend with Flood List Source: EVPN

Remote MAC learning via EVPN

VNI mapping to VLANs

Static VLAN to VNI mapping is

[10, 10010]	[11, 10011]	[12, 10012]	[13, 10013]
[14, 10014]	[15, 10015]	[16, 10016]	[17, 10017]
[18, 10018]	[19, 10019]	[20, 10020]	[21, 10021]
[22, 10022]	[23, 10023]	[24, 10024]	[25, 10025]
[26, 10026]	[27, 10027]	[28, 10028]	[29, 10029]
[30, 10030]	[31, 10031]	[32, 10032]	[33, 10033]
[34, 10034]	[35, 10035]	[36, 10036]	[37, 10037]
[38, 10038]	[39, 10039]	[40, 10040]	[41, 10041]
[42, 10042]	[43, 10043]	[44, 10044]	[45, 10045]
[46, 10046]	[47, 10047]	[48, 10048]	[49, 10049]
[50, 10050]			

Dynamic VLAN to VNI mapping for 'evpn' is

[4094, 50001]

Note: All Dynamic VLANs used by VCS are internal VLANs.

Use 'show vxlan vni' for details.

Static VRF to VNI mapping is

[VRF-TENANT-A, 50001]

Headend replication flood vtep list is:

10	2.2.2.113	2.2.2.102
20	2.2.2.102	
30	2.2.2.113	

MLAG Shared Router MAC is 0000.0000.0000

VTEP address mask is None

Leaf-1#show bgp evpn

BGP routing table information for VRF default

Router identifier 1.1.1.101, local AS number 65001

Route status codes: s - suppressed, * - valid, > - active, # - not installed, E - ECMP head, e - ECMP

S - Stale, c - Contributing to ECMP, b - backup

% - Pending BGP convergence

Origin codes: i - IGP, e - EGP, ? - incomplete

AS Path Attributes: Or-ID - Originator ID, C-LST - Cluster List, LL Nexthop - Link Local Nexthop

	Network	Next Hop	Metric	LocPref	Weight	Path
* >	RD: 1.1.1.101:1 mac-ip	10030 0050.7966.6800				
		-	-	-	0	i
* >	RD: 1.1.1.101:1 mac-ip	10030 0050.7966.6800	192.168.30.1			
		-	-	-	0	i
* >Ec	RD: 1.1.1.103:1 mac-ip	10030 0050.7966.6802				
		2.2.2.113	-	100	0	
66001 65013 i						
* ec	RD: 1.1.1.103:1 mac-ip	10030 0050.7966.6802				
		2.2.2.113	-	100	0	
66001 65013 i						
* >Ec	RD: 1.1.1.104:1 mac-ip	10030 0050.7966.6802				
		2.2.2.113	-	100	0	
66001 65013 i						
* ec	RD: 1.1.1.104:1 mac-ip	10030 0050.7966.6802				
		2.2.2.113	-	100	0	
66001 65013 i						
* >Ec	RD: 1.1.1.103:1 mac-ip	10030 0050.7966.6802	192.168.30.3			
		2.2.2.113	-	100	0	
66001 65013 i						
* ec	RD: 1.1.1.103:1 mac-ip	10030 0050.7966.6802	192.168.30.3			
		2.2.2.113	-	100	0	
66001 65013 i						
* >Ec	RD: 1.1.1.104:1 mac-ip	10030 0050.7966.6802	192.168.30.3			
		2.2.2.113	-	100	0	
66001 65013 i						
* ec	RD: 1.1.1.104:1 mac-ip	10030 0050.7966.6802	192.168.30.3			
		2.2.2.113	-	100	0	
66001 65013 i						
* >Ec	RD: 1.1.1.102:1 mac-ip	10040 0050.7966.6801				
		2.2.2.102	-	100	0	
66001 65002 i						
* ec	RD: 1.1.1.102:1 mac-ip	10040 0050.7966.6801				
		2.2.2.102	-	100	0	
66001 65002 i						
* >Ec	RD: 1.1.1.102:1 mac-ip	10040 0050.7966.6801	192.168.40.2			
		2.2.2.102	-	100	0	
66001 65002 i						
* ec	RD: 1.1.1.102:1 mac-ip	10040 0050.7966.6801	192.168.40.2			
		2.2.2.102	-	100	0	
66001 65002 i						
* >Ec	RD: 1.1.1.103:1 mac-ip	10050 0050.7966.6803				
		2.2.2.113	-	100	0	
66001 65013 i						
* ec	RD: 1.1.1.103:1 mac-ip	10050 0050.7966.6803				

```

                2.2.2.113          -      100      0
66001 65013 i
* >Ec  RD: 1.1.1.104:1 mac-ip 10050 0050.7966.6803
                2.2.2.113          -      100      0
66001 65013 i
* ec   RD: 1.1.1.104:1 mac-ip 10050 0050.7966.6803
                2.2.2.113          -      100      0
66001 65013 i
* >Ec  RD: 1.1.1.103:1 mac-ip 10050 0050.7966.6803 192.168.50.4
                2.2.2.113          -      100      0
66001 65013 i
* ec   RD: 1.1.1.103:1 mac-ip 10050 0050.7966.6803 192.168.50.4
                2.2.2.113          -      100      0
66001 65013 i
* >Ec  RD: 1.1.1.104:1 mac-ip 10050 0050.7966.6803 192.168.50.4
                2.2.2.113          -      100      0
66001 65013 i
* ec   RD: 1.1.1.104:1 mac-ip 10050 0050.7966.6803 192.168.50.4
                2.2.2.113          -      100      0
66001 65013 i
* >    RD: 1.1.1.101:1 imet 10010 2.2.2.101
                -                  -      -      0      i
* >Ec  RD: 1.1.1.102:1 imet 10010 2.2.2.102
                2.2.2.102          -      100      0
66001 65002 i
* ec   RD: 1.1.1.102:1 imet 10010 2.2.2.102
                2.2.2.102          -      100      0
66001 65002 i
* >Ec  RD: 1.1.1.103:1 imet 10010 2.2.2.113
                2.2.2.113          -      100      0
66001 65013 i
* ec   RD: 1.1.1.103:1 imet 10010 2.2.2.113
                2.2.2.113          -      100      0
66001 65013 i
* >Ec  RD: 1.1.1.104:1 imet 10010 2.2.2.113
                2.2.2.113          -      100      0
66001 65013 i
* ec   RD: 1.1.1.104:1 imet 10010 2.2.2.113
                2.2.2.113          -      100      0
66001 65013 i
* >    RD: 1.1.1.101:1 imet 10020 2.2.2.101
                -                  -      -      0      i
* >Ec  RD: 1.1.1.102:1 imet 10020 2.2.2.102
                2.2.2.102          -      100      0
66001 65002 i
* ec   RD: 1.1.1.102:1 imet 10020 2.2.2.102
                2.2.2.102          -      100      0
66001 65002 i
* >    RD: 1.1.1.101:1 imet 10030 2.2.2.101
                -                  -      -      0      i
* >Ec  RD: 1.1.1.103:1 imet 10030 2.2.2.113
                2.2.2.113          -      100      0
66001 65013 i
* ec   RD: 1.1.1.103:1 imet 10030 2.2.2.113
                2.2.2.113          -      100      0
66001 65013 i
* >Ec  RD: 1.1.1.104:1 imet 10030 2.2.2.113

```

		2.2.2.113	-	100	0	
66001	65013	i				
	* ec	RD: 1.1.1.104:1 imet 10030	2.2.2.113	-	100	0
			2.2.2.113			
66001	65013	i				
	* >Ec	RD: 1.1.1.102:1 imet 10040	2.2.2.102	-	100	0
			2.2.2.102			
66001	65002	i				
	* ec	RD: 1.1.1.102:1 imet 10040	2.2.2.102	-	100	0
			2.2.2.102			
66001	65002	i				
	* >Ec	RD: 1.1.1.103:1 imet 10050	2.2.2.113	-	100	0
			2.2.2.113			
66001	65013	i				
	* ec	RD: 1.1.1.103:1 imet 10050	2.2.2.113	-	100	0
			2.2.2.113			
66001	65013	i				
	* >Ec	RD: 1.1.1.104:1 imet 10050	2.2.2.113	-	100	0
			2.2.2.113			
66001	65013	i				
	* ec	RD: 1.1.1.104:1 imet 10050	2.2.2.113	-	100	0
			2.2.2.113			
66001	65013	i				
	* >Ec	RD: 1.1.1.103:1 ip-prefix 192.0.0.0/24	2.2.2.113	-	100	0
			2.2.2.113			
66001	65013	i				
	* ec	RD: 1.1.1.103:1 ip-prefix 192.0.0.0/24	2.2.2.113	-	100	0
			2.2.2.113			
66001	65013	i				
	* >Ec	RD: 1.1.1.104:1 ip-prefix 192.0.0.0/24	2.2.2.113	-	100	0
			2.2.2.113			
66001	65013	i				
	* ec	RD: 1.1.1.104:1 ip-prefix 192.0.0.0/24	2.2.2.113	-	100	0
			2.2.2.113			
66001	65013	i				
	* >	RD: 1.1.1.101:1 ip-prefix 192.168.30.0/24	-	-	0	i
	* >Ec	RD: 1.1.1.103:1 ip-prefix 192.168.30.0/24	2.2.2.113	-	100	0
			2.2.2.113			
66001	65013	i				
	* ec	RD: 1.1.1.103:1 ip-prefix 192.168.30.0/24	2.2.2.113	-	100	0
			2.2.2.113			
66001	65013	i				
	* >Ec	RD: 1.1.1.104:1 ip-prefix 192.168.30.0/24	2.2.2.113	-	100	0
			2.2.2.113			
66001	65013	i				
	* ec	RD: 1.1.1.104:1 ip-prefix 192.168.30.0/24	2.2.2.113	-	100	0
			2.2.2.113			
66001	65013	i				
	* >Ec	RD: 1.1.1.102:1 ip-prefix 192.168.40.0/24	2.2.2.102	-	100	0
			2.2.2.102			
66001	65002	i				
	* ec	RD: 1.1.1.102:1 ip-prefix 192.168.40.0/24	2.2.2.102	-	100	0
			2.2.2.102			
66001	65002	i				
	* >Ec	RD: 1.1.1.103:1 ip-prefix 192.168.50.0/24	2.2.2.113	-	100	0
			2.2.2.113			
66001	65013	i				

```

* ec RD: 1.1.1.103:1 ip-prefix 192.168.50.0/24
                2.2.2.113 - 100 0
66001 65013 i
* >Ec RD: 1.1.1.104:1 ip-prefix 192.168.50.0/24
                2.2.2.113 - 100 0
66001 65013 i
* ec RD: 1.1.1.104:1 ip-prefix 192.168.50.0/24
                2.2.2.113 - 100 0
66001 65013 i
* > RD: 1.1.1.101:1 ip-prefix 201.0.0.101/32
                - - 0 i
* >Ec RD: 1.1.1.102:1 ip-prefix 201.0.0.102/32
                2.2.2.102 - 100 0
66001 65002 i
* ec RD: 1.1.1.102:1 ip-prefix 201.0.0.102/32
                2.2.2.102 - 100 0
66001 65002 i
* >Ec RD: 1.1.1.103:1 ip-prefix 201.0.0.103/32
                2.2.2.113 - 100 0
66001 65013 i
* ec RD: 1.1.1.103:1 ip-prefix 201.0.0.103/32
                2.2.2.113 - 100 0
66001 65013 i
* >Ec RD: 1.1.1.104:1 ip-prefix 201.0.0.103/32
                2.2.2.113 - 100 0
66001 65013 i
* ec RD: 1.1.1.104:1 ip-prefix 201.0.0.103/32
                2.2.2.113 - 100 0
66001 65013 i
* >Ec RD: 1.1.1.103:1 ip-prefix 201.0.0.104/32
                2.2.2.113 - 100 0
66001 65013 i
* ec RD: 1.1.1.103:1 ip-prefix 201.0.0.104/32
                2.2.2.113 - 100 0
66001 65013 i
* >Ec RD: 1.1.1.104:1 ip-prefix 201.0.0.104/32
                2.2.2.113 - 100 0
66001 65013 i
* ec RD: 1.1.1.104:1 ip-prefix 201.0.0.104/32
                2.2.2.113 - 100 0
66001 65013 i
Leaf-1#

```

```

Leaf-1#show bgp evpn route-type mac-ip detail
BGP routing table information for VRF default
Router identifier 1.1.1.101, local AS number 65001
BGP routing table entry for mac-ip 10030 0050.7966.6800, Route Distinguisher:
1.1.1.101:1
  Paths: 1 available
  Local
    - from - (0.0.0.0)
      Origin IGP, metric -, localpref -, weight 0, valid, local, best
      Extended Community: Route-Target-AS:1:1 TunnelEncap:tunnelTypeVxlan
      VNI: 10030 ESI: 0000:0000:0000:0000:0000
BGP routing table entry for mac-ip 10030 0050.7966.6800 192.168.30.1, Route
Distinguisher: 1.1.1.101:1
  Paths: 1 available

```

Local

```
- from - (0.0.0.0)
  Origin IGP, metric -, localpref -, weight 0, valid, local, best
  Extended Community: Route-Target-AS:1:1 Route-Target-AS:2:2
TunnelEncap:tunnelTypeVxlan EvpnRouterMac:0c:7e:d1:c1:92:48
  VNI: 10030 L3 VNI: 50001 ESI: 0000:0000:0000:0000:0000
BGP routing table entry for mac-ip 10030 0050.7966.6802, Route Distinguisher:
1.1.1.103:1
  Paths: 2 available
  66001 65013
    2.2.2.113 from 1.1.1.202 (1.1.1.202)
      Origin IGP, metric -, localpref 100, weight 0, valid, external, ECMP head,
      ECMP, best, ECMP contributor
      Extended Community: Route-Target-AS:1:1 TunnelEncap:tunnelTypeVxlan
      VNI: 10030 ESI: 0000:0000:0000:0000:0000
    66001 65013
      2.2.2.113 from 1.1.1.201 (1.1.1.201)
        Origin IGP, metric -, localpref 100, weight 0, valid, external, ECMP, ECMP
        contributor
        Extended Community: Route-Target-AS:1:1 TunnelEncap:tunnelTypeVxlan
        VNI: 10030 ESI: 0000:0000:0000:0000:0000
BGP routing table entry for mac-ip 10030 0050.7966.6802, Route Distinguisher:
1.1.1.104:1
  Paths: 2 available
  66001 65013
    2.2.2.113 from 1.1.1.201 (1.1.1.201)
      Origin IGP, metric -, localpref 100, weight 0, valid, external, ECMP head,
      ECMP, best, ECMP contributor
      Extended Community: Route-Target-AS:1:1 TunnelEncap:tunnelTypeVxlan
      VNI: 10030 ESI: 0000:0000:0000:0000:0000
    66001 65013
      2.2.2.113 from 1.1.1.202 (1.1.1.202)
        Origin IGP, metric -, localpref 100, weight 0, valid, external, ECMP, ECMP
        contributor
        Extended Community: Route-Target-AS:1:1 TunnelEncap:tunnelTypeVxlan
        VNI: 10030 ESI: 0000:0000:0000:0000:0000
BGP routing table entry for mac-ip 10030 0050.7966.6802 192.168.30.3, Route
Distinguisher: 1.1.1.103:1
  Paths: 2 available
  66001 65013
    2.2.2.113 from 1.1.1.202 (1.1.1.202)
      Origin IGP, metric -, localpref 100, weight 0, valid, external, ECMP head,
      ECMP, best, ECMP contributor
      Extended Community: Route-Target-AS:1:1 Route-Target-AS:2:2
      TunnelEncap:tunnelTypeVxlan EvpnRouterMac:0e:7e:d1:2c:c9:a1
      VNI: 10030 L3 VNI: 50001 ESI: 0000:0000:0000:0000:0000
    66001 65013
      2.2.2.113 from 1.1.1.201 (1.1.1.201)
        Origin IGP, metric -, localpref 100, weight 0, valid, external, ECMP, ECMP
        contributor
        Extended Community: Route-Target-AS:1:1 Route-Target-AS:2:2
        TunnelEncap:tunnelTypeVxlan EvpnRouterMac:0e:7e:d1:2c:c9:a1
        VNI: 10030 L3 VNI: 50001 ESI: 0000:0000:0000:0000:0000
BGP routing table entry for mac-ip 10030 0050.7966.6802 192.168.30.3, Route
Distinguisher: 1.1.1.104:1
  Paths: 2 available
  66001 65013
    2.2.2.113 from 1.1.1.201 (1.1.1.201)
```

Origin IGP, metric -, localpref 100, weight 0, valid, external, ECMP head, ECMP, best, ECMP contributor
 Extended Community: Route-Target-AS:1:1 Route-Target-AS:2:2
 TunnelEncap:tunnelTypeVxlan EvpnRouterMac:0e:7e:d1:2c:c9:a1
VNI: 10030 L3 VNI: 50001 ESI: 0000:0000:0000:0000:0000
 66001 65013
 2.2.2.113 from 1.1.1.202 (1.1.1.202)
 Origin IGP, metric -, localpref 100, weight 0, valid, external, ECMP, ECMP contributor
 Extended Community: Route-Target-AS:1:1 Route-Target-AS:2:2
 TunnelEncap:tunnelTypeVxlan EvpnRouterMac:0e:7e:d1:2c:c9:a1
VNI: 10030 L3 VNI: 50001 ESI: 0000:0000:0000:0000:0000
 BGP routing table entry for mac-ip 10040 0050.7966.6801, Route Distinguisher: 1.1.1.102:1
 Paths: 2 available
 66001 65002
 2.2.2.102 from 1.1.1.202 (1.1.1.202)
 Origin IGP, metric -, localpref 100, weight 0, valid, external, ECMP head, ECMP, best, ECMP contributor
 Extended Community: Route-Target-AS:1:1 TunnelEncap:tunnelTypeVxlan
VNI: 10040 ESI: 0000:0000:0000:0000:0000
 66001 65002
 2.2.2.102 from 1.1.1.201 (1.1.1.201)
 Origin IGP, metric -, localpref 100, weight 0, valid, external, ECMP, ECMP contributor
 Extended Community: Route-Target-AS:1:1 TunnelEncap:tunnelTypeVxlan
VNI: 10040 ESI: 0000:0000:0000:0000:0000
 BGP routing table entry for mac-ip **10040 0050.7966.6801 192.168.40.2**, Route Distinguisher: 1.1.1.102:1
 Paths: 2 available
 66001 65002
 2.2.2.102 from 1.1.1.202 (1.1.1.202)
 Origin IGP, metric -, localpref 100, weight 0, valid, external, ECMP head, ECMP, best, ECMP contributor
 Extended Community: Route-Target-AS:1:1 Route-Target-AS:2:2
 TunnelEncap:tunnelTypeVxlan EvpnRouterMac:0c:7e:d1:b5:c5:bb
VNI: 10040 L3 VNI: 50001 ESI: 0000:0000:0000:0000:0000
 66001 65002
 2.2.2.102 from 1.1.1.201 (1.1.1.201)
 Origin IGP, metric -, localpref 100, weight 0, valid, external, ECMP, ECMP contributor
 Extended Community: Route-Target-AS:1:1 Route-Target-AS:2:2
 TunnelEncap:tunnelTypeVxlan EvpnRouterMac:0c:7e:d1:b5:c5:bb
VNI: 10040 L3 VNI: 50001 ESI: 0000:0000:0000:0000:0000
 BGP routing table entry for mac-ip 10050 0050.7966.6803, Route Distinguisher: 1.1.1.103:1
 Paths: 2 available
 66001 65013
 2.2.2.113 from 1.1.1.202 (1.1.1.202)
 Origin IGP, metric -, localpref 100, weight 0, valid, external, ECMP head, ECMP, best, ECMP contributor
 Extended Community: Route-Target-AS:1:1 TunnelEncap:tunnelTypeVxlan
VNI: 10050 ESI: 0000:0000:0000:0000:0000
 66001 65013
 2.2.2.113 from 1.1.1.201 (1.1.1.201)
 Origin IGP, metric -, localpref 100, weight 0, valid, external, ECMP, ECMP contributor
 Extended Community: Route-Target-AS:1:1 TunnelEncap:tunnelTypeVxlan

VNI: 10050 ESI: 0000:0000:0000:0000:0000

BGP routing table entry for mac-ip 10050 0050.7966.6803, Route Distinguisher: 1.1.1.104:1

Paths: 2 available

66001 65013

2.2.2.113 from 1.1.1.201 (1.1.1.201)

Origin IGP, metric -, localpref 100, weight 0, valid, external, ECMP head, ECMP, best, ECMP contributor

Extended Community: Route-Target-AS:1:1 TunnelEncap:tunnelTypeVxlan

VNI: 10050 ESI: 0000:0000:0000:0000:0000

66001 65013

2.2.2.113 from 1.1.1.202 (1.1.1.202)

Origin IGP, metric -, localpref 100, weight 0, valid, external, ECMP, ECMP contributor

Extended Community: Route-Target-AS:1:1 TunnelEncap:tunnelTypeVxlan

VNI: 10050 ESI: 0000:0000:0000:0000:0000

BGP routing table entry for mac-ip 10050 0050.7966.6803 192.168.50.4, Route Distinguisher: 1.1.1.103:1

Paths: 2 available

66001 65013

2.2.2.113 from 1.1.1.202 (1.1.1.202)

Origin IGP, metric -, localpref 100, weight 0, valid, external, ECMP head, ECMP, best, ECMP contributor

Extended Community: Route-Target-AS:1:1 Route-Target-AS:2:2

TunnelEncap:tunnelTypeVxlan EvpnRouterMac:0e:7e:d1:2c:c9:a1

VNI: 10050 L3 VNI: 50001 ESI: 0000:0000:0000:0000:0000

66001 65013

2.2.2.113 from 1.1.1.201 (1.1.1.201)

Origin IGP, metric -, localpref 100, weight 0, valid, external, ECMP, ECMP contributor

Extended Community: Route-Target-AS:1:1 Route-Target-AS:2:2

TunnelEncap:tunnelTypeVxlan EvpnRouterMac:0e:7e:d1:2c:c9:a1

VNI: 10050 L3 VNI: 50001 ESI: 0000:0000:0000:0000:0000

BGP routing table entry for mac-ip 10050 0050.7966.6803 192.168.50.4, Route Distinguisher: 1.1.1.104:1

Paths: 2 available

66001 65013

2.2.2.113 from 1.1.1.201 (1.1.1.201)

Origin IGP, metric -, localpref 100, weight 0, valid, external, ECMP head, ECMP, best, ECMP contributor

Extended Community: Route-Target-AS:1:1 Route-Target-AS:2:2

TunnelEncap:tunnelTypeVxlan EvpnRouterMac:0e:7e:d1:2c:c9:a1

VNI: 10050 L3 VNI: 50001 ESI: 0000:0000:0000:0000:0000

66001 65013

2.2.2.113 from 1.1.1.202 (1.1.1.202)

Origin IGP, metric -, localpref 100, weight 0, valid, external, ECMP, ECMP contributor

Extended Community: Route-Target-AS:1:1 Route-Target-AS:2:2

TunnelEncap:tunnelTypeVxlan EvpnRouterMac:0e:7e:d1:2c:c9:a1

VNI: 10050 L3 VNI: 50001 ESI: 0000:0000:0000:0000:0000

IPv6 BGP unnumbered for Fabric Underlay

When using BGP unnumbered the Interface **IPv6 Link Local Address (LLA)** is automatically allocated for Fabric interconnects based on eui64 from an **fe80::/10** range. Thus, the IPv4 address range planning for Fabric interconnects is not necessary and IPv4 Loopbacks for EVPN and shared VTEP IP addresses are resolved over IPv6 next hops.

Leaf-1:

```
interface Ethernet1
  description Spine-1
  no switchport
  ipv6 enable
!
interface Ethernet2
  description Spine-2
  no switchport
  ipv6 enable
!
ip routing ipv6 interfaces
!
ipv6 unicast-routing
!
router bgp 65001
  neighbor IPv6-UNDERLAY-SPINES peer
  group
  neighbor IPv6-UNDERLAY-SPINES
  password secretpw
  neighbor IPv6-UNDERLAY-SPINES
  send-community
  neighbor IPv6-UNDERLAY-SPINES
  maximum-routes 12000
  redistribute connected
  neighbor interface Ethernet1
  peer-group IPv6-UNDERLAY-SPINES
  remote-as 66001
  neighbor interface Ethernet2
  peer-group IPv6-UNDERLAY-SPINES
  remote-as 66001
!
  address-family ipv4
    no neighbor EVPN-OVERLAY-PEERS
  activate
  neighbor IPv6-UNDERLAY-SPINES
  next-hop address-family ipv6 originate
!
```

Spine-1:

```
interface Ethernet1
  description Leaf-1
  no switchport
  ipv6 enable
!
interface Ethernet2
  description Leaf-2
  no switchport
  ipv6 enable
!
interface Ethernet3
  description Leaf-3
  no switchport
  ipv6 enable
!
interface Ethernet4
  description Leaf-4
  no switchport
  ipv6 enable
!
ip routing ipv6 interfaces
!
ipv6 unicast-routing
!
peer-filter LEAF-AS-RANGE
  10 match as-range 65001-65256
  result accept
!
router bgp 66001
  bgp listen range fe80::/10
  peer-group IPv4-UNDERLAY-PEERS
  peer-filter LEAF-AS-RANGE
  neighbor IPv6-UNDERLAY-LEAFS peer
  group
  neighbor IPv6-UNDERLAY-LEAFS
  password secretpw
  neighbor IPv6-UNDERLAY-LEAFS
  send-community
```

Spine-1:

```
    neighbor IPv6-UNDERLAY-LEAFS
maximum-routes 12000
    redistribute connected
    !
    address-family ipv4
        no neighbor EVPN-OVERLAY-PEERS
activate
        neighbor IPv6-UNDERLAY-LEAFS
next-hop address-family ipv6 originate
    !
```

Automated IP Fabric setup via Ansible AVD

CloudVision can be automated with the help of Ansible Playbooks provided by Arista. Especially **Arista AVD** provides a solution which supports automation of BGP EVPN based Fabrics. Further information can be found at:

<https://github.com/aristanetworks/ansible-avd>

Within this Manual we will be using an external **Ubuntu Server 20.04.2** and **CloudVision 2020.2.4**
Please make sure that a DNS Server is configured on the Ubuntu Server (e.g. 8.8.8.8 from Google) and that you can reach GitHub!

It may be useful to **enable SSH root login on Ubuntu**, especially when using tools like Visual Studio to edit .yaml files on the Ubuntu Server.

Here is the procedure where you have to edit the sshd_config file and then create a root password:

```
mkrausch@ubuntu:~$ sudo su
[sudo] password for mkrausch: *****
root@ubuntu:/home/mkrausch# nano /etc/ssh/sshd_config
```

```
*****
```

```
...
```

```
# Ciphers and keying
#RekeyLimit default none
```

```
# Logging
#SyslogFacility AUTH
#LogLevel INFO
```

```
# Authentication:
```

```
#LoginGraceTime 2m
#PermitRootLogin prohibit-password
```

```
PermitRootLogin yes
```

```
#StrictModes yes
```

```
#MaxAuthTries 6
```

```
#MaxSessions 10
```

```
...
```

```
*****
```

```
root@ubuntu:/home/mkrausch# systemctl restart ssh
```

```
root@ubuntu:/home/mkrausch# passwd
```

```
New password: *****
```

```
Retype new password: *****
```

```
passwd: password updated successfully
```

```
root@ubuntu:/home/mkrausch#
```

Installing Python3-pip

```
mkrausch@ubuntu:~$ sudo su
[sudo] password for mkrausch: *****
root@ubuntu:/home/mkrausch# apt install python3-pip
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  binutils binutils-common binutils-x86-64-linux-gnu build-essential dpkg-dev
  fakeroot g++ g++-9 gcc gcc-9 libalgorithm-diff-perl
  libalgorithm-diff-xs-perl libalgorithm-merge-perl libasan5 libatomic1
  libbinutils libc-dev-bin libc6-dev libcrypt-dev libctf-nobfd0 libctf0
  libexpat1-dev libfakeroot libgcc-9-dev libitm1 liblsan0 libpython3-dev
  libpython3.8-dev libquadmath0 libstdc++-9-dev libtsan0 libubsan1
  linux-libc-dev make manpages-dev python-pip-whl python3-dev python3-wheel
  python3.8-dev zlib1g-dev
Suggested packages:
  binutils-doc debian-keyring g++-multilib g++-9-multilib gcc-9-doc
  gcc-multilib autoconf automake libtool flex bison gcc-doc gcc-9-multilib
  gcc-9-locales glibc-doc libstdc++-9-doc make-doc
The following NEW packages will be installed:
  binutils binutils-common binutils-x86-64-linux-gnu build-essential dpkg-dev
  fakeroot g++ g++-9 gcc gcc-9 libalgorithm-diff-perl
  libalgorithm-diff-xs-perl libalgorithm-merge-perl libasan5 libatomic1
  libbinutils libc-dev-bin libc6-dev libcrypt-dev libctf-nobfd0 libctf0
  libexpat1-dev libfakeroot libgcc-9-dev libitm1 liblsan0 libpython3-dev
  libpython3.8-dev libquadmath0 libstdc++-9-dev libtsan0 libubsan1
  linux-libc-dev make manpages-dev python-pip-whl python3-dev python3-pip
  python3-wheel python3.8-dev zlib1g-dev
0 upgraded, 41 newly installed, 0 to remove and 31 not upgraded.
Need to get 41.2 MB of archives.
After this operation, 182 MB of additional disk space will be used.
Do you want to continue? [Y/n] y
Get:1 http://de.archive.ubuntu.com/ubuntu focal-updates/main amd64 binutils-common
amd64 2.34-6ubuntu1.3 [207 kB]
Get:2 http://de.archive.ubuntu.com/ubuntu focal-updates/main amd64 libbinutils
amd64 2.34-6ubuntu1.3 [474 kB]
...
Setting up g++ (4:9.3.0-1ubuntu2) ...
update-alternatives: using /usr/bin/g++ to provide /usr/bin/c++ (c++) in auto mode
Setting up build-essential (12.8ubuntu1.1) ...
Setting up python3-dev (3.8.2-0ubuntu2) ...
Processing triggers for man-db (2.9.1-1) ...
Processing triggers for libc-bin (2.31-0ubuntu9.2) ...
root@ubuntu:/home/mkrausch#
```

Installing Ansible and additional Requirements

```
root@ubuntu:/home/mkrausch# cd Downloads/  
root@ubuntu:/home/mkrausch/Downloads# ls  
root@ubuntu:/home/mkrausch/Downloads# nano requirements.txt
```

```
*****
```

```
netaddr>=0.7.19  
Jinja2>=2.11.3  
treelib>=1.5.5  
cvprac>=1.0.7  
paramiko>=2.7.1  
jsonschema>=3.2.0  
requests>=2.25.1  
PyYAML>=5.4.1  
md-toc>=7.1.0
```

```
*****
```

```
root@ubuntu:/home/mkrausch/Downloads# pip3 install ansible==2.11.3  
Collecting ansible-core==2.11.3  
  Downloading ansible-core-2.11.3.tar.gz (6.8 MB)  
    |████████████████████████████████████████████████████████████████████████████████| 6.8 MB 21.9 MB/s  
Requirement already satisfied: PyYAML in /usr/lib/python3/dist-packages (from  
ansible-core==2.11.3) (5.3.1)  
Requirement already satisfied: cryptography in /usr/lib/python3/dist-packages  
(from ansible-core==2.11.3) (2.8)  
Requirement already satisfied: jinja2 in /usr/lib/python3/dist-packages (from  
ansible-core==2.11.3) (2.10.1)  
Collecting packaging  
  Downloading packaging-21.2-py3-none-any.whl (40 kB)  
    |████████████████████████████████████████████████████████████████████████████████| 40 kB 11.6 MB/s  
Collecting resolvelib<0.6.0,>=0.5.3  
  Downloading resolvelib-0.5.4-py2.py3-none-any.whl (12 kB)  
Collecting pyparsing<3,>=2.0.2  
  Downloading pyparsing-2.4.7-py2.py3-none-any.whl (67 kB)  
    |████████████████████████████████████████████████████████████████████████████████| 67 kB 11.4 MB/s  
Building wheels for collected packages: ansible-core  
  Building wheel for ansible-core (setup.py) ... done  
  Created wheel for ansible-core: filename=ansible_core-2.11.3-py3-none-any.whl  
size=1955093  
sha256=10fa9b74aed18797f8b7cd5a3d6cff55ab93d0f94e26f0304c1749dc3ddf6efd  
  Stored in directory:  
/root/.cache/pip/wheels/ed/b9/f5/d582647a0dd03283f688808d476bfbd39de23c957fc54428d  
f  
Successfully built ansible-core  
Installing collected packages: pyparsing, packaging, resolvelib, ansible-core  
Successfully installed ansible-core-2.11.3 packaging-21.2 pyparsing-2.4.7  
resolvelib-0.5.4  
root@ubuntu:/home/mkrausch/Downloads#
```

```

root@ubuntu:/home/mkrausch/Downloads# pip3 install -r requirements.txt
Collecting netaddr>=0.7.19
  Downloading netaddr-0.8.0-py2.py3-none-any.whl (1.9 MB)
  |████████████████████████████████████████| 1.9 MB 435 kB/s
Collecting Jinja2>=2.11.3
  Downloading Jinja2-3.0.3-py3-none-any.whl (133 kB)
  |████████████████████████████████████████| 133 kB 21.4 MB/s
Collecting treelib>=1.5.5
  Downloading treelib-1.6.1.tar.gz (24 kB)
Collecting cvprac>=1.0.7
  Downloading cvprac-1.0.7.tar.gz (4.2 MB)
  |████████████████████████████████████████| 4.2 MB 16.5 MB/s
Collecting paramiko>=2.7.1
  Downloading paramiko-2.8.0-py2.py3-none-any.whl (206 kB)
  |████████████████████████████████████████| 206 kB 43.3 MB/s
Requirement already satisfied: jsonschema>=3.2.0 in /usr/lib/python3/dist-packages
(from -r requirements.txt (line 6)) (3.2.0)
Collecting requests>=2.25.1
  Downloading requests-2.26.0-py2.py3-none-any.whl (62 kB)
  |████████████████████████████████████████| 62 kB 2.8 MB/s
Collecting PyYAML>=5.4.1
  Downloading
PyYAML-6.0-cp38-cp38-manylinux_2_5_x86_64_manylinux1_x86_64_manylinux_2_12_x86_64_
manylinux2010_x86_64.whl (701 kB)
  |████████████████████████████████████████| 701 kB 49.6 MB/s
Collecting md-toc>=7.1.0
  Downloading md_toc-8.0.1-py3-none-any.whl (44 kB)
  |████████████████████████████████████████| 44 kB 7.4 MB/s
Collecting MarkupSafe>=2.0
  Downloading MarkupSafe-2.0.1-cp38-cp38-manylinux2010_x86_64.whl (30 kB)
Requirement already satisfied: future in /usr/lib/python3/dist-packages (from
treelib>=1.5.5->-r requirements.txt (line 3)) (0.18.2)
Requirement already satisfied: cryptography>=2.5 in /usr/lib/python3/dist-packages
(from paramiko>=2.7.1->-r requirements.txt (line 5)) (2.8)
Requirement already satisfied: bcrypt>=3.1.3 in /usr/lib/python3/dist-packages
(from paramiko>=2.7.1->-r requirements.txt (line 5)) (3.1.7)
...
Can't uninstall 'paramiko'. No files were found to uninstall.
Attempting uninstall: PyYAML
  Found existing installation: PyYAML 5.3.1
  Not uninstalling pyyaml at /usr/lib/python3/dist-packages, outside environment
/usr
  Can't uninstall 'PyYAML'. No files were found to uninstall.
Successfully installed Jinja2-3.0.3 MarkupSafe-2.0.1 PyYAML-6.0 atomicwrites-1.4.0
charset-normalizer-2.0.7 cvprac-1.0.7 fpyutils-2.0.1 md-toc-8.0.1 netaddr-0.8.0
paramiko-2.8.0 requests-2.26.0 treelib-1.6.1
root@ubuntu:/home/mkrausch/Downloads#

```

Cloning an Arista Demo Repository from Github

```
root@ubuntu:/home/mkrausch/Downloads# cd ..
root@ubuntu:/home/mkrausch# git clone
https://github.com/arista-netdevops-community/ansible-avd-cloudvision-demo.git
Cloning into 'ansible-avd-cloudvision-demo'...
remote: Enumerating objects: 1147, done.
remote: Counting objects: 100% (725/725), done.
remote: Compressing objects: 100% (354/354), done.
remote: Total 1147 (delta 396), reused 621 (delta 310), pack-reused 422
Receiving objects: 100% (1147/1147), 2.54 MiB | 2.26 MiB/s, done.
Resolving deltas: 100% (629/629), done.
root@ubuntu:/home/mkrausch#
```

Installing Arista AVD

```
root@ubuntu:/home/mkrausch# cd ansible-avd-cloudvision-demo/
root@ubuntu:/home/mkrausch/ansible-avd-cloudvision-demo# ansible-galaxy collection
install arista.avd
[DEPRECATION WARNING]: COMMAND_WARNINGS option, the command warnings feature is
being removed. This feature will be
removed from ansible-core in version 2.14. Deprecation warnings can be disabled by
setting deprecation_warnings=False in
ansible.cfg.
Starting galaxy collection install process
Process install dependency map
Starting collection install process
Downloading https://galaxy.ansible.com/download/arista-avd-3.0.0.tar.gz to
/root/.ansible/tmp/ansible-local-9426swpai0co/tmpdeahs_cg/arista-avd-3.0.0-pw6277d
p
Installing 'arista.avd:3.0.0' to
'/root/.ansible/collections/ansible_collections/arista/avd'
Downloading https://galaxy.ansible.com/download/arista-cvp-3.2.0.tar.gz to
/root/.ansible/tmp/ansible-local-9426swpai0co/tmpdeahs_cg/arista-cvp-3.2.0-yx6gjmh
t
arista.avd:3.0.0 was installed successfully
Installing 'arista.cvp:3.2.0' to
'/root/.ansible/collections/ansible_collections/arista/cvp'
Downloading https://galaxy.ansible.com/download/arista-eos-3.1.0.tar.gz to
/root/.ansible/tmp/ansible-local-9426swpai0co/tmpdeahs_cg/arista-eos-3.1.0-7s4i7sr
3
arista.cvp:3.2.0 was installed successfully
Installing 'arista.eos:3.1.0' to
'/root/.ansible/collections/ansible_collections/arista/eos'
Downloading https://galaxy.ansible.com/download/ansible-netcommon-2.4.0.tar.gz to
/root/.ansible/tmp/ansible-local-9426swpai0co/tmpdeahs_cg/ansible-netcommon-2.4.0-
ie61ufnc
arista.eos:3.1.0 was installed successfully
Installing 'ansible.netcommon:2.4.0' to
'/root/.ansible/collections/ansible_collections/ansible/netcommon'
Downloading https://galaxy.ansible.com/download/ansible-utils-2.4.2.tar.gz to
/root/.ansible/tmp/ansible-local-9426swpai0co/tmpdeahs_cg/ansible-utils-2.4.2-4law
81dh
ansible.netcommon:2.4.0 was installed successfully
Installing 'ansible.utils:2.4.2' to
'/root/.ansible/collections/ansible_collections/ansible/utils'
ansible.utils:2.4.2 was installed successfully
root@ubuntu:/home/mkrausch/ansible-avd-cloudvision-demo#
```

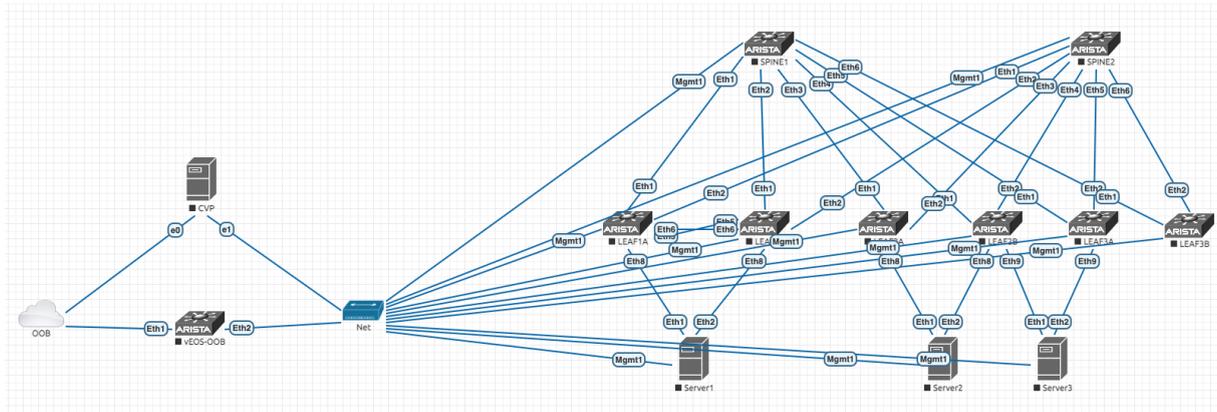
Installing newest Development versions

In order to use the newest AVD features it may be necessary to use specific branches of the code. In this example the newest development versions are copied to the Ubuntu Server to use e.g. BGP EVPN Multihoming and IPv6 unnumbered features.

```
root@ubuntu:/home/mkrausch/ansible-avd-cloudvision-demo# rm -R ansible-avd
root@ubuntu:/home/mkrausch/ansible-avd-cloudvision-demo# git clone --depth 1
--branch devel https://github.com/aristanetworks/ansible-avd.git
Cloning into 'ansible-avd'...
remote: Enumerating objects: 1951, done.
remote: Counting objects: 100% (1951/1951), done.
remote: Compressing objects: 100% (1482/1482), done.
remote: Total 1951 (delta 918), reused 1009 (delta 392), pack-reused 0
Receiving objects: 100% (1951/1951), 3.58 MiB | 3.10 MiB/s, done.
Resolving deltas: 100% (918/918), done.
root@ubuntu:/home/mkrausch/ansible-avd-cloudvision-demo# rm -R ansible-cvp
root@ubuntu:/home/mkrausch/ansible-avd-cloudvision-demo# git clone --depth 1
--branch devel https://github.com/aristanetworks/ansible-cvp.git
Cloning into 'ansible-cvp'...
remote: Enumerating objects: 254, done.
remote: Counting objects: 100% (254/254), done.
remote: Compressing objects: 100% (215/215), done.
remote: Total 254 (delta 36), reused 136 (delta 9), pack-reused 0
Receiving objects: 100% (254/254), 2.23 MiB | 2.06 MiB/s, done.
Resolving deltas: 100% (36/36), done.
root@ubuntu:/home/mkrausch/ansible-avd-cloudvision-demo# ls
ansible-avd  ansible-cvp  data      INSTALLATION.md  LICENSE  playbooks
requirements.txt
ansible.cfg  configlets  DEMO.md  inventory        Makefile  README.md
```

Example Lab Setup

The Lab setups have been done on a virtual basis on **EVE-NG** using **vEOS 4.26.1F** and **CVP 2021.2.0**. **CVP** has been assigned the IP address **192.168.1.195** and can be reached via the OOB Management Interfaces.



Leaf pair 1 is set up for MLAG and Leafs from pairs 2 & 3 are set up for BGP EVPN with VXLAN using IPv6 unnumbered in the underlay.

The used credentials for CloudVision are:

Username: **cvpadmin**

Password: **pAssw0rd**

Configuring the Inventory file

The inventory file basically describes the existing nodes within a Fabric and a CVP Server. With the help of “#” you can **uncomment configuration parameters**.

```
root@ubuntu:/home/mkrausch/ansible-avd-cloudvision-demo/inventory# nano
inventory.yml
```

```
*****

---
all:
  children:
    CVP:
      hosts:
        cv_ztp:
          ansible_host: 192.168.1.195
          ansible_user: root
          ansible_password: pAssw0rd
        cv_server:
          ansible_httpapi_host: 192.168.1.195
          ansible_host: 192.168.1.195
          ansible_user: cvpadmin
          ansible_password: pAssw0rd
          ansible_connection: httpapi
          ansible_httpapi_use_ssl: true
          ansible_httpapi_validate_certs: false
          ansible_network_os: eos
          ansible_httpapi_port: 443
          # Configuration to get Virtual Env information
          ansible_python_interpreter: $(which python3)
# DC1_Fabric - EVPN Fabric running in home lab
DC1:
  children:
    DC1_FABRIC:
      children:
        DC1_SPINES:
          vars:
            type: spine
          hosts:
            DC1-SPINE1:
              ansible_host: 192.168.20.101
            DC1-SPINE2:
              ansible_host: 192.168.20.102
        DC1_L3LEAFS:
          vars:
            type: l3leaf
          children:
            DC1_LEAF1:
              hosts:
```

```

        DC1-LEAF1A:
            ansible_host: 192.168.20.111
        DC1-LEAF1B:
            ansible_host: 192.168.20.112
    DC1_LEAF2:
        hosts:
            DC1-LEAF2A:
                ansible_host: 192.168.20.113
            DC1-LEAF2B:
                ansible_host: 192.168.20.114
    DC1_LEAF3:
        hosts:
            DC1-LEAF3A:
                ansible_host: 192.168.20.115
            DC1-LEAF3B:
                ansible_host: 192.168.20.116
#    DC1_L2LEAFS:
#        vars:
#            type: l2leaf
#        children:
#            DC1_L2LEAF1:
#                hosts:
#                    DC1-L2LEAF1A:
#                        ansible_host: 10.255.0.17
#            DC1_L2LEAF2:
#                hosts:
#                    DC1-L2LEAF2A:
#                        ansible_host: 10.255.0.18
vars:
    ansible_connection: httpapi
    ansible_network_os: eos
    ansible_user: ansible
    # should use vault for passwords
    ansible_ssh_pass: ansible
    ansible_become: true
    ansible_become_method: enable
    ansible_httpapi_use_ssl: true
    ansible_httpapi_validate_certs: false
DC1_TENANTS_NETWORKS:
    children:
        DC1_L3LEAFS:
#            DC1_L2LEAFS:
DC1_SERVERS:
    children:
        DC1_L3LEAFS:
#            DC1_L2LEAFS:

```

Adjusting the dc1-upload-configlets.yml file

Within the **dc1_upload_configlets.yml** file you will have to adjust three **vars parameters**:

```
root@ubuntu:/home/mkrausch/ansible-avd-cloudvision-demo/playbooks# nano
dc1-upload-configlets.yml
```

```
*****
```

```
---
```

```
- name: Configlet upload management
  hosts: cv_server
  connection: local
  gather_facts: false
  collections:
    - arista.avd
    - arista.cvp
  tasks:
    - name: 'upload configlets to cloudvision {{inventory_hostname}}'
      import_role:
        name: arista.avd.cvp_configlet_upload
      vars:
        configlet_directory: 'inventory/intended/configs'
        configlets_cvp_prefix: 'DC1-AVD'
        file_extension: 'cfg'
```

```
*****
```

Configuring the CVP.yml file

With these settings Ansible can automatically create a **dhcpd.conf** file so that when using **ZTP+** the Spine & Leaf nodes will get automatically preassigned IP addresses.

```
root@ubuntu:/home/mkrausch/ansible-avd-cloudvision-demo# cd inventory/
root@ubuntu:/home/mkrausch/ansible-avd-cloudvision-demo/inventory# cd group_vars/
root@ubuntu:/home/mkrausch/ansible-avd-cloudvision-demo/inventory/group_vars# ls
CVP.yml DC1_FABRIC.yml DC1_SERVERS.yml DC1_TENANTS_NETWORKS.yml DC1.yml
root@ubuntu:/home/mkrausch/ansible-avd-cloudvision-demo/inventory/group_vars# nano
CVP.yml
```

```
*****
```

```
---
```

```
ztp:
  default:
    registration: 'http://192.168.20.195/ztp/bootstrap'
    gateway: 192.168.20.199
    nameservers:
      - '192.168.1.1'
  general:
    subnets:
      - network: 192.168.20.0
        netmask: 255.255.255.0
        registration: 'http://192.168.20.195/ztp/bootstrap'
        gateway: 192.168.20.199
        nameservers:
          - '192.168.1.1'
        start: 192.168.20.200
        end: 192.168.20.230
        lease_time: 300
    clients:
      # GNS3 VM Request
      #   - name: gns3
      #     mac: '00:50:56:86:8b:2f'
      #     ip4: 10.255.0.2
      # AVD/CVP Integration
      - name: DC1-SPINE1
        mac: '50:00:00:04:00:00'
        ip4: 192.168.20.101
      - name: DC1-SPINE2
        mac: '50:00:00:09:00:00'
        ip4: 192.168.20.102
      - name: DC1-LEAF1A
        mac: '50:00:00:05:00:00'
        ip4: 192.168.20.111
      - name: DC1-LEAF1B
        mac: '50:00:00:06:00:00'
        ip4: 192.168.20.112
      - name: DC1-LEAF2A
```

```
mac: '50:00:00:07:00:00'  
ip4: 192.168.20.113  
- name: DC1-LEAF2B  
mac: '50:00:00:08:00:00'  
ip4: 192.168.20.114  
- name: DC1-LEAF3A  
mac: '50:00:00:0a:00:00'  
ip4: 192.168.20.115  
- name: DC1-LEAF3B  
mac: '50:00:00:0b:00:00'  
ip4: 192.168.20.116
```

Configuring the DC1.yml file

Within the DC1.yml File you will have to config basic **Management Access parameters** settings.

```
root@ubuntu:/home/mkrausch/ansible-avd-cloudvision-demo/inventory/group_vars# nano DC1.yml
```

```
*****
```

```
---
```

```
# Validation lab
# local users
local_users:
  admin:
    privilege: 15
    role: network-admin
    sha512_password:
"$6$.S.ri8u107kH1xHA$RUC5K9YQ0sJGhWWM1XxjEPymjMfxDJl2qnjn.xBYyAfkjm9MxGZSh6ISG52Gs
4doHZhorIGKZkcPDkOUHSHat/"

  cvpadmin:
    privilege: 15
    role: network-admin
    sha512_password:
"$6$Jwp8ndoDuNoW60f3$5J/zb7pTg05Cbxxpy9oLIpz6k4AMRb0YeGqXV24y0hYH8aa3e1FinLZ0.ltnW
/e22AWjW4mmlti7Tjxk6IELB/"

  ansible:
    privilege: 15
    role: network-admin
    sha512_password:
"$6$Jwp8ndoDuNoW60f3$5J/zb7pTg05Cbxxpy9oLIpz6k4AMRb0YeGqXV24y0hYH8aa3e1FinLZ0.ltnW
/e22AWjW4mmlti7Tjxk6IELB/"

# Cloud Vision server information
cvp_instance_ip: 192.168.20.195
cvp_ingestauth_key: magickey
# cvp_ingestauth_key: telarista

# OOB Management network default gateway.
mgmt_interface: Management1
mgmt_interface_vrf: MGMT
mgmt_gateway: 192.168.20.199

# dns servers.
name_servers:
  - 192.168.1.1
  - 8.8.8.8
```

```
# NTP Servers IP or DNS name, first NTP server will be preferred, and sourced from
Management VRF
custom_structured_configuration_ntp:
  local_interface:
    name: Management1
    vrf: MGMT
  servers:
    - name: 192.168.20.199
      preferred: true
      vrf: MGMT
# - name: 1.fr.pool.ntp.org
#   vrf:
```

```
*****
```

Configuring the DC1_FABRIC.yml file

Within the DC1_FABRIC.yml File you will have to configure the **individual Spine & Leaf details** within the Fabric.

Please make sure to comment “update wait-for-convergence” and “update wait-install” when using vEOS!

```
root@ubuntu:/home/mkrausch/ansible-avd-cloudvision-demo/inventory/group_vars# nano DC1_FABRIC.yml
```

```
*****
```

```
---
```

```
# L3LS Fabric Values - update these values with caution,  
# SOME CHANGES COULD BE DISRUPTIVE.
```

```
fabric_name: DC1_FABRIC
```

```
# Point to Point Underlay with RFC 5549(eBGP), i.e. IPv6 Unnumbered.  
# Requires "underlay_routing_protocol: EBGP"  
underlay_rfc5549: true
```

```
# Enable vlan aware bundles  
vxlan_vlan_aware_bundles: true
```

```
# bgp peer groups passwords  
bgp_peer_groups:  
  IPv4_UNDERLAY_PEERS:  
    password: "AQQvKeimxJu+uGQ/yYvv9w=="  
  EVPN_OVERLAY_PEERS:  
    password: "q+VNViP5i4rVjW1cxFv2wA=="  
  MLAG_IPv4_UNDERLAY_PEER:  
    password: "vnEaG8gMeQf3d3cN6PktXQ=="
```

```
# Spine Switches
```

```
spine:  
  defaults:  
    platform: vEOS-LAB  
    bgp_as: 66001  
    loopback_ipv4_pool: 1.1.1.0/24  
    bgp_defaults:  
      - 'no bgp default ipv4-unicast'  
      - 'distance bgp 20 200 200'  
      - 'graceful-restart restart-time 300'  
      - 'graceful-restart'  
#  mlag_peer_ipv4_pool: 10.255.252.0/24  
#  mlag_peer_l3_ipv4_pool: 10.255.251.0/24  
nodes:  
  DC1-SPINE1:  
    id: 1
```

```

    mgmt_ip: 192.168.20.101/24
DC1-SPINE2:
    id: 2
    mgmt_ip: 192.168.20.102/24

# Leaf switch groups
# A maximum of two nodes can form a leaf group
# When two nodes are in a leaf group this will automatically form mlag pair

l3leaf:
  defaults:
    platform: vEOS-LAB
    loopback_ipv4_pool: 1.1.1.0/24
    loopback_ipv4_offset: 2
    vtep_loopback_ipv4_pool: 2.2.2.0/24
    uplink_interfaces: ['Ethernet1', 'Ethernet2']
    uplink_switches: ['DC1-SPINE1', 'DC1-SPINE2']
#   uplink_ipv4_pool: 172.31.255.0/24
    mlag_interfaces: [ Ethernet5, Ethernet6 ]
    mlag_peer_ipv4_pool: 10.255.252.0/24
    mlag_peer_l3_ipv4_pool: 10.255.251.0/24
    virtual_router_mac_address: 00:1c:73:00:dc:01
    bgp_defaults:
      - 'no bgp default ipv4-unicast'
      - 'distance bgp 20 200 200'
      - 'graceful-restart restart-time 300'
      - 'graceful-restart'
    spanning_tree_mode: mstp
    spanning_tree_priority: 16384
  node_groups:
# MLAG Leaf peers
    DC1_LEAF1:
      bgp_as: 65101
      nodes:
        DC1-LEAF1A:
          id: 1
          mgmt_ip: 192.168.20.111/24
          uplink_switch_interfaces: [Ethernet1, Ethernet1]
        DC1-LEAF1B:
          id: 2
          mgmt_ip: 192.168.20.112/24
          uplink_switch_interfaces: [Ethernet2, Ethernet2]
# No MLAG Leaf peers
    DC1_LEAF3:
      bgp_as: 65103
      nodes:
        DC1-LEAF2A:
          id: 3
          mgmt_ip: 192.168.20.113/24
          uplink_switch_interfaces: [Ethernet3, Ethernet3]
    DC1_LEAF4:

```

```

    bgp_as: 65104
    nodes:
      DC1-LEAF2B:
        id: 4
        mgmt_ip: 192.168.20.114/24
        uplink_switch_interfaces: [Ethernet4, Ethernet4]
DC1_LEAF5:
  bgp_as: 65105
  nodes:
    DC1-LEAF3A:
      id: 5
      mgmt_ip: 192.168.20.115/24
      uplink_switch_interfaces: [Ethernet5, Ethernet5]
DC1_LEAF6:
  bgp_as: 65106
  nodes:
    DC1-LEAF3B:
      id: 6
      mgmt_ip: 192.168.20.116/24
      uplink_switch_interfaces: [Ethernet6, Ethernet6]
#l2leaf:
# defaults:
#   platform: vEOS-LAB
#   uplink_switches: ['DC1-LEAF1A', 'DC1-LEAF1B']
#   uplink_interfaces: [ Ethernet1, Ethernet2 ]
#   mlag_interfaces: [ Ethernet3, Ethernet4 ]
#   mlag_peer_ipv4_pool: 10.255.252.0/24
#   mlag_peer_l3_ipv4_pool: 10.255.251.0/24
#   virtual_router_mac_address: 00:1c:73:00:dc:01
#   spanning_tree_mode: mstp
#   spanning_tree_priority: 16384
#   mlag: false
# node_groups:
#   DC1_L2LEAF1:
#     uplink_switches: [ DC1-LEAF1A, DC1-LEAF1B ]
#     filter:
#       tenants: [ Tenant_A ]
#       tags: [ opzone, web, app ]
#     nodes:
#       DC1-L2LEAF1A:
#         id: 5
#         mgmt_ip: 10.255.0.17/24
#         uplink_switch_interfaces: [ Ethernet5, Ethernet5 ]
#   DC1_L2LEAF2:
#     uplink_switches: [ DC1-LEAF2A, DC1-LEAF2B ]
#     filter:
#       tenants: [ Tenant_A ]
#       tags: [ opzone, web, app ]
#     nodes:
#       DC1-L2LEAF2A:
#         id: 7

```

```
#          mgmt_ip: 10.255.0.18/24
#          uplink_switch_interfaces: [ Ethernet5, Ethernet5 ]

#### Override for vEOS Lab Caveats ####

# Disable update wait-for-convergence and update wait-for-install,
# which is not supported in vEOS-LAB.
# Refer to design guide

# Update p2p mtu 9000 -> 1500
p2p_uplinks_mtu: 1500

# Adjust default bfd values
bfd_multihop:
  interval: 1200
  min_rx: 1200
  multiplier: 3

# List of additional CVP configlets to bind to devices and containers
# Configlets MUST be configured on CVP before running AVD playbooks.

# cv_configlets:
#   containers:
#     DC1_L3LEAFS:
#       - ASE_GLOBAL-ALIASES

*****
```

Configuring the DC1_SERVERS.yml file

Within the DC1_SERVERS.yml file you configure the **attached Servers links and/or Port Channels** so that Ansible will create the correct Port Channels and Uplink Ports.

```
root@ubuntu:/home/mkrausch/ansible-avd-cloudvision-demo/inventory/group_vars# nano DC1_SERVERS.yml
```

```
*****
```

```
---
```

```
port_profiles:
  TENANT_A:
    mode: trunk
    vlans: "10,20,30,40"
  TENANT_B:
    mode: trunk
    vlans: "10,20,40,50"

servers:
  server01:
    rack: RackA
    adapters:
      - type: nic
        server_ports: [Eth1,Eth2]
        switch_ports: [Ethernet8,Ethernet8]
        switches: [DC1-LEAF1A,DC1-LEAF1B]
        profile: TENANT_A
        port_channel:
          state: present
          description: PortChannel8
          mode: active
  server02:
    rack: RackB
    adapters:
      - type: nic
        server_ports: [Eth1,Eth2]
        switch_ports: [Ethernet8,Ethernet8]
        switches: [DC1-LEAF2A,DC1-LEAF2B]
        profile: TENANT_B
        port_channel:
          state: present
          description: PortChannel8
          mode: active
          short_esl: 0003:0004:0008
  server03:
    rack: RackC
    adapters:
      - type: nic
```

```
server_ports: [Eth1,Eth2]
switch_ports: [Ethernet9,Ethernet9]
switches: [DC1-LEAF2B,DC1-LEAF3A]
profile: TENANT_B
port_channel:
  state: present
  description: PortChannel9
  mode: active
  short_esi: 0004:0005:0009
```

```
*****
```

Configuring the DC1_TENANTS_NETWORKS.yml file

Here you configure the **L2 VLANs and VRFs** which should be used within the Fabric for each Tenant.

```
root@ubuntu:/home/mkrausch/ansible-avd-cloudvision-demo/inventory/group_vars# nano DC1_TENANTS_NETWORKS.yml
```

```
*****
```

```
---
```

```
# DC1 Tenants Networks
# Documentation of Tenant specific information - Vlans/VRFs
tenants:
  # Tenant A Specific Information - VRFs / VLANs
  Tenant_A:
    mac_vrf_vni_base: 10000
    l2vlans:
      10:
        vni_override: 10010
        name: VLAN10
        tags: [VLAN10]
      20:
        name: VLAN20
        tags: [VLAN20]

  # Tenant B Specific Information - VRFs / VLANs
  Tenant_B:
    mac_vrf_vni_base: 20000
    vrfs:
      Tenant_B_OP_Zone:
        vrf_vni: 20
        vtep_diagnostic:
          loopback: 100
          loopback_ip_range: 10.255.1.0/24
        svis:
          30:
            name: Tenant_B_OP_Zone_1
            tags: [opzone]
            enabled: true
            ip_subnet: 192.168.30.0/24
            ip_address_virtual: 192.168.30.250/24
          40:
            name: Tenant_B_OP_Zone_2
            tags: [opzone]
            enabled: true
            ip_address_virtual: 192.168.40.250/24
          50:
            name: Tenant_B_OP_Zone_3
            tags: [wan]
```

```
        enabled: true
        ip_address_virtual: 192.168.50.250/24
# Tenant C Specific Information - VRFs / VLANs
# Tenant_C:
#   mac_vrf_vni_base: 30000
#   vrfs:
#     Tenant_C_OP_Zone:
#       vrf_vni: 30
#       svis:
#         310:
#           name: Tenant_C_OP_Zone_1
#           tags: [opzone]
#           enabled: true
#           ip_subnet: 10.3.10.0/24
#         311:
#           name: Tenant_C_OP_Zone_2
#           tags: [opzone]
#           enabled: true
#           ip_subnet: 10.3.11.1/24
#     Tenant_C_WAN_Zone:
#       vrf_vni: 31
#       svis:
#         350:
#           name: Tenant_C_WAN_Zone_1
#           tags: [wan]
#           enabled: true
#           ip_subnet: 10.3.50.1/24
```

Creating DHCP ZTP Configuration Files on CloudVision

You can create ready to go DHCP Config Files on a CloudVision Server based on the client information configured within the CVP-yml File.

Important hint:

When trying to execute a playbook you may get the error message:

ERROR! Invalid callback for stdout specified: yaml

To solve this issue you might have to uncomment the YAML callback plugin option within the ansible.cfg file:

```
root@ubuntu:/home/mkrausch/ansible-avd-cloudvision-demo# cat ansible.cfg
*****
...
action_plugins = /usr/lib/python2.7/site-packages/napalm_ansible/plugins/action
jinja2_extensions = jinja2.ext.loopcontrols,jinja2.ext.do,jinja2.ext.i18n
# enable the YAML callback plugin.
# stdout_callback = yaml
# enable the stdout_callback when running ad-hoc commands.
bin_ansible_callbacks = True
command_warnings=False
...
*****
```

A second option is to install the Ansible community.general collection to use stdout callback for YAML files:

```
root@ubuntu:/home/mkrausch/ansible-avd-cloudvision-demo# ansible-galaxy collection install community.general
[DEPRECATION WARNING]: COMMAND_WARNINGS option, the command warnings feature is being removed. This feature will be removed from ansible-core in version 2.14. Deprecation warnings can be disabled by setting deprecation_warnings=False in ansible.cfg.
Starting galaxy collection install process
Process install dependency map
Starting collection install process
Downloading https://galaxy.ansible.com/download/community-general-4.0.1.tar.gz to /root/.ansible/tmp/ansible-local-38496x35y1v4/tmpplpr19vf/community-general-4.0.1-5z1z_y2a
Installing 'community.general:4.0.1' to '/root/.ansible/collections/ansible_collections/community/general'
```

```
community.general:4.0.1 was installed successfully
root@ubuntu:/home/mkrausch/ansible-avd-cloudvision-demo#
root@ubuntu:/home/mkrausch/ansible-avd-cloudvision-demo#
```

Before starting please make sure that the parameter **gather_facts: true** is set correctly within the **dc1-ztp-configuration.yml** file to prevent error messages:

```
root@ubuntu:/home/mkrausch/ansible-avd-cloudvision-demo/playbooks# cat
dc1-ztp-configuration.yml
```

```
*****
---
- name: Configure ZTP/DHCP service on CloudVision
  hosts: cv_ztp
  gather_facts: true
  tasks:
  - name: 'Execute ZTP configuration role'
    import_role:
      name: arista.cvp.dhcp_configuration
*****
```

Please make sure that a DHCP Server is installed correctly on CVP which should be installed by default:

```
[root@cvp ~]# rpm -qa | grep dhcp
dhcp-common-4.2.5-82.el7.centos.x86_64
dhcp-libs-4.2.5-82.el7.centos.x86_64
dhcp-4.2.5-82.el7.centos.x86_64
```

Now you can create a dhcp config file for ZTP onboarding of new devices on CloudVision remotely from the Ubuntu Ansible system:

```
root@ubuntu:/home/mkrausch/ansible-avd-cloudvision-demo# ansible-playbook
playbooks/dc1-ztp-configuration.yml
[DEPRECATION WARNING]: COMMAND_WARNINGS option, the command warnings feature is
being removed. This feature will be
removed from ansible-core in version 2.14. Deprecation warnings can be disabled by
setting deprecation_warnings=False in
ansible.cfg.
[WARNING]: Collection arista.cvp does not support Ansible version 2.11.3
```

```
PLAY [Configure ZTP/DHCP service on CloudVision]
*****
```

TASK [Gathering Facts]

[WARNING]: Platform linux on host cv_ztp is using the discovered Python interpreter at /usr/bin/python, but future installation of another Python interpreter could change the meaning of that path. See https://docs.ansible.com/ansible/2.11/reference_appendices/interpreter_discovery.html for more information.
ok: [cv_ztp]

TASK [arista.cvp.dhcp_configuration : Start creation/update process.]

included:
/root/.ansible/collections/ansible_collections/arista/cvp/roles/dhcp_configuration/tasks/./online.yml for cv_ztp

TASK [arista.cvp.dhcp_configuration : gather os specific variables for Centos / Red Hat OS] *****

ok: [cv_ztp]

TASK [arista.cvp.dhcp_configuration : gather os specific variables for Debian / Ubuntu OS] *****

skipping: [cv_ztp]

TASK [arista.cvp.dhcp_configuration : update packages list]

skipping: [cv_ztp]

TASK [arista.cvp.dhcp_configuration : install packages]

ok: [cv_ztp]

TASK [arista.cvp.dhcp_configuration : include_tasks]

skipping: [cv_ztp]

TASK [arista.cvp.dhcp_configuration : generate DHCPd configuration file]

ok: [cv_ztp]

TASK [arista.cvp.dhcp_configuration : check & activate DHCP service]

ok: [cv_ztp]

PLAY RECAP

cv_ztp : ok=6 changed=0 unreachable=0 failed=0
skipped=3 rescued=0 ignored=0

On the CVP Server the DHCP process has been restarted and a DHCP config file has been generated by Ansible:

```
[root@cvp ~]# systemctl status dhcpd
● dhcpd.service - DHCPv4 Server Daemon
   Loaded: loaded (/usr/lib/systemd/system/dhcpd.service; enabled; vendor preset: disabled)
   Active: active (running) since Thu 2021-11-11 10:15:42 CET; 2h 42min ago
     Docs: man:dhcpd(8)
           man:dhcpd.conf(5)
  Main PID: 1502 (dhcpd)
    Status: "Dispatching packets..."
     Tasks: 1
    Memory: 7.3M
    CGroup: /system.slice/dhcpd.service
            └─1502 /usr/sbin/dhcpd -f -cf /etc/dhcp/dhcpd.conf -user dhcpd -group dhcpd
--no-pid
```

```
Nov 11 10:15:42 cvp.lab.local systemd[1]: Started DHCPv4 Server Daemon.
Nov 11 10:15:42 cvp.lab.local dhcpd[1502]: Sending on
LPF/eth1/00:50:00:00:1a:01/192.168.20.0/24
Nov 11 10:15:42 cvp.lab.local dhcpd[1502]:
Nov 11 10:15:42 cvp.lab.local dhcpd[1502]: No subnet declaration for eth0 (192.168.1.195).
Nov 11 10:15:42 cvp.lab.local dhcpd[1502]: ** Ignoring requests on eth0.  If this is not
what
Nov 11 10:15:42 cvp.lab.local dhcpd[1502]:   you want, please write a subnet declaration
Nov 11 10:15:42 cvp.lab.local dhcpd[1502]:   in your dhcpd.conf file for the network
segment
Nov 11 10:15:42 cvp.lab.local dhcpd[1502]:   to which interface eth0 is attached. **
Nov 11 10:15:42 cvp.lab.local dhcpd[1502]:
Nov 11 10:15:42 cvp.lab.local dhcpd[1502]: Sending on   Socket/fallback/fallback-net
```

```
[root@cvp ~]# cat /etc/dhcp/dhcpd.conf

# Ansible managed - cv_ztp
# Subnet of ZTP interface
subnet 192.168.20.0 netmask 255.255.255.0 {
    range 192.168.20.200 192.168.20.230;
    option routers 192.168.20.199;
    option domain-name-servers 192.168.1.1;
    option bootfile-name "http://192.168.20.195/ztp/bootstrap";
    max-lease-time 300;
}

# Per host definition
host DC1-SPINE1 {
    option host-name "DC1-SPINE1";
    hardware ethernet 50:00:00:04:00:00;
    fixed-address 192.168.20.101;
    option bootfile-name "http://192.168.20.195/ztp/bootstrap";
    option routers 192.168.20.199;
    option domain-name-servers 192.168.1.1;
}
```

```

host DC1-SPINE2 {
    option host-name "DC1-SPINE2";
    hardware ethernet 50:00:00:09:00:00;
    fixed-address 192.168.20.102;
    option bootfile-name "http://192.168.20.195/ztp/bootstrap";
    option routers 192.168.20.199;
    option domain-name-servers 192.168.1.1;
}

host DC1-LEAF1A {
    option host-name "DC1-LEAF1A";
    hardware ethernet 50:00:00:05:00:00;
    fixed-address 192.168.20.111;
    option bootfile-name "http://192.168.20.195/ztp/bootstrap";
    option routers 192.168.20.199;
    option domain-name-servers 192.168.1.1;
}

host DC1-LEAF1B {
    option host-name "DC1-LEAF1B";
    hardware ethernet 50:00:00:06:00:00;
    fixed-address 192.168.20.112;
    option bootfile-name "http://192.168.20.195/ztp/bootstrap";
    option routers 192.168.20.199;
    option domain-name-servers 192.168.1.1;
}

host DC1-LEAF2A {
    option host-name "DC1-LEAF2A";
    hardware ethernet 50:00:00:07:00:00;
    fixed-address 192.168.20.113;
    option bootfile-name "http://192.168.20.195/ztp/bootstrap";
    option routers 192.168.20.199;
    option domain-name-servers 192.168.1.1;
}

host DC1-LEAF2B {
    option host-name "DC1-LEAF2B";
    hardware ethernet 50:00:00:08:00:00;
    fixed-address 192.168.20.114;
    option bootfile-name "http://192.168.20.195/ztp/bootstrap";
    option routers 192.168.20.199;
    option domain-name-servers 192.168.1.1;
}

host DC1-LEAF3A {
    option host-name "DC1-LEAF3A";
    hardware ethernet 50:00:00:0a:00:00;
    fixed-address 192.168.20.115;
    option bootfile-name "http://192.168.20.195/ztp/bootstrap";
    option routers 192.168.20.199;
    option domain-name-servers 192.168.1.1;
}

```

```
host DC1-LEAF3B {  
    option host-name "DC1-LEAF3B";  
    hardware ethernet 50:00:00:0b:00:00;  
    fixed-address 192.168.20.116;  
    option bootfile-name "http://192.168.20.195/ztp/bootstrap";  
    option routers 192.168.20.199;  
    option domain-name-servers 192.168.1.1;  
}
```

Creating offline Config Files

You can create configs without sending them to CVP **offline** by using the “**make build**” or “**ansible-playbook dc1-fabric-deploy-cvp.yml --tags build**” command.

```
root@ubuntu:/home/mkrausch/ansible-avd-cloudvision-demo# ansible-playbook
playbooks/dc1-fabric-deploy-cvp.yml --tags build
[DEPRECATION WARNING]: COMMAND_WARNINGS option, the command warnings feature is
being removed. This feature will be
removed from ansible-core in version 2.14. Deprecation warnings can be disabled by
setting deprecation_warnings=False in
ansible.cfg.
```

```
PLAY [Manage Arista EOS EVPN/VXLAN Configuration]
*****
```

```
[WARNING]: Collection arista.cvp does not support Ansible version 2.11.3
```

```
TASK [arista.avd.build_output_folders : Cleanup existing folders in
/home/mkrausch/ansible-avd-cloudvision-demo/inventory/intended] ***
changed: [DC1-SPINE1 -> localhost]
```

```
TASK [arista.avd.build_output_folders : Create folder
/home/mkrausch/ansible-avd-cloudvision-demo/inventory/intended] *****
changed: [DC1-SPINE1 -> localhost]
```

```
TASK [arista.avd.build_output_folders : Create folder structured_configs for
structured YAML files] *****
changed: [DC1-SPINE1 -> localhost]
```

```
TASK [arista.avd.build_output_folders : Create folder configs for EOS
Configuration files] *****
changed: [DC1-SPINE1 -> localhost]
```

```
TASK [arista.avd.build_output_folders : Create folder cvp for CVP structured YAML
files] *****
changed: [DC1-SPINE1 -> localhost]
```

```
TASK [arista.avd.build_output_folders : Create documentation folder:
documentation] *****
changed: [DC1-SPINE1 -> localhost]
```

```
TASK [arista.avd.build_output_folders : Create folder fabric for Fabric
documentation] *****
changed: [DC1-SPINE1 -> localhost]
```

```
TASK [arista.avd.build_output_folders : Create folder devices for EOS
documentation] *****
changed: [DC1-SPINE1 -> localhost]
```

```
TASK [arista.avd.build_output_folders : Create folder
/home/mkrausch/ansible-avd-cloudvision-demo/inventory/config_backup for EOS post
backup dir] ***
changed: [DC1-SPINE1 -> localhost]
```

```
TASK [arista.avd.build_output_folders : Create folder
/home/mkrausch/ansible-avd-cloudvision-demo/inventory/config_backup for EOS pre
backup dir] ***
ok: [DC1-SPINE1 -> localhost]
```

```
TASK [arista.avd.build_output_folders : Create folder reports for EOS state report
dir] *****
changed: [DC1-SPINE1 -> localhost]
```

```
TASK [arista.avd.build_output_folders : Create output directory for each EOS
device] *****
changed: [DC1-SPINE1 -> localhost]
```

```
TASK [arista.avd.eos_designs : Create required output directories if not present]
*****
ok: [DC1-SPINE1 -> localhost] =>
(item=/home/mkrausch/ansible-avd-cloudvision-demo/inventory/intended/structured_co
nfigs)
ok: [DC1-SPINE1 -> localhost] =>
(item=/home/mkrausch/ansible-avd-cloudvision-demo/inventory/documentation/fabric)
```

```
TASK [arista.avd.eos_designs : Set AVD facts]
*****
ok: [DC1-SPINE1 -> localhost]
ok: [DC1-LEAF1A -> localhost]
ok: [DC1-SPINE2 -> localhost]
ok: [DC1-LEAF1B -> localhost]
ok: [DC1-LEAF2A -> localhost]
ok: [DC1-LEAF3A -> localhost]
ok: [DC1-LEAF2B -> localhost]
ok: [DC1-LEAF3B -> localhost]
```

```
TASK [arista.avd.eos_designs : Set AVD topology facts]
*****
ok: [DC1-SPINE1 -> localhost]
ok: [DC1-SPINE2 -> localhost]
ok: [DC1-LEAF1A -> localhost]
ok: [DC1-LEAF1B -> localhost]
ok: [DC1-LEAF2B -> localhost]
ok: [DC1-LEAF2A -> localhost]
ok: [DC1-LEAF3A -> localhost]
ok: [DC1-LEAF3B -> localhost]
```

```
TASK [arista.avd.eos_designs : Generate YAML file with hostvars (only for
debugging)] *****
ok: [DC1-SPINE1 -> localhost]
```

```
ok: [DC1-SPINE2 -> localhost]
ok: [DC1-LEAF1A -> localhost]
ok: [DC1-LEAF1B -> localhost]
ok: [DC1-LEAF2A -> localhost]
ok: [DC1-LEAF2B -> localhost]
ok: [DC1-LEAF3A -> localhost]
ok: [DC1-LEAF3B -> localhost]
```

```
TASK [arista.avd.eos_designs : Generate device configuration in structured format]
*****
```

```
ok: [DC1-SPINE2 -> localhost]
ok: [DC1-SPINE1 -> localhost]
ok: [DC1-LEAF1A -> localhost]
ok: [DC1-LEAF1B -> localhost]
ok: [DC1-LEAF2A -> localhost]
ok: [DC1-LEAF2B -> localhost]
ok: [DC1-LEAF3A -> localhost]
ok: [DC1-LEAF3B -> localhost]
```

```
TASK [arista.avd.eos_designs : Write device structured configuration to YAML file]
*****
```

```
ok: [DC1-SPINE2 -> localhost]
ok: [DC1-LEAF1B -> localhost]
ok: [DC1-LEAF1A -> localhost]
ok: [DC1-SPINE1 -> localhost]
ok: [DC1-LEAF2A -> localhost]
ok: [DC1-LEAF3A -> localhost]
ok: [DC1-LEAF2B -> localhost]
ok: [DC1-LEAF3B -> localhost]
```

```
TASK [arista.avd.eos_designs : Include device structured configuration, that was
previously generated.] *****
```

```
ok: [DC1-SPINE1 -> localhost]
ok: [DC1-SPINE2 -> localhost]
ok: [DC1-LEAF1A -> localhost]
ok: [DC1-LEAF1B -> localhost]
ok: [DC1-LEAF2A -> localhost]
ok: [DC1-LEAF2B -> localhost]
ok: [DC1-LEAF3A -> localhost]
ok: [DC1-LEAF3B -> localhost]
```

```
TASK [arista.avd.eos_designs : Store checksum of existing fabric documentation]
*****
```

```
ok: [DC1-SPINE1 -> localhost]
```

```
TASK [arista.avd.eos_designs : Generate fabric documentation in Markdown Format.]
*****
```

```
ok: [DC1-SPINE1 -> localhost]
```

```
TASK [arista.avd.eos_designs : Generate TOC for fabric documentation]
*****
```

changed: [DC1-SPINE1 -> localhost]

TASK [arista.avd.eos_designs : Generate fabric point-to-point links summary in csv format.] *****

changed: [DC1-SPINE1 -> localhost]

TASK [arista.avd.eos_designs : Generate fabric topology in csv format.]

changed: [DC1-SPINE1 -> localhost]

TASK [arista.avd.eos_cli_config_gen : Create required output directories if not present] *****

ok: [DC1-SPINE1 -> localhost] =>

(item=/home/mkrausch/ansible-avd-cloudvision-demo/inventory/intended/structured_configs)

ok: [DC1-SPINE1 -> localhost] =>

(item=/home/mkrausch/ansible-avd-cloudvision-demo/inventory/documentation)

ok: [DC1-SPINE1 -> localhost] =>

(item=/home/mkrausch/ansible-avd-cloudvision-demo/inventory/intended/configs)

ok: [DC1-SPINE1 -> localhost] =>

(item=/home/mkrausch/ansible-avd-cloudvision-demo/inventory/documentation/devices)

TASK [arista.avd.eos_cli_config_gen : Check if structure configuration file exists] *****

skipping: [DC1-SPINE1]

skipping: [DC1-SPINE2]

skipping: [DC1-LEAF1A]

skipping: [DC1-LEAF1B]

skipping: [DC1-LEAF2A]

skipping: [DC1-LEAF2B]

skipping: [DC1-LEAF3A]

skipping: [DC1-LEAF3B]

TASK [arista.avd.eos_cli_config_gen : Include device intended structure configuration variables] *****

skipping: [DC1-SPINE1]

skipping: [DC1-SPINE2]

skipping: [DC1-LEAF1A]

skipping: [DC1-LEAF1B]

skipping: [DC1-LEAF2A]

skipping: [DC1-LEAF2B]

skipping: [DC1-LEAF3A]

skipping: [DC1-LEAF3B]

TASK [arista.avd.eos_cli_config_gen : Generate eos intended configuration]

changed: [DC1-SPINE1 -> localhost]

changed: [DC1-LEAF1B -> localhost]

changed: [DC1-SPINE2 -> localhost]

changed: [DC1-LEAF2A -> localhost]

changed: [DC1-LEAF1A -> localhost]

```
changed: [DC1-LEAF2B -> localhost]
changed: [DC1-LEAF3B -> localhost]
changed: [DC1-LEAF3A -> localhost]
```

```
TASK [arista.avd.eos_cli_config_gen : Store checksum of existing device
documentation] *****
```

```
ok: [DC1-SPINE2 -> localhost]
ok: [DC1-SPINE1 -> localhost]
ok: [DC1-LEAF2A -> localhost]
ok: [DC1-LEAF1B -> localhost]
ok: [DC1-LEAF1A -> localhost]
ok: [DC1-LEAF3A -> localhost]
ok: [DC1-LEAF2B -> localhost]
ok: [DC1-LEAF3B -> localhost]
```

```
TASK [arista.avd.eos_cli_config_gen : Generate content of device documentation]
*****
```

```
ok: [DC1-LEAF1B -> localhost]
ok: [DC1-SPINE2 -> localhost]
ok: [DC1-SPINE1 -> localhost]
ok: [DC1-LEAF2A -> localhost]
ok: [DC1-LEAF1A -> localhost]
ok: [DC1-LEAF2B -> localhost]
ok: [DC1-LEAF3A -> localhost]
ok: [DC1-LEAF3B -> localhost]
```

```
TASK [arista.avd.eos_cli_config_gen : Generate TOC for device documentation]
*****
```

```
changed: [DC1-SPINE1 -> localhost]
changed: [DC1-SPINE2 -> localhost]
changed: [DC1-LEAF2A -> localhost]
changed: [DC1-LEAF1A -> localhost]
changed: [DC1-LEAF1B -> localhost]
changed: [DC1-LEAF2B -> localhost]
changed: [DC1-LEAF3A -> localhost]
changed: [DC1-LEAF3B -> localhost]
```

```
PLAY [Configuration deployment with CVP]
```

```
*****
```

```
TASK [arista.avd.eos_config_deploy_cvp : Create required output directories if not
present] *****
```

```
ok: [cv_server -> localhost] =>
(item=/home/mkrausch/ansible-avd-cloudvision-demo/inventory/intended/structured_co
nfigs/cvp)
```

```
TASK [arista.avd.eos_config_deploy_cvp : Start creation/update process.]
```

```
*****
```

```
included:
```

```
/root/.ansible/collections/ansible_collections/arista/avd/roles/eos_config_deploy_
cvp/tasks/v3/main.yml for cv_server
```

TASK [arista.avd.eos_config_deploy_cvp : Generate intended variables]

ok: [cv_server]

TASK [arista.avd.eos_config_deploy_cvp : Build DEVICES and CONTAINER definition for cv_server] *****

changed: [cv_server -> localhost]

PLAY RECAP

DC1-LEAF1A		: ok=10	changed=2	unreachable=0	failed=0
skipped=2	rescued=0	ignored=0			
DC1-LEAF1B		: ok=10	changed=2	unreachable=0	failed=0
skipped=2	rescued=0	ignored=0			
DC1-LEAF2A		: ok=10	changed=2	unreachable=0	failed=0
skipped=2	rescued=0	ignored=0			
DC1-LEAF2B		: ok=10	changed=2	unreachable=0	failed=0
skipped=2	rescued=0	ignored=0			
DC1-LEAF3A		: ok=10	changed=2	unreachable=0	failed=0
skipped=2	rescued=0	ignored=0			
DC1-LEAF3B		: ok=10	changed=2	unreachable=0	failed=0
skipped=2	rescued=0	ignored=0			
DC1-SPINE1		: ok=29	changed=16	unreachable=0	failed=0
skipped=2	rescued=0	ignored=0			
DC1-SPINE2		: ok=10	changed=2	unreachable=0	failed=0
skipped=2	rescued=0	ignored=0			
cv_server		: ok=4	changed=1	unreachable=0	failed=0
skipped=0	rescued=0	ignored=0			

The generated and ready to use Configs can be found at the folder **inventory/intended/configs**:

```
root@ubuntu:/home/mkrausch/ansible-avd-cloudvision-demo# cd
```

```
inventory/intended/configs/
```

```
root@ubuntu:/home/mkrausch/ansible-avd-cloudvision-demo/inventory/intended/configs
```

```
# ls
```

```
DC1-LEAF1A.cfg DC1-LEAF2A.cfg DC1-LEAF3A.cfg DC1-SPINE1.cfg
```

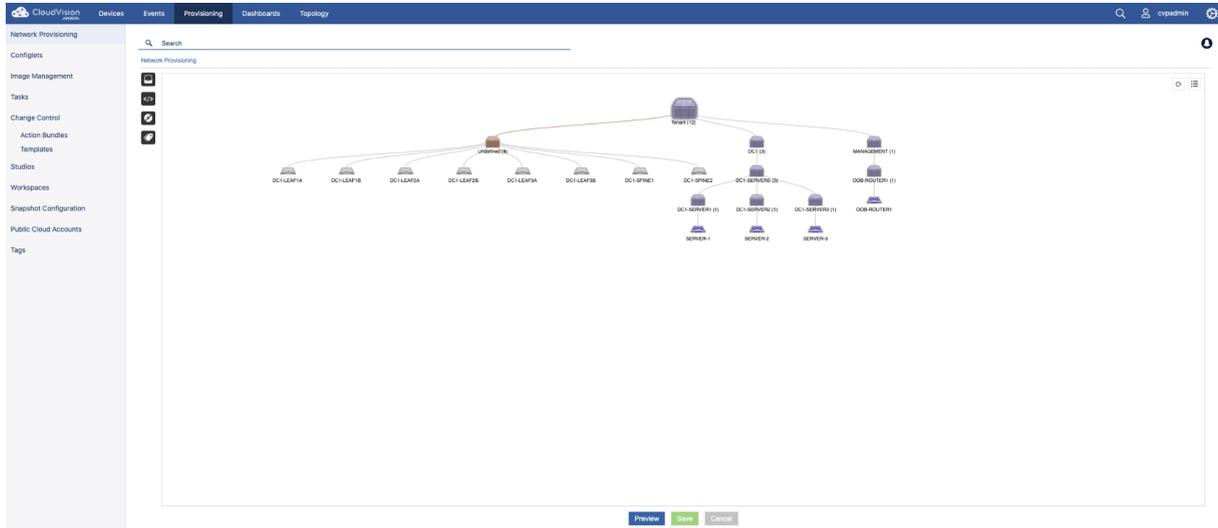
```
DC1-LEAF1B.cfg DC1-LEAF2B.cfg DC1-LEAF3B.cfg DC1-SPINE2.cfg
```

```
root@ubuntu:/home/mkrausch/ansible-avd-cloudvision-demo/inventory/intended/configs
```

```
#
```

Creating Config Files and Provisioning via CVP

In case Arista EOS devices are onboarded via ZTP or manually they will be placed into the Container **“Undefined”**:



Please make sure that the assigned IP addresses and the node names are identical to the ones configured within the AVD **“.yml” files before proceeding!**

You can create the configs and push them to CVP with the use of the command:

```
root@ubuntu:/home/mkrausch/ansible-avd-cloudvision-demo# ansible-playbook
playbooks/dc1-fabric-deploy-cvp.yml --tags provision
[DEPRECATION WARNING]: COMMAND_WARNINGS option, the command warnings feature is
being removed. This feature will be removed from ansible-core in
version 2.14. Deprecation warnings can be disabled by setting
deprecation_warnings=False in ansible.cfg.
```

```
PLAY [Manage Arista EOS EVPN/VXLAN Configuration]
*****
*****
```

```
[WARNING]: Collection arista.cvp does not support Ansible version 2.11.3
```

```
TASK [arista.avd.eos_designs : Create required output directories if not present]
*****
ok: [DC1-SPINE1 -> localhost] =>
(item=/home/mkrausch/ansible-avd-cloudvision-demo/inventory/intended/structured_co
nfigs)
ok: [DC1-SPINE1 -> localhost] =>
(item=/home/mkrausch/ansible-avd-cloudvision-demo/inventory/documentation/fabric)
```

TASK [arista.avd.eos_designs : Set AVD facts]

ok: [DC1-LEAF1A -> localhost]
ok: [DC1-SPINE1 -> localhost]
ok: [DC1-LEAF2A -> localhost]
ok: [DC1-SPINE2 -> localhost]
ok: [DC1-LEAF1B -> localhost]
ok: [DC1-LEAF3A -> localhost]
ok: [DC1-LEAF2B -> localhost]
ok: [DC1-LEAF3B -> localhost]

TASK [arista.avd.eos_designs : Set AVD topology facts]

ok: [DC1-SPINE1 -> localhost]
ok: [DC1-LEAF1A -> localhost]
ok: [DC1-SPINE2 -> localhost]
ok: [DC1-LEAF1B -> localhost]
ok: [DC1-LEAF2A -> localhost]
ok: [DC1-LEAF3A -> localhost]
ok: [DC1-LEAF2B -> localhost]
ok: [DC1-LEAF3B -> localhost]

TASK [arista.avd.eos_designs : Generate device configuration in structured format]

ok: [DC1-SPINE1 -> localhost]
ok: [DC1-SPINE2 -> localhost]
ok: [DC1-LEAF2A -> localhost]
ok: [DC1-LEAF1A -> localhost]
ok: [DC1-LEAF1B -> localhost]
ok: [DC1-LEAF3A -> localhost]
ok: [DC1-LEAF2B -> localhost]
ok: [DC1-LEAF3B -> localhost]

TASK [arista.avd.eos_designs : Write device structured configuration to YAML file]

ok: [DC1-LEAF1B -> localhost]
ok: [DC1-LEAF2A -> localhost]
ok: [DC1-SPINE1 -> localhost]
ok: [DC1-LEAF1A -> localhost]
ok: [DC1-SPINE2 -> localhost]
ok: [DC1-LEAF3A -> localhost]
ok: [DC1-LEAF2B -> localhost]
ok: [DC1-LEAF3B -> localhost]

TASK [arista.avd.eos_designs : Include device structured configuration, that was previously generated.] *****

ok: [DC1-SPINE1 -> localhost]
ok: [DC1-SPINE2 -> localhost]
ok: [DC1-LEAF1A -> localhost]

```
ok: [DC1-LEAF1B -> localhost]
ok: [DC1-LEAF2A -> localhost]
ok: [DC1-LEAF2B -> localhost]
ok: [DC1-LEAF3A -> localhost]
ok: [DC1-LEAF3B -> localhost]
```

```
TASK [arista.avd.eos_designs : Store checksum of existing fabric documentation]
*****
```

```
ok: [DC1-SPINE1 -> localhost]
```

```
TASK [arista.avd.eos_designs : Generate fabric documentation in Markdown Format.]
*****
```

```
ok: [DC1-SPINE1 -> localhost]
```

```
TASK [arista.avd.eos_designs : Generate TOC for fabric documentation]
*****
```

```
ok: [DC1-SPINE1 -> localhost]
```

```
TASK [arista.avd.eos_designs : Generate fabric point-to-point links summary in csv
format.] *****
```

```
ok: [DC1-SPINE1 -> localhost]
```

```
TASK [arista.avd.eos_designs : Generate fabric topology in csv format.]
*****
```

```
ok: [DC1-SPINE1 -> localhost]
```

```
TASK [arista.avd.eos_cli_config_gen : Create required output directories if not
present] *****
```

```
ok: [DC1-SPINE1 -> localhost] =>
```

```
(item=/home/mkrausch/ansible-avd-cloudvision-demo/inventory/intended/structured_co
nfigs)
```

```
ok: [DC1-SPINE1 -> localhost] =>
```

```
(item=/home/mkrausch/ansible-avd-cloudvision-demo/inventory/documentation)
```

```
ok: [DC1-SPINE1 -> localhost] =>
```

```
(item=/home/mkrausch/ansible-avd-cloudvision-demo/inventory/intended/configs)
```

```
ok: [DC1-SPINE1 -> localhost] =>
```

```
(item=/home/mkrausch/ansible-avd-cloudvision-demo/inventory/documentation/devices)
```

```
TASK [arista.avd.eos_cli_config_gen : Check if structure configuration file
exists] *****
```

```
skipping: [DC1-SPINE1]
```

```
skipping: [DC1-SPINE2]
```

```
skipping: [DC1-LEAF1A]
```

```
skipping: [DC1-LEAF1B]
```

```
skipping: [DC1-LEAF2A]
```

```
skipping: [DC1-LEAF2B]
```

```
skipping: [DC1-LEAF3A]
```

```
skipping: [DC1-LEAF3B]
```

```
TASK [arista.avd.eos_cli_config_gen : Include device intended structure
configuration variables] *****
```

skipping: [DC1-SPINE1]
skipping: [DC1-SPINE2]
skipping: [DC1-LEAF1A]
skipping: [DC1-LEAF1B]
skipping: [DC1-LEAF2A]
skipping: [DC1-LEAF2B]
skipping: [DC1-LEAF3A]
skipping: [DC1-LEAF3B]

TASK [arista.avd.eos_cli_config_gen : Generate eos intended configuration]

ok: [DC1-LEAF2A -> localhost]
ok: [DC1-SPINE1 -> localhost]
ok: [DC1-LEAF1B -> localhost]
ok: [DC1-LEAF1A -> localhost]
ok: [DC1-SPINE2 -> localhost]
ok: [DC1-LEAF3A -> localhost]
ok: [DC1-LEAF2B -> localhost]
ok: [DC1-LEAF3B -> localhost]

TASK [arista.avd.eos_cli_config_gen : Store checksum of existing device documentation]

ok: [DC1-SPINE2 -> localhost]
ok: [DC1-LEAF1B -> localhost]
ok: [DC1-LEAF1A -> localhost]
ok: [DC1-SPINE1 -> localhost]
ok: [DC1-LEAF2A -> localhost]
ok: [DC1-LEAF3B -> localhost]
ok: [DC1-LEAF3A -> localhost]
ok: [DC1-LEAF2B -> localhost]

TASK [arista.avd.eos_cli_config_gen : Generate content of device documentation]

ok: [DC1-SPINE2 -> localhost]
ok: [DC1-LEAF1B -> localhost]
ok: [DC1-LEAF2A -> localhost]
ok: [DC1-SPINE1 -> localhost]
ok: [DC1-LEAF1A -> localhost]
ok: [DC1-LEAF3A -> localhost]
ok: [DC1-LEAF3B -> localhost]
ok: [DC1-LEAF2B -> localhost]

TASK [arista.avd.eos_cli_config_gen : Generate TOC for device documentation]

ok: [DC1-SPINE2 -> localhost]
ok: [DC1-LEAF1B -> localhost]
ok: [DC1-SPINE1 -> localhost]
ok: [DC1-LEAF2A -> localhost]
ok: [DC1-LEAF1A -> localhost]
ok: [DC1-LEAF2B -> localhost]
ok: [DC1-LEAF3A -> localhost]

ok: [DC1-LEAF3B -> localhost]

PLAY [Configuration deployment with CVP]

TASK [arista.avd.eos_config_deploy_cvp : Create required output directories if not present] *****

ok: [cv_server -> localhost] =>

(item=/home/mkrausch/ansible-avd-cloudvision-demo/inventory/intended/structured_configs/cvp)

TASK [arista.avd.eos_config_deploy_cvp : Start creation/update process.]

included:

/root/.ansible/collections/ansible_collections/arista/avd/roles/eos_config_deploy_cvp/tasks/v3/main.yml for cv_server

TASK [arista.avd.eos_config_deploy_cvp : Generate intended variables]

ok: [cv_server]

TASK [arista.avd.eos_config_deploy_cvp : Build DEVICES and CONTAINER definition for cv_server] *****

ok: [cv_server -> localhost]

TASK [arista.avd.eos_config_deploy_cvp : Start creation/update process.]

included:

/root/.ansible/collections/ansible_collections/arista/avd/roles/eos_config_deploy_cvp/tasks/v3/present.yml for cv_server

TASK [arista.avd.eos_config_deploy_cvp : Load CVP device information for cv_server] *****

ok: [cv_server]

TASK [arista.avd.eos_config_deploy_cvp : Create configlets on CVP cv_server.]

changed: [cv_server]

TASK [arista.avd.eos_config_deploy_cvp : Building Containers topology on cv_server] *****

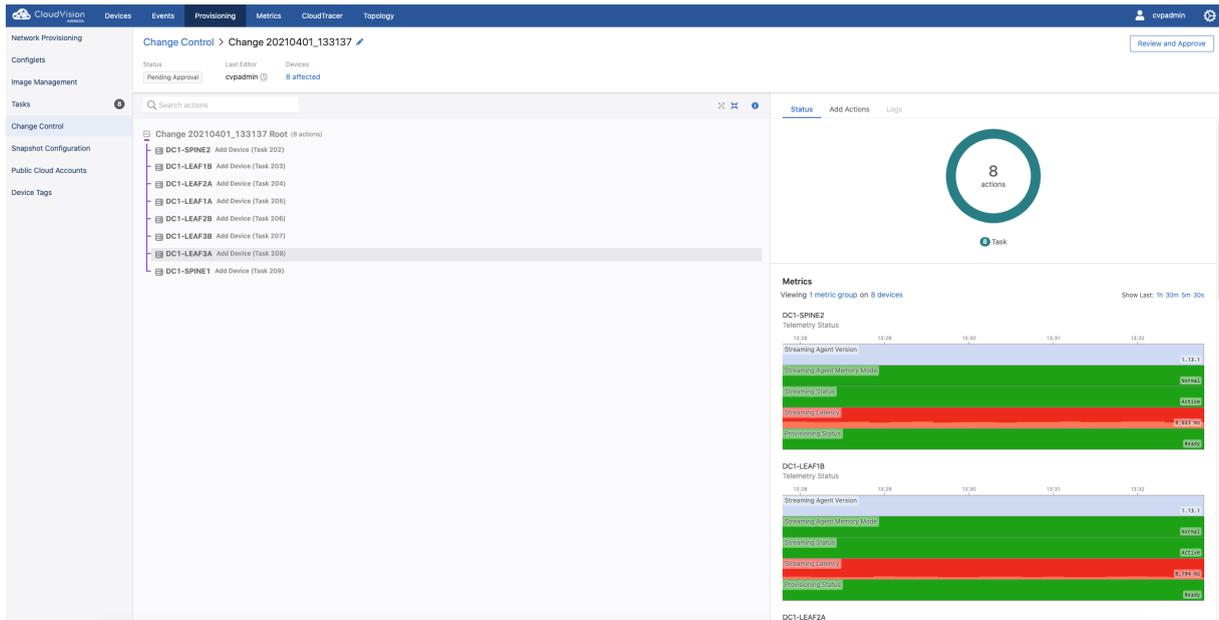
changed: [cv_server]

TASK [arista.avd.eos_config_deploy_cvp : Configure devices on cv_server]

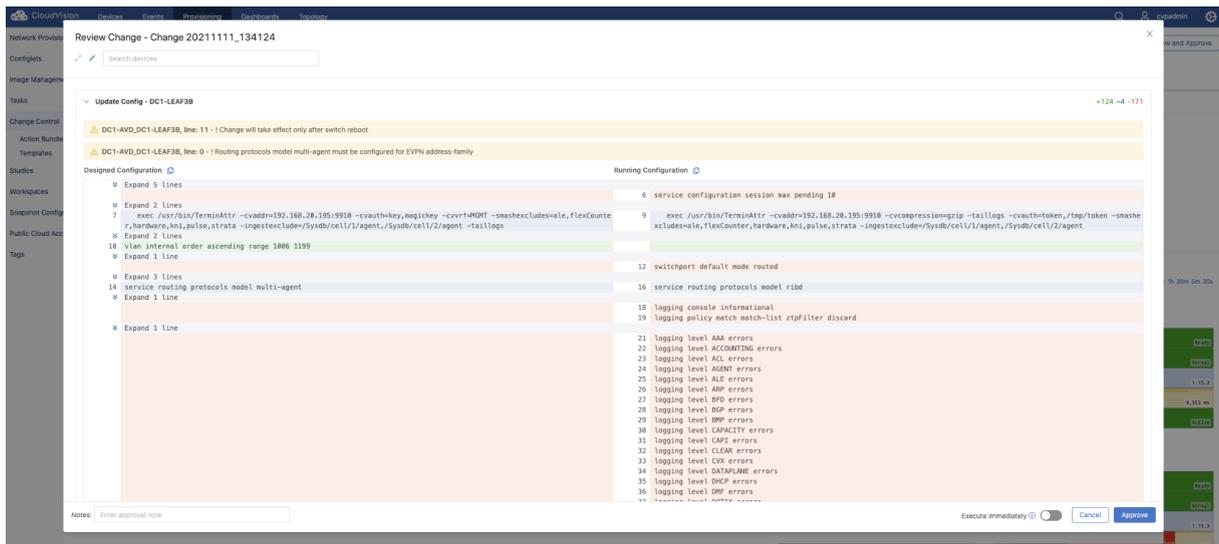
changed: [cv_server]

PLAY RECAP

DC1-LEAF1A		: ok=9	changed=0	unreachable=0	failed=0
skipped=2	rescued=0	ignored=0			
DC1-LEAF1B		: ok=9	changed=0	unreachable=0	failed=0
skipped=2	rescued=0	ignored=0			
DC1-LEAF2A		: ok=9	changed=0	unreachable=0	failed=0
skipped=2	rescued=0	ignored=0			
DC1-LEAF2B		: ok=9	changed=0	unreachable=0	failed=0
skipped=2	rescued=0	ignored=0			
DC1-LEAF3A		: ok=9	changed=0	unreachable=0	failed=0
skipped=2	rescued=0	ignored=0			
DC1-LEAF3B		: ok=9	changed=0	unreachable=0	failed=0
skipped=2	rescued=0	ignored=0			
DC1-SPINE1		: ok=16	changed=0	unreachable=0	failed=0
skipped=2	rescued=0	ignored=0			
DC1-SPINE2		: ok=9	changed=0	unreachable=0	failed=0
skipped=2	rescued=0	ignored=0			
cv_server		: ok=9	changed=3	unreachable=0	failed=0
skipped=0	rescued=0	ignored=0			



Possible differences and changes within the individual configurations will be displayed:



Click "Approve" -> "Execute Change Control" and then "Execute" to start the process.
 All devices will reboot to leave the ZTP state

The screenshot displays the CloudVision Change Control interface. The main header shows the change control ID 'Change 20211111_134124' and its status as 'Running'. A summary card indicates '8 actions' and '1 Task'. Below this, a table lists the actions for each device, all of which are 'Add Device' tasks. The 'Metrics' section shows a group of 8 devices with various performance metrics like 'Streaming Status', 'Streaming Agent version', and 'Streaming Latency'.

Device	Action	Status	Started
Change 20211111_134124 Root	(8 actions)	Running	3 minutes ago
DC1-LEAF1A	Add Device (Task 150)	Running	3 minutes ago
DC1-LEAF1B	Add Device (Task 151)	Running	3 minutes ago
DC1-LEAF2A	Add Device (Task 152)	Running	3 minutes ago
DC1-LEAF2B	Add Device (Task 153)	Running	3 minutes ago
DC1-LEAF3A	Add Device (Task 154)	Running	3 minutes ago
DC1-LEAF3B	Add Device (Task 155)	Running	3 minutes ago
DC1-SPINE1	Add Device (Task 156)	Running	3 minutes ago
DC1-SPINE2	Add Device (Task 157)	Running	3 minutes ago

The newly created Configlets can be reviewed and are marked with “Managed by Ansible”:

The screenshot shows the CloudVision interface with the 'Configlets' section selected. A table lists several configlets, all of which are marked as 'Managed by Ansible' in the 'Notes' column. The table columns are: Name, Containers, Devices, Notes, Type, Created By, and Created Date.

Name	Containers	Devices	Notes	Type	Created By	Created Date
ISG-SP7M-Multitenant-Setup	0	0	ISG-SP7M	Static	cpadmn	2021-08-20 15:08:05
DC1-AVD_DCH-LEAF1A	0	1	Managed by Ansible	Static	cpadmn	2021-11-10 17:15:39
DC1-AVD_DCH-LEAF1B	0	1	Managed by Ansible	Static	cpadmn	2021-11-10 17:15:38
DC1-AVD_DCH-LEAF2A	0	1	Managed by Ansible	Static	cpadmn	2021-11-10 17:15:39
DC1-AVD_DCH-LEAF2B	0	1	Managed by Ansible	Static	cpadmn	2021-11-10 17:15:38
DC1-AVD_DCH-LEAF3A	0	1	Managed by Ansible	Static	cpadmn	2021-11-10 17:15:39
DC1-AVD_DCH-LEAF3B	0	1	Managed by Ansible	Static	cpadmn	2021-11-10 17:15:38
DC1-AVD_DCH-SPINE1	0	1	Managed by Ansible	Static	cpadmn	2021-11-10 17:15:38
DC1-AVD_DCH-SPINE2	0	1	Managed by Ansible	Static	cpadmn	2021-11-10 17:15:40
DEMO-LEAF1-MLAG-PORTCHANNEL	0	0	MLAG Role	Static	cpadmn	2021-08-20 15:08:05
DEMO-LEAF1A-EVPN-SERVICES	0	0	MLAG Role	Static	cpadmn	2021-08-20 15:08:05

Updating Config Files and Provisioning via CVP

Whenever changing the config files via Ansible you can send the updated versions to CloudVision in order to activate them on the affected systems.

After changing any “.yaml” parameters first build new offline config files:

```
root@ubuntu:/home/mkrausch/ansible-avd-cloudvision-demo# ansible-playbook
playbooks/dc1-fabric-deploy-cvp.yml --tags build
[DEPRECATION WARNING]: COMMAND_WARNINGS option, the command warnings feature is
being removed. This feature will be removed from ansible-core in
version 2.14. Deprecation warnings can be disabled by setting
deprecation_warnings=False in ansible.cfg.

PLAY [Manage Arista EOS EVPN/VXLAN Configuration]
*****
*****
[WARNING]: Collection arista.cvp does not support Ansible version 2.11.3

TASK [arista.avd.build_output_folders : Cleanup existing folders in
/home/mkrausch/ansible-avd-cloudvision-demo/inventory/intended] *****
changed: [DC1-SPINE1 -> localhost]

TASK [arista.avd.build_output_folders : Create folder
/home/mkrausch/ansible-avd-cloudvision-demo/inventory/intended]
*****
changed: [DC1-SPINE1 -> localhost]

TASK [arista.avd.build_output_folders : Create folder structured_configs for
structured YAML files] *****
changed: [DC1-SPINE1 -> localhost]

TASK [arista.avd.build_output_folders : Create folder configs for EOS
Configuration files] *****
changed: [DC1-SPINE1 -> localhost]

TASK [arista.avd.build_output_folders : Create folder cvp for CVP structured YAML
files] *****
changed: [DC1-SPINE1 -> localhost]

TASK [arista.avd.build_output_folders : Create documentation folder:
documentation] *****
ok: [DC1-SPINE1 -> localhost]

TASK [arista.avd.build_output_folders : Create folder fabric for Fabric
documentation] *****
ok: [DC1-SPINE1 -> localhost]

TASK [arista.avd.build_output_folders : Create folder devices for EOS
documentation] *****
```

```

ok: [DC1-SPINE1 -> localhost]

TASK [arista.avd.build_output_folders : Create folder
/home/mkrausch/ansible-avd-cloudvision-demo/inventory/config_backup for EOS post
backup dir] ***
ok: [DC1-SPINE1 -> localhost]

TASK [arista.avd.build_output_folders : Create folder
/home/mkrausch/ansible-avd-cloudvision-demo/inventory/config_backup for EOS pre
backup dir] ***
ok: [DC1-SPINE1 -> localhost]

TASK [arista.avd.build_output_folders : Create folder reports for EOS state report
dir] *****
ok: [DC1-SPINE1 -> localhost]

TASK [arista.avd.build_output_folders : Create output directory for each EOS
device] *****
ok: [DC1-SPINE1 -> localhost]

TASK [arista.avd.eos_designs : Create required output directories if not present]
*****
ok: [DC1-SPINE1 -> localhost] =>
(item=/home/mkrausch/ansible-avd-cloudvision-demo/inventory/intended/structured_co
nfigs)
ok: [DC1-SPINE1 -> localhost] =>
(item=/home/mkrausch/ansible-avd-cloudvision-demo/inventory/documentation/fabric)

TASK [arista.avd.eos_designs : Set AVD facts]
*****
*****
ok: [DC1-SPINE1 -> localhost]
ok: [DC1-SPINE2 -> localhost]
ok: [DC1-LEAF2A -> localhost]
ok: [DC1-LEAF1A -> localhost]
ok: [DC1-LEAF1B -> localhost]
ok: [DC1-LEAF3A -> localhost]
ok: [DC1-LEAF2B -> localhost]
ok: [DC1-LEAF3B -> localhost]

TASK [arista.avd.eos_designs : Set AVD topology facts]
*****
*****
ok: [DC1-SPINE1 -> localhost]
ok: [DC1-SPINE2 -> localhost]
ok: [DC1-LEAF1A -> localhost]
ok: [DC1-LEAF1B -> localhost]
ok: [DC1-LEAF2A -> localhost]
ok: [DC1-LEAF2B -> localhost]
ok: [DC1-LEAF3A -> localhost]
ok: [DC1-LEAF3B -> localhost]

```

TASK [arista.avd.eos_designs : Generate YAML file with hostvars (only for debugging)] *****

ok: [DC1-SPINE1 -> localhost]
ok: [DC1-LEAF1B -> localhost]
ok: [DC1-LEAF1A -> localhost]
ok: [DC1-LEAF2A -> localhost]
ok: [DC1-SPINE2 -> localhost]
ok: [DC1-LEAF2B -> localhost]
ok: [DC1-LEAF3A -> localhost]
ok: [DC1-LEAF3B -> localhost]

TASK [arista.avd.eos_designs : Generate device configuration in structured format] *****

ok: [DC1-SPINE1 -> localhost]
ok: [DC1-SPINE2 -> localhost]
ok: [DC1-LEAF1A -> localhost]
ok: [DC1-LEAF2A -> localhost]
ok: [DC1-LEAF1B -> localhost]
ok: [DC1-LEAF3A -> localhost]
ok: [DC1-LEAF2B -> localhost]
ok: [DC1-LEAF3B -> localhost]

TASK [arista.avd.eos_designs : Write device structured configuration to YAML file] *****

ok: [DC1-LEAF1B -> localhost]
ok: [DC1-LEAF2A -> localhost]
ok: [DC1-SPINE1 -> localhost]
ok: [DC1-SPINE2 -> localhost]
ok: [DC1-LEAF1A -> localhost]
ok: [DC1-LEAF2B -> localhost]
ok: [DC1-LEAF3A -> localhost]
ok: [DC1-LEAF3B -> localhost]

TASK [arista.avd.eos_designs : Include device structured configuration, that was previously generated.] *****

ok: [DC1-SPINE1 -> localhost]
ok: [DC1-SPINE2 -> localhost]
ok: [DC1-LEAF1A -> localhost]
ok: [DC1-LEAF1B -> localhost]
ok: [DC1-LEAF2A -> localhost]
ok: [DC1-LEAF2B -> localhost]
ok: [DC1-LEAF3A -> localhost]
ok: [DC1-LEAF3B -> localhost]

TASK [arista.avd.eos_designs : Store checksum of existing fabric documentation] *****

ok: [DC1-SPINE1 -> localhost]

TASK [arista.avd.eos_designs : Generate fabric documentation in Markdown Format.] *****

ok: [DC1-SPINE1 -> localhost]

TASK [arista.avd.eos_designs : Generate TOC for fabric documentation]

ok: [DC1-SPINE1 -> localhost]

TASK [arista.avd.eos_designs : Generate fabric point-to-point links summary in csv format.] *****

ok: [DC1-SPINE1 -> localhost]

TASK [arista.avd.eos_designs : Generate fabric topology in csv format.]

ok: [DC1-SPINE1 -> localhost]

TASK [arista.avd.eos_cli_config_gen : Create required output directories if not present] *****

ok: [DC1-SPINE1 -> localhost] =>

(item=/home/mkrausch/ansible-avd-cloudvision-demo/inventory/intended/structured_configs)

ok: [DC1-SPINE1 -> localhost] =>

(item=/home/mkrausch/ansible-avd-cloudvision-demo/inventory/documentation)

ok: [DC1-SPINE1 -> localhost] =>

(item=/home/mkrausch/ansible-avd-cloudvision-demo/inventory/intended/configs)

ok: [DC1-SPINE1 -> localhost] =>

(item=/home/mkrausch/ansible-avd-cloudvision-demo/inventory/documentation/devices)

TASK [arista.avd.eos_cli_config_gen : Check if structure configuration file exists] *****

skipping: [DC1-SPINE1]

skipping: [DC1-SPINE2]

skipping: [DC1-LEAF1A]

skipping: [DC1-LEAF1B]

skipping: [DC1-LEAF2A]

skipping: [DC1-LEAF2B]

skipping: [DC1-LEAF3A]

skipping: [DC1-LEAF3B]

TASK [arista.avd.eos_cli_config_gen : Include device intended structure configuration variables] *****

skipping: [DC1-SPINE1]

skipping: [DC1-SPINE2]

skipping: [DC1-LEAF1A]

skipping: [DC1-LEAF1B]

skipping: [DC1-LEAF2A]

skipping: [DC1-LEAF2B]

skipping: [DC1-LEAF3A]

skipping: [DC1-LEAF3B]

TASK [arista.avd.eos_cli_config_gen : Generate eos intended configuration]

changed: [DC1-LEAF1A -> localhost]

```
changed: [DC1-LEAF2A -> localhost]
changed: [DC1-SPINE2 -> localhost]
changed: [DC1-SPINE1 -> localhost]
changed: [DC1-LEAF1B -> localhost]
changed: [DC1-LEAF3B -> localhost]
changed: [DC1-LEAF2B -> localhost]
changed: [DC1-LEAF3A -> localhost]
```

```
TASK [arista.avd.eos_cli_config_gen : Store checksum of existing device
documentation] *****
```

```
ok: [DC1-LEAF1A -> localhost]
ok: [DC1-SPINE1 -> localhost]
ok: [DC1-SPINE2 -> localhost]
ok: [DC1-LEAF2A -> localhost]
ok: [DC1-LEAF1B -> localhost]
ok: [DC1-LEAF2B -> localhost]
ok: [DC1-LEAF3A -> localhost]
ok: [DC1-LEAF3B -> localhost]
```

```
TASK [arista.avd.eos_cli_config_gen : Generate content of device documentation]
*****
```

```
ok: [DC1-SPINE2 -> localhost]
ok: [DC1-LEAF1A -> localhost]
ok: [DC1-LEAF2A -> localhost]
ok: [DC1-SPINE1 -> localhost]
ok: [DC1-LEAF1B -> localhost]
ok: [DC1-LEAF2B -> localhost]
ok: [DC1-LEAF3A -> localhost]
ok: [DC1-LEAF3B -> localhost]
```

```
TASK [arista.avd.eos_cli_config_gen : Generate TOC for device documentation]
*****
```

```
ok: [DC1-SPINE1 -> localhost]
ok: [DC1-SPINE2 -> localhost]
changed: [DC1-LEAF1B -> localhost]
changed: [DC1-LEAF1A -> localhost]
ok: [DC1-LEAF2A -> localhost]
ok: [DC1-LEAF2B -> localhost]
ok: [DC1-LEAF3A -> localhost]
ok: [DC1-LEAF3B -> localhost]
```

```
PLAY [Configuration deployment with CVP]
```

```
*****
*****
```

```
TASK [arista.avd.eos_config_deploy_cvp : Create required output directories if not
present] *****
```

```
ok: [cv_server -> localhost] =>
(item=/home/mkrausch/ansible-avd-cloudvision-demo/inventory/intended/structured_co
nfigs/cvp)
```

```
TASK [arista.avd.eos_config_deploy_cvp : Start creation/update process.]
*****
included:
/root/.ansible/collections/ansible_collections/arista/avd/roles/eos_config_deploy_
cvp/tasks/v3/main.yml for cv_server
```

```
TASK [arista.avd.eos_config_deploy_cvp : Generate intended variables]
*****
ok: [cv_server]
```

```
TASK [arista.avd.eos_config_deploy_cvp : Build DEVICES and CONTAINER definition
for cv_server] *****
changed: [cv_server -> localhost]
```

```
PLAY RECAP
*****
*****
```

DC1-LEAF1A		: ok=10	changed=2	unreachable=0	failed=0
skipped=2	rescued=0	ignored=0			
DC1-LEAF1B		: ok=10	changed=2	unreachable=0	failed=0
skipped=2	rescued=0	ignored=0			
DC1-LEAF2A		: ok=10	changed=1	unreachable=0	failed=0
skipped=2	rescued=0	ignored=0			
DC1-LEAF2B		: ok=10	changed=1	unreachable=0	failed=0
skipped=2	rescued=0	ignored=0			
DC1-LEAF3A		: ok=10	changed=1	unreachable=0	failed=0
skipped=2	rescued=0	ignored=0			
DC1-LEAF3B		: ok=10	changed=1	unreachable=0	failed=0
skipped=2	rescued=0	ignored=0			
DC1-SPINE1		: ok=29	changed=6	unreachable=0	failed=0
skipped=2	rescued=0	ignored=0			
DC1-SPINE2		: ok=10	changed=1	unreachable=0	failed=0
skipped=2	rescued=0	ignored=0			
cv_server		: ok=4	changed=1	unreachable=0	failed=0
skipped=0	rescued=0	ignored=0			

In the next step you can send the updated config files to CloudVision:

```
root@ubuntu:/home/mkrausch/ansible-avd-cloudvision-demo# ansible-playbook
playbooks/dc1-upload-configlets.yml
[DEPRECATION WARNING]: COMMAND_WARNINGS option, the command warnings feature is
being removed. This feature will be removed from ansible-core in
version 2.14. Deprecation warnings can be disabled by setting
deprecation_warnings=False in ansible.cfg.
```

```
PLAY [Configlet upload management]
*****
*****
[WARNING]: Collection arista.cvp does not support Ansible version 2.11.3
```

```

TASK [arista.avd.cvp_configlet_upload : generate intended variables]
*****
ok: [cv_server]

TASK [arista.avd.cvp_configlet_upload : Execute upload with collection in version
v1] *****
included:
/root/.ansible/collections/ansible_collections/arista/avd/roles/cvp_configlet_uplo
ad/tasks/cv_collection_v1.yml for cv_server

TASK [arista.avd.cvp_configlet_upload : collecting facts from CVP cv_server.]
*****
ok: [cv_server]

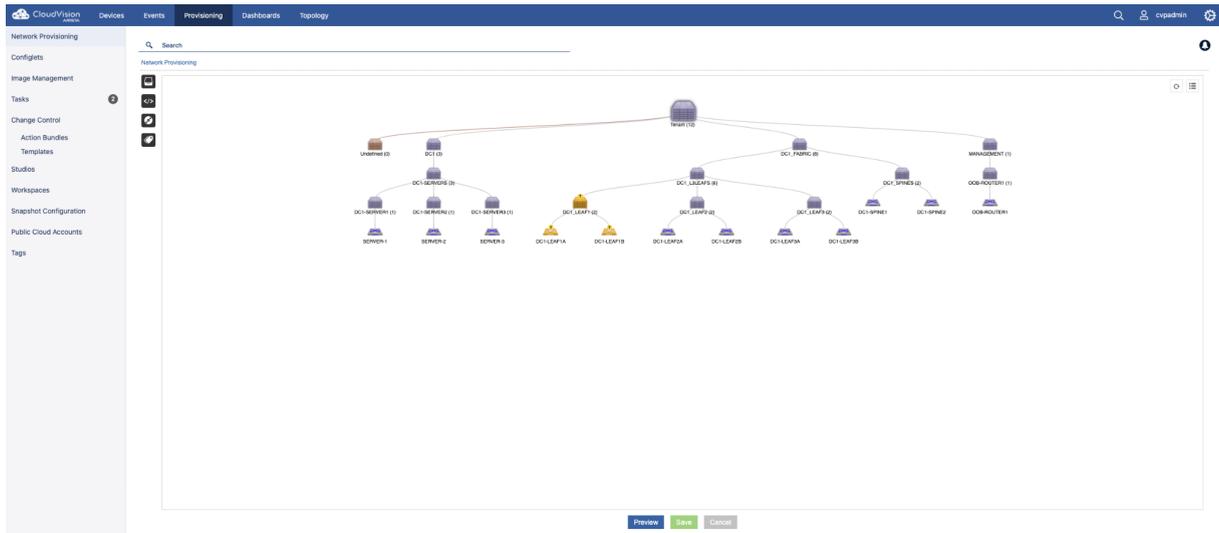
TASK [arista.avd.cvp_configlet_upload : create configlets on CVP cv_server.]
*****
changed: [cv_server]

TASK [arista.avd.cvp_configlet_upload : Execute pending tasks on cv_server]
*****
skipping: [cv_server]

PLAY RECAP
*****
*****
cv_server          : ok=4    changed=1    unreachable=0    failed=0
skipped=1    rescued=0    ignored=0

```

Ansible will update the existing Configlets on CVP and create new Tasks:



Creating a Change Control will update the Arista systems with the new configurations:

ID	Device	Creator	Type	Updated	Status	Change Control
159	DCI-LEAF1A MAC: 50:00:00:15:F4:e8 IP: 192.168.20.111	cvpadmin	Update Config	2 minutes ago	Pending	
158	DCI-LEAF1B MAC: 50:00:00:72:8b:31 IP: 192.168.20.112	cvpadmin	Update Config	2 minutes ago	Pending	
157	DCI-SPINE2 MAC: 50:00:00:6b:2e:70 IP: 192.168.20.102	cvpadmin	Add Device	26 minutes ago	Completed	Change 20211111_134124
156	DCI-SPINE1 MAC: 50:00:00:03:37:66 IP: 192.168.20.101	cvpadmin	Add Device	26 minutes ago	Completed	Change 20211111_134124
155	DCI-LEAF3B MAC: 50:00:00:88:1e:27 IP: 192.168.20.116	cvpadmin	Add Device	26 minutes ago	Completed	Change 20211111_134124

CloudVision **Provisioning** Dashboards Topology cpadmin

Change Control > Change 20211111_141655 ⏪ Rollback

Status: Completed | Scheduled For: Unscheduled | Approver: cpadmin | Started: Nov 11, 2021 14:18:18 | Last Editor: cpadmin | Devices: 2 affected | UID: az2P20MXS | Help

Tasks

Change Control

- Change 20211111_141655 Root (2 actions) ✓
- DC1-LEAF1B Update Config (Task 184) +0 -1 -0 ✓
- DC1-LEAF1A Update Config (Task 185) +0 -1 -0 ✓

Status | Add Actions | Logs

2 actions

Task

Action Status 2 of 2

Metrics
Viewing 1 metric group on 2 devices Show List: 7h 30m 5m 30s

DC1-LEAF1B
Telemetry Status

DC1-LEAF1A
Telemetry Status

Customized Playbooks with Arista AVD

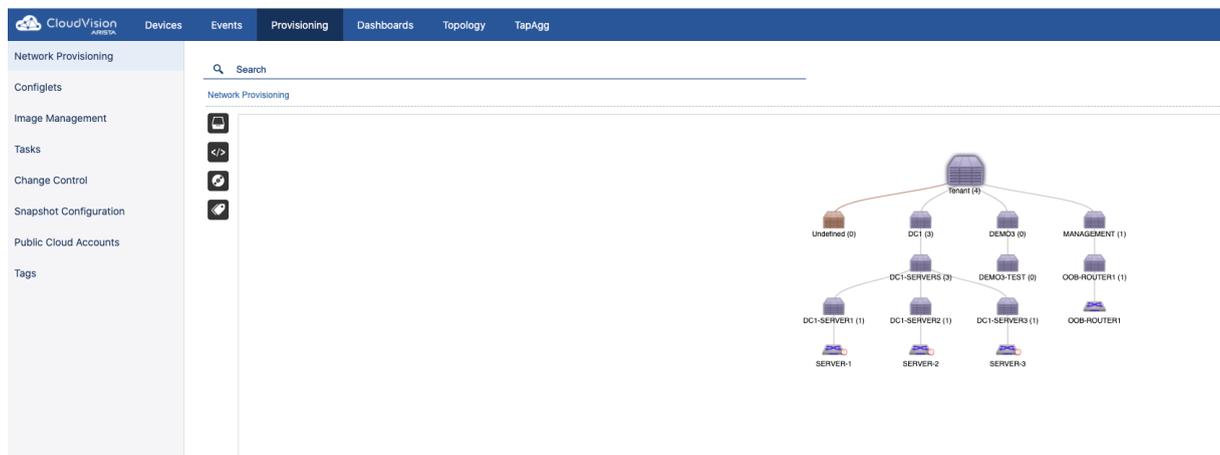
It's also possible to create customized playbooks to configure EOS based systems via CVP based on the Arista Ansible AVD collection.

Further information about the currently available modules can be found at:

<https://cvp.avd.sh/en/latest/>

Within the following example a simple playbook “**create-containers.yml**” will create Containers on CloudVision and assign an already existing configlet with the name “**OOB_BASE**” to a Container.

Here is a snapshot of CloudVision before the Playbook has been ran



Displaying and running the Playbook:

```
root@ubuntu:/home/mkrausch/ansible-avd-cloudvision-demo# cat
playbooks/create-containers.yml
```

```
- name: Playbook to create containers on CVP.
  hosts: cv_server
  connection: local
  gather_facts: no
  collections:
    - arista.cvp
  vars:
    # Container definition
    verbose: False
    containers:
      DEMO-Fabric:
        parent_container: Tenant
        # Configlets must exist on CVP
        configlets:
          - OOB_BASE
      DEMO-Spines:
        parent_container: DEMO-Fabric
```

```
DEMO-Leaves:
  parent_container: DEMO-Fabric
DEMO-MLAG01:
  parent_container: DEMO-Leaves
```

tasks:

- name: Gather CVP facts
 - arista.cvp.cv_facts:
 - register: CVP_FACTS
- name: Build Container topology on CVP
 - arista.cvp.cv_container:
 - cvp_facts: '{{CVP_FACTS.ansible_facts}}'
 - topology: '{{containers}}'
 - mode: merge
 - register: CVP_CONTAINERS_RESULT

root@ubuntu:/home/mkrausch/ansible-avd-cloudvision-demo#

root@ubuntu:/home/mkrausch/ansible-avd-cloudvision-demo# **ansible-playbook
playbooks/create-containers.yml**

PLAY [Playbook to create containers on CVP.]

```
*****
*****
*****
```

TASK [Gather CVP facts]

```
*****
*****
*****
```

ok: [cv_server]

TASK [Build Container topology on CVP]

```
*****
*****
*****
```

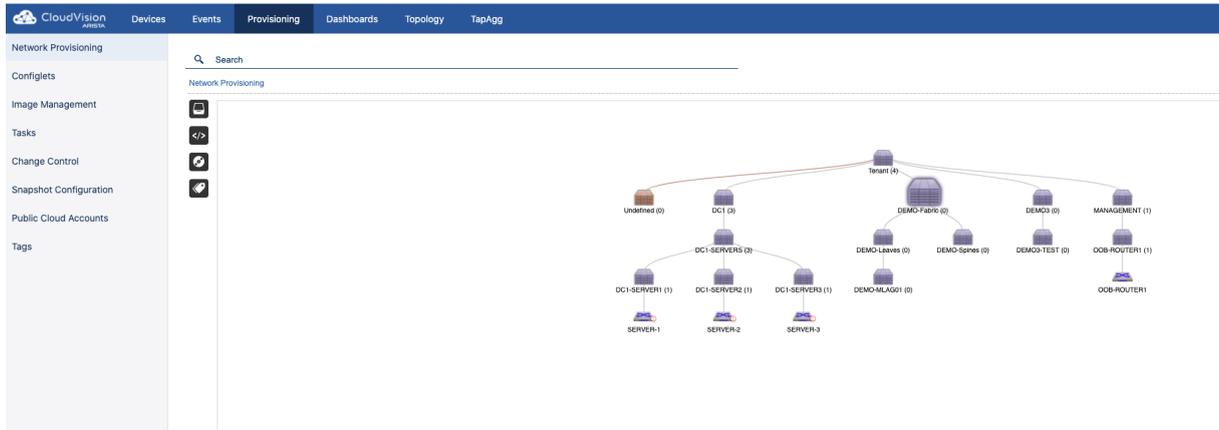
ok: [cv_server]

PLAY RECAP

```
*****
*****
*****
```

```
cv_server      : ok=2    changed=0    unreachable=0    failed=0
skipped=0     rescued=0    ignored=0
```

After the Playbook has been executed the new Containers have been created in CloudVision and the Configlet “OOB_BASE” has been assigned correctly to the Container “DEMO-Fabric”:



Network Provisioning > DEMO-Fabric > Configlet

Item	Notes	Type	Created By	Created Date	Proposed Configuration	Expand All
<input type="checkbox"/> RSP-SPIN Multitenant Setup		Static	opadmin	2021-09-28 13:08:03		
<input type="checkbox"/> DC1-AWD_DC1-LEAF1A	Managed by Ansible	Static	opadmin	2021-09-28 12:15:13	OOB_BASE	
<input type="checkbox"/> DC1-AWD_DC1-LEAF1B	Managed by Ansible	Static	opadmin	2021-09-28 12:15:14		
<input type="checkbox"/> DC1-AWD_DC1-LEAF2A	Managed by Ansible	Static	opadmin	2021-09-28 12:15:16		
<input type="checkbox"/> DC1-AWD_DC1-LEAF2B	Managed by Ansible	Static	opadmin	2021-09-28 12:15:12		
<input type="checkbox"/> DC1-AWD_DC1-LEAF3A	Managed by Ansible	Static	opadmin	2021-09-28 12:15:04		
<input type="checkbox"/> DC1-AWD_DC1-LEAF3B	Managed by Ansible	Static	opadmin	2021-09-28 12:15:07		
<input type="checkbox"/> DC1-AWD_DC1-SPINE1	Managed by Ansible	Static	opadmin	2021-09-28 12:15:06		
<input type="checkbox"/> DC1-AWD_DC1-SPINE2	Managed by Ansible	Static	opadmin	2021-09-28 12:15:09		

Docker Virtualization on EOS



A **Docker container** is a standard unit of software that packages up code and all its dependencies so the application runs quickly and reliably from one computing environment to another.

A Docker container image is a lightweight, standalone, executable package of software that includes everything needed to run an application: code, runtime, system tools, system libraries and settings.

Container **images** become containers at **runtime** and in the case of Docker containers - images become containers when they run on a **Docker Engine**.

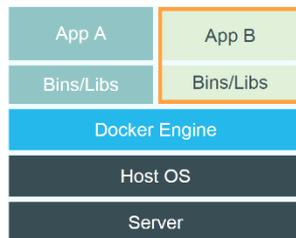
Available for both **Linux and Windows-based applications**, containerized software will always run the same, regardless of the infrastructure. **Containers isolate software** from its environment and ensure that it works uniformly despite differences for instance between development and staging.

Differences between Docker Containers and VMs

Docker Containers:

Containers are an abstraction at the app layer that packages code and dependencies together.

Multiple containers can run on the same machine and share the OS kernel with other containers, each running as **isolated processes** in user space. Containers take up **less space than VMs** (container images are typically tens of MBs in size), can handle **more applications** and require **fewer VMs and Operating systems**.



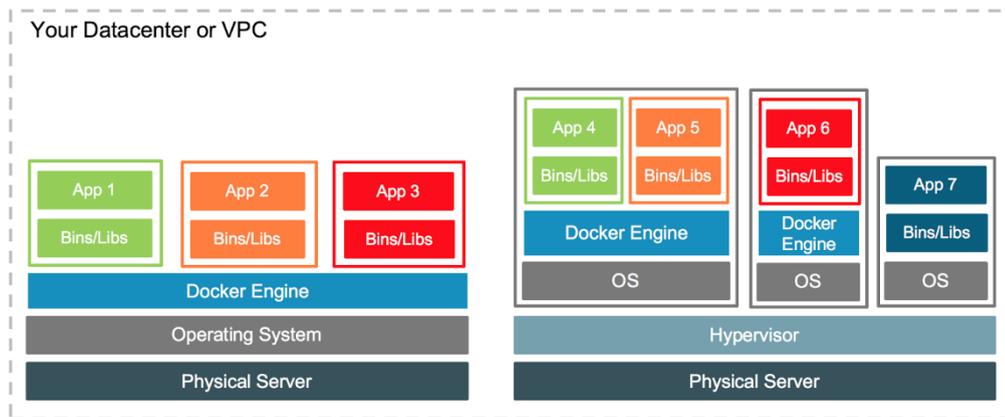
Virtual machines:

Virtual machines (VMs) are an **abstraction of physical hardware** turning one server into many servers. The **hypervisor** allows multiple VMs to run on a single machine. **Each VM includes a full copy of an operating system**, the application, necessary binaries and libraries - taking up a lot of memory and storage. VMs may also be **slow to boot**.



VMs and Docker Containers are different but not mutually exclusive:

By mixing and matching Docker hosts with “traditional” VMs, sysadmins can be assured they are getting the maximum utilization out of their physical hardware especially when Docker Engines are running on Linux and Windows OS in parallel on the same physical Server.



Docker support on EOS

Docker is already supported in Linux based EOS!

It may be necessary to define a **DNS Server** (here e.g. 8.8.8.8) to pull Docker Images and to make sure that the address <https://hub.docker.com> can be accessed from the system to make sure that new Images can be pulled from there.

Also take into consideration possible **VRF dependencies** if configured on the EOS system!

```
hostname Router-1
ip name-server vrf default 8.8.8.8
!
spanning-tree mode mstp
!
no aaa root
!
username admin privilege 15 role network-admin secret sha512
$6$e2J8z/HDC03RmLPu$i02zAYE/7wOfySAKI0PbgL7r6PA6IUfsh6Bf7puI1Zi11eUP1bxQgr73JHItjH
vkGyJMj5X1vR0xvRp7E9iPf.
!
interface Management1
    ip address 192.168.10.71/24
!
ip route 0.0.0.0/0 192.168.10.1
```

```
Router-1#ping www.docker.com
```

```
PING d1syzps6kort6n.cloudfront.net (99.84.151.2) 72(100) bytes of data.
80 bytes from server-99-84-151-2.tx152.r.cloudfront.net (99.84.151.2): icmp_seq=1
ttl=246 time=18.0 ms
80 bytes from server-99-84-151-2.tx152.r.cloudfront.net (99.84.151.2): icmp_seq=2
ttl=246 time=17.7 ms
80 bytes from server-99-84-151-2.tx152.r.cloudfront.net (99.84.151.2): icmp_seq=3
ttl=246 time=17.0 ms
```

```
80 bytes from server-99-84-151-2.tx152.r.cloudfront.net (99.84.151.2): icmp_seq=4  
ttl=246 time=17.2 ms  
80 bytes from server-99-84-151-2.tx152.r.cloudfront.net (99.84.151.2): icmp_seq=5  
ttl=246 time=17.5 ms
```

```
--- d1syzps6kort6n.cloudfront.net ping statistics ---  
5 packets transmitted, 5 received, 0% packet loss, time 78ms  
rtt min/avg/max/mdev = 17.054/17.545/18.097/0.380 ms, pipe 2, ipg/ewma  
19.642/17.810 ms
```

Installing Ubuntu in a Docker Container via Bash

Docker will place the Ubuntu container in its default **bridge/nat mode**.

Connections to the “outer world” will use the OOB Management Interface per default.

```
Router-1#bash
[admin@Router-1 ~]$ sudo su
bash-4.3# service docker start
Starting docker (via systemctl): Starting docker:
.....
.....[ OK ]
[ OK ]
```

Testing Docker via Bash

The integrated Docker “Hello-World” test can be used to check that Docker is running correctly on the system.

```
bash-4.3# docker run hello-world
Unable to find image 'hello-world:latest' locally
latest: Pulling from library/hello-world
d8aec4eeb95f: Pull complete
Digest: sha256:c3b4ada4687bbaa170745b3e4dd8ac3f194ca95b2d0518b417fb47e5879d9b5f
Status: Downloaded newer image for hello-world:latest
```

Hello from Docker!

This message shows that your installation appears to be working correctly.

To generate this message, Docker took the following steps:

1. The Docker client contacted the Docker daemon.
2. The Docker daemon pulled the “hello-world” image from the Docker Hub.
(i386)
3. The Docker daemon created a new container from that image which runs the executable that produces the output you are currently reading.
4. The Docker daemon streamed that output to the Docker client, which sent it to your terminal.

To try something more ambitious, you can run an Ubuntu container with:

```
$ docker run -it ubuntu bash
```

Share images, automate workflows, and more with a free Docker ID:

<https://hub.docker.com/>

For more examples and ideas, visit:

<https://docs.docker.com/get-started/>

```
bash-4.3# docker ps -a
CONTAINER ID        IMAGE               COMMAND             CREATED
STATUS            PORTS              NAMES
9fdafd40915d      hello-world        "/hello"           4 hours ago
Exited (0) 4 hours ago                distracted_babbage
```

Creating a new Ubuntu Image and a Docker Container via Bash

```
bash-4.3# docker run -it ubuntu bash
Unable to find image 'ubuntu:latest' locally
latest: Pulling from library/ubuntu
9273eca053fc: Downloading 7.806MB/27.12MB
32bb043e9224: Download complete
a9526b45dc3f: Download complete
694d3060ad92: Download complete
9273eca053fc: Pull complete
32bb043e9224: Pull complete
a9526b45dc3f: Pull complete
694d3060ad92: Pull complete
Digest: sha256:6e9f67fa63b0323e9a1e587fd71c561ba48a034504fb804fd26fd8800039835d
Status: Downloaded newer image for ubuntu:latest
root@da5b00c48763:/#
root@da5b00c48763:/# exit
bash-4.3#
bash-4.3# docker ps -a
CONTAINER ID        IMAGE               COMMAND             CREATED
STATUS            PORTS              NAMES
da5b00c48763      ubuntu             "bash"             3 minutes ago
Exited (100) About a minute ago    nostalgic_wescoff
9fdafd40915d      hello-world        "/hello"           3 hours ago
Exited (0) 3 hours ago             distracted_babbage
```

Removing a Docker Container via Bash

```
bash-4.3# docker stop da5b00c48763
bash-4.3# docker rm da5b00c48763
da5b00c48763
bash-4.3# docker ps -a
CONTAINER ID        IMAGE               COMMAND             CREATED
STATUS            PORTS              NAMES
9fdafd40915d      hello-world        "/hello"           4 hours ago
Exited (0) 4 hours ago             distracted_babbage
```

Creating a new Ubuntu Container and configuring Ubuntu via Bash

As the Ubuntu Docker Container doesn't include many features by default some basic applications have to be installed.

```
bash-4.3# docker run -it ubuntu bash
root@f205bfb1df04:/# apt-get update
Get:1 http://security.ubuntu.com/ubuntu bionic-security InRelease [88.7 kB]
Get:2 http://archive.ubuntu.com/ubuntu bionic InRelease [242 kB]
Get:3 http://security.ubuntu.com/ubuntu bionic-security/multiverse i386
Packages [4510 B]
Get:4 http://security.ubuntu.com/ubuntu bionic-security/universe i386
Packages [754 kB]
Get:5 http://archive.ubuntu.com/ubuntu bionic-updates InRelease [88.7 kB]
Get:6 http://archive.ubuntu.com/ubuntu bionic-backports InRelease [74.6 kB]
```

```

Get:7 http://archive.ubuntu.com/ubuntu bionic/multiverse i386 Packages [177
kB]
Get:8 http://archive.ubuntu.com/ubuntu bionic/main i386 Packages [1328 kB]
...
root@ f205bfb1df04:/# apt-get install nano
Reading package lists... Done
Building dependency tree
Reading state information... Done
Suggested packages:
  spell
The following NEW packages will be installed:
  nano
0 upgraded, 1 newly installed, 0 to remove and 0 not upgraded.
...
root@ f205bfb1df04:/# apt-get install net-tools
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following NEW packages will be installed:
  net-tools
0 upgraded, 1 newly installed, 0 to remove and 0 not upgraded.
Need to get 185 kB of archives.
After this operation, 745 kB of additional disk space will be used.
Get:1 http://archive.ubuntu.com/ubuntu bionic/main i386 net-tools i386
1.60+git20161116.90da8a0-1ubuntu1 [185 kB]
Fetched 185 kB in 6s (32.3 kB/s)
debconf: delaying package configuration, since apt-utils is not installed
Selecting previously unselected package net-tools.
(Reading database ... 4108 files and directories currently installed.)
Preparing to unpack
.../net-tools_1.60+git20161116.90da8a0-1ubuntu1_i386.deb ...
Unpacking net-tools (1.60+git20161116.90da8a0-1ubuntu1) ...
Setting up net-tools (1.60+git20161116.90da8a0-1ubuntu1) ...
root@f205bfb1df04:/# apt-get install iputils-ping
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  libcap2 libcap2-bin libidn11 libpam-cap
The following NEW packages will be installed:
  iputils-ping libcap2 libcap2-bin libidn11 libpam-cap
0 upgraded, 5 newly installed, 0 to remove and 0 not upgraded.
Need to get 145 kB of archives.
After this operation, 530 kB of additional disk space will be used.
Do you want to continue? [Y/n] y
Get:1 http://archive.ubuntu.com/ubuntu bionic/main i386 libcap2 i386
1:2.25-1.2 [13.4 kB]
Get:2 http://archive.ubuntu.com/ubuntu bionic-updates/main i386 libidn11
i386 1.33-2.1ubuntu1.2 [46.8 kB]
...

```

Docker Networking Modes

Bridge mode:

This is the **default mode** of Docker networking. A container comes online, bridges itself to the docker0 interface and is natted to the outside world.

Host mode:

This method allows a container to see the host's native networking stack. So it can bridge to an interface or add kernel routes.

None:

Not connected to an interface.

Macvlan:

Allows an interface to be bridged to a physical interface or subinterface. However, this differs from host mode as it does not directly talk to host networking. Each container receives a unique mac address.

IPvlan:

Is similar to Macvlan however, each container can receive the same mac address with different IP address. IPvlan also allows for routing on the host or similar to router on a stick.

VXLAN

VXLAN mode is the default networking mode for host overlays in Docker Swarm mode.

Configuring a MACVLAN Interface on Docker and attaching it to an existing Container via Bash

```
root@f205bfb1df04:/# ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 172.17.0.2 netmask 255.255.0.0 broadcast 0.0.0.0
    ether 02:42:ac:11:00:02 txqueuelen 0 (Ethernet)
    RX packets 10910 bytes 17813573 (17.8 MB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 9215 bytes 669275 (669.2 KB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
    inet 127.0.0.1 netmask 255.0.0.0
    loop txqueuelen 1 (Local Loopback)
    RX packets 0 bytes 0 (0.0 B)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 0 bytes 0 (0.0 B)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
root@f205bfb1df04:/# exit
exit
bash-4.3# docker ps -a
CONTAINER ID        IMAGE               COMMAND             CREATED
STATUS            PORTS              NAMES
f205bfb1df04       ubuntu             "bash"             2 hours ago
Exited (0) 4 minutes ago
d0599c9f5a0f       hello-world        "/hello"           3 hours ago
Exited (0) 3 hours ago
bash-4.3# docker inspect f205bfb1df04
[
  {
    "Id": "f205bfb1df04740a0db1f999a73c680b4d47d397b00b4d7d0bd7be000ab0637",
    "Created": "2019-11-07T10:54:28.359813484Z",
    "Path": "bash",
    "Args": [],
    "State": {
      "Status": "exited",
      "Running": false,
      "Paused": false,
      ...
    "NetworkSettings": {
      "Bridge": "",
      "SandboxID":
"b8767a7111b7bf1949ca0f3d57efdc49537b562844901bd8e39be671f1eaf00e",
      "HairpinMode": false,
      "LinkLocalIPv6Address": "",
      "LinkLocalIPv6PrefixLen": 0,
      "Ports": {},
      "SandboxKey": "/var/run/docker/netns/b8767a7111b7",
      "SecondaryIPAddresses": null,
      "SecondaryIPv6Addresses": null,
      "EndpointID": "",
      "Gateway": "",
      "GlobalIPv6Address": "",
      "GlobalIPv6PrefixLen": 0,
      "IPAddress": "",
```

```

    "IPPrefixLen": 0,
    "IPv6Gateway": "",
    "MacAddress": "",
    "Networks": {
      "bridge": {
        "IPAMConfig": null,
        "Links": null,
        "Aliases": null,
        "NetworkID":
"1bcf3ad2e9fab255fc47bcc98f1ab8712bfc34cb02f9b9816dcaae6f2485682f",
        "EndpointID": "",
        "Gateway": "",
        "IPAddress": "",
        "IPPrefixLen": 0,
        "IPv6Gateway": "",
        "GlobalIPv6Address": "",
        "GlobalIPv6PrefixLen": 0,
        "MacAddress": "",
        "DriverOpts": null
      }
    }
  }
}
]

```

```

bash-4.3# docker network create -d macvlan --subnet=192.168.20.0/24
--ip-range=192.168.20.128/25 --gateway=192.168.20.10 -o parent=vlan20
vlan20-192.168.20.0 && ip link add mac0 link vlan20 type macvlan mode bridge
4a56867b974d11d1af5c710dd804a7b1f484d04e0eadf294df1261a4ce63a83f

```

```
bash-4.3# docker network ls
```

NETWORK ID	NAME	DRIVER	SCOPE
1bcf3ad2e9fa	bridge	bridge	local
0bc348ca7b04	host	host	local
a5d7d9b0d835	none	null	local
4a56867b974d	vlan20-192.168.20.0	macvlan	local

```
bash-4.3# docker ps -a
```

CONTAINER ID	IMAGE	COMMAND	CREATED
f205bfb1df04	ubuntu	"bash"	3 hours ago
Exited (0) 27 minutes ago		admiring_goodall	
d0599c9f5a0f	hello-world	"/hello"	4 hours ago
Exited (0) 4 hours ago		gifted_mahavira	

```
bash-4.3# docker network connect vlan20-192.168.20.0 f205bfb1df04
```

```
bash-4.3# docker start f205bfb1df04
```

```
f205bfb1df04
```

```
bash-4.3# docker ps -a
```

CONTAINER ID	IMAGE	COMMAND	CREATED	
f205bfb1df04	ubuntu	"bash"	4 hours ago	Up
7 seconds		admiring_goodall		
d0599c9f5a0f	hello-world	"/hello"	4 hours ago	
Exited (0) 4 hours ago		gifted_mahavira		

```
bash-4.3# docker inspect f205bfb1df04
```

```

[
  {
    "Id": "f205bfb1df04740a0db1f999a73c680b4d47d397b00b4d7d0bd7be000ab0637",
    "Created": "2019-11-07T10:54:28.359813484Z",
    "Path": "bash",

```

```

"Args": [],
"State": {
  "Status": "running",
  "Running": true,

  "Paused": false,
  "Restarting": false,
...
  "NetworkSettings": {
    "Bridge": "",
    "SandboxID":
"7933d284044484ea162b3ae61c137aac80920772b0afb948e649d903897ae8c1",
    "HairpinMode": false,
    "LinkLocalIPv6Address": "",
    "LinkLocalIPv6PrefixLen": 0,
    "Ports": {},
    "SandboxKey": "/var/run/docker/netns/7933d2840444",
    "SecondaryIPAddresses": null,
    "SecondaryIPv6Addresses": null,
    "EndpointID":
"9b43ec82f7c7a848161dac123544cf339fda4197bc981173e1a1da1bf0cb230e",
    "Gateway": "172.17.0.1",
    "GlobalIPv6Address": "",
    "GlobalIPv6PrefixLen": 0,
    "IPAddress": "172.17.0.2",
    "IPPrefixLen": 16,
    "IPv6Gateway": "",
    "MacAddress": "02:42:ac:11:00:02",
    "Networks": {
      "bridge": {
        "IPAMConfig": null,
        "Links": null,
        "Aliases": null,
        "NetworkID":
"1bcf3ad2e9fab255fc47bcc98f1ab8712bfc34cb02f9b9816dcaae6f2485682f",
        "EndpointID":
"9b43ec82f7c7a848161dac123544cf339fda4197bc981173e1a1da1bf0cb230e",
        "Gateway": "172.17.0.1",
        "IPAddress": "172.17.0.2",
        "IPPrefixLen": 16,
        "IPv6Gateway": "",
        "GlobalIPv6Address": "",
        "GlobalIPv6PrefixLen": 0,
        "MacAddress": "02:42:ac:11:00:02",
        "DriverOpts": null
      },
      "vlan20-192.168.20.0": {
        "IPAMConfig": {},
        "Links": null,
        "Aliases": [
          "f205bfb1df04"
        ],
        "NetworkID":
"4a56867b974d11d1af5c710dd804a7b1f484d04e0eadf294df1261a4ce63a83f",
        "EndpointID":
"5307896f99284b40c7b8090c34cacb58429719fe6ba20d218508beb55c548fd4",
        "Gateway": "192.168.20.10",
        "IPAddress": "192.168.20.128",

```

```

        "IPPrefixLen": 24,
        "IPv6Gateway": "",
        "GlobalIPv6Address": "",
        "GlobalIPv6PrefixLen": 0,
        "MacAddress": "02:42:c0:a8:14:80",
        "DriverOpts": null
    }
}
}
]
bash-4.3# docker exec -it f205bfb1df04 /bin/bash
root@f205bfb1df04:/# ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 172.17.0.2 netmask 255.255.0.0 broadcast 0.0.0.0
    ether 02:42:ac:11:00:02 txqueuelen 0 (Ethernet)
    RX packets 0 bytes 0 (0.0 B)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 0 bytes 0 (0.0 B)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

eth1: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 192.168.20.128 netmask 255.255.255.0 broadcast 0.0.0.0
    ether 02:42:c0:a8:14:80 txqueuelen 0 (Ethernet)
    RX packets 2 bytes 128 (128.0 B)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 1 bytes 42 (42.0 B)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
    inet 127.0.0.1 netmask 255.0.0.0
    loop txqueuelen 1 (Local Loopback)
    RX packets 0 bytes 0 (0.0 B)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 0 bytes 0 (0.0 B)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
root@f205bfb1df04:/# exit
exit

```

Creating a new Ubuntu Container with the attached Macvlan Interface on Docker via Bash

```
bash-4.3# docker run --net=vlan20-192.168.20.0 -dit ubuntu
e94968abf3e1b0674e4847d12be40b5b31fb6b5ee405f538e86fdc05f146ed3a
bash-4.3# docker ps -a
CONTAINER ID        IMAGE               COMMAND             CREATED
STATUS            PORTS              NAMES
e94968abf3e1      ubuntu            "/bin/bash"        7 seconds ago    Up
6 seconds         gifted_morse
f205bfb1df04      ubuntu            "bash"             4 hours ago      Up
About an hour     admiring_goodall
d0599c9f5a0f      hello-world       "/hello"           5 hours ago
Exited (0) 5 hours ago    gifted_mahavira
bash-4.3# docker inspect e94968abf3e1
...
    "Networks": {
      "vlan20-192.168.20.0": {
        "IPAMConfig": null,
        "Links": null,
        "Aliases": [
          "e94968abf3e1"
        ],
        "NetworkID":
"4a56867b974d11d1af5c710dd804a7b1f484d04e0eadf294df1261a4ce63a83f",
        "EndpointID":
"ea6ee207f00957821eb7d8a2f4d08560257c444ac29f20a38f9b11bb20e25e",
        "Gateway": "192.168.20.10",
        "IPAddress": "192.168.20.129",
        "IPPrefixLen": 24,
        "IPv6Gateway": "",
        "GlobalIPv6Address": "",
        "GlobalIPv6PrefixLen": 0,
        "MacAddress": "02:42:c0:a8:14:81",
        "DriverOpts": null
      }
    }
  }
}
]
bash-4.3#
bash-4.3# docker exec -it f205bfb1df04 /bin/bash
root@f205bfb1df04:/# ping 192.168.20.129
PING 192.168.20.129 (192.168.20.129) 56(84) bytes of data.
 64 bytes from 192.168.20.129: icmp_seq=1 ttl=64 time=0.490 ms
 64 bytes from 192.168.20.129: icmp_seq=2 ttl=64 time=0.095 ms
 64 bytes from 192.168.20.129: icmp_seq=3 ttl=64 time=0.163 ms
 64 bytes from 192.168.20.129: icmp_seq=4 ttl=64 time=0.081 ms
 64 bytes from 192.168.20.129: icmp_seq=5 ttl=64 time=0.096 ms
 64 bytes from 192.168.20.129: icmp_seq=6 ttl=64 time=0.094 ms
 64 bytes from 192.168.20.129: icmp_seq=7 ttl=64 time=0.095 ms
^C
--- 192.168.20.129 ping statistics ---
 7 packets transmitted, 7 received, 0% packet loss, time 6143ms
 rtt min/avg/max/mdev = 0.081/0.159/0.490/0.137 ms
```

Installing Ubuntu in a Docker Container via EOS CLI

Instead of using Linux Bash EOS also supports CLI commands related to Docker.

Please make sure that before starting “**container-manager**” is configured on the system to start the Docker Daemon!

EOS config:

container-manager

```
!  
ip name-server vrf default 8.8.8.8  
!  
interface Management1  
    ip address 192.168.10.71/24  
!  
ip route 0.0.0.0/0 192.168.10.1
```

Downloading an Ubuntu Image via EOS CLI

```
Router-1#container-manager pull ubuntu  
Using default tag: latest  
latest: Pulling from library/ubuntu  
9273eca053fc: Pull complete  
32bb043e9224: Pull complete  
a9526b45dc3f: Pull complete  
694d3060ad92: Pull complete  
Digest: sha256:6e9f67fa63b0323e9a1e587fd71c561ba48a034504fb804fd26fd8800039835d  
Status: Downloaded newer image for ubuntu:latest
```

```
Router-1#show container-manager images
```

Name	Tag	Id	Created	Size
-----	-----	-----	-----	-----
ubuntu	latest	a7931a69babc	6 days ago	61.82 MB

```
Router-1#conf terminal
```

```
Router-1(config)#container-manager
```

```
Router-1(config-container-mgr)#container Ubuntu
```

```
Router-1(config-container-mgr-container-Ubuntu)#image ubuntu
```

```
Router-1(config-container-mgr-container-Ubuntu)#end
```

Creating a new Ubuntu Container via EOS CLI

```
Router-1#container-manager start Ubuntu
```

```
Ubuntu
```

```
Router-1#show container-manager containers
```

```
Container Name: Ubuntu
```

```
Container Id: ca76f9738ccadb2d0704334f4e41392ff6f9f6e4fed9cf431dad5ba4d1ac0a39
```

```
Image Name: ubuntu
```

```
Image Id: sha256:a7931a69babcdf74b0347c2a2224995f7d2dc1beca13f64ac637ce9b9635c4cc
```

```
Onboot: False
```

```
Command: /bin/bash
```

```
Created: 1 minutes ago
```

```
Ports:
```

```
State: exited
```

Creating a Docker Container backup via EOS CLI

It is important to know that after a reboot all Docker Containers on the system won't be present anymore! Therefore, Docker Containers can be backed up to stay persistent during a reboot.

```
Router-1#container-manager backup container Ubuntu
Container Ubuntu has been committed. Backing up created
sha256:b43b12b9bb59d3780538ea665d6a9001ecf9813251b5e1f5f7b896987c8313ee image at
/mnt/flash/.containermgr/Ubuntu.tar
Router-1#show container-manager backup
  Files                Directory
-----
  Ubuntu                /mnt/flash/
```

Deleting a backed up Docker Container via EOS CLI

```
Router-1#container-manager backup remove Ubuntu
Router-1#show container-manager backup
  Files                Directory
-----
Router-1#
```

KVM based Virtualization on EOS

Arista's EOS has leveraged its unmodified Linux kernel, and brought operating system virtualization to Ethernet switching, utilizing the kernel-based virtual machine (KVM) as follows:

- The hypervisor is the Linux kernel.
- The core virtualization infrastructure is provided by the kernel module.
- The CPU-specific implementation is provided by the processor-specific module (Intel or AMD).
- The generic machine emulator and virtualizer KVM is provided by a Modified Quick Emulator (QEMU), which transforms the Linux kernel into the hypervisor.

The standard Linux kernel is the hypervisor, resulting in changes to the standard kernel (such as memory support and scheduler). Optimizations to these Linux components benefit both the hypervisor (host operating system) and Linux guest operating systems.

With the kernel acting as the hypervisor, the switch can run other operating systems, such as Windows or Linux.

EOS supports VM **VMDK** files where this extension is associated with the Virtual Machine Disk (**VMDK**) format specification used with VMware virtual machine files.

A good Source for ready to use VMDK Images is e.g. **gns3**.



In order to find and download e.g. an Ubuntu Server VMDK image you may use the link

<https://docs.gns3.com/appliances/ubuntu-server-18.html>.

Virtual machines are not supported on vEOS-lab systems!

Copying VM Image to the system

Very similar to a Software transfer to the box you can simply upload a vmdk file via FTP
Here, I used **Tiny Core Linux** which is a very small Linux Distribution for tests.

```
Router-1#copy ftp://mkrausch:getaccess@192.168.10.206/Tiny_Linux_VM.vmdk
flash:/Tiny_Linux_VM.vmdk
Copy completed successfully.
Router-1#dir
Directory of flash:/
```

```
-rw-          3927      Nov 11 08:08  AsuFastPktTransmit.log
drwx          4096      Oct 4 10:56  Fossil
-rw-          2414      Nov 11 08:08  SsuRestore.log
-rw-          2414      Nov 11 08:08  SsuRestoreLegacy.log
-rw-       12251136      Nov 11 10:35  Tiny_Linux_VM.vmdk
-rw-           24      Sep 26 22:56  boot-config
drwx          4096      Oct 14 11:05  debug
drwx          4096      Oct 4 10:56  fastpkttx.backup
drwx       16384      Sep 26 22:56  lost+found
drwx          4096      Nov 11 10:54  persist
drwx          4096      Oct 4 11:01  schedule
-rw-          1270      Nov 7 08:41  startup-config
-rw-       418919380     Sep 26 22:56  vEOS-lab.swi
-rw-           13      Oct 4 12:37  zerotouch-config
```

```
4093313024 bytes total (3214979072 bytes free)
```

Creating a VM on EOS

```
Router-1(config)#virtual-machine Test-1
Router-1(config-vm-Test-1)#disk-image flash: Tiny_Linux_VM.vmdk
Router-1(config-vm-Test-1)#memory-size 2048
Router-1(config-vm-Test-1)#virtual.nic 1 Management1
Router-1(config-vm-Test-1)#virtual.nic 2 vlan 10
Router-1(config-vm-Test-1)#enable
```

Connecting to the Virtual Machine:

```
Router-1#virtual-machine Test-1 console
```

Troubleshooting

Tech-Support Files

The Arista System will create Tech-Support Files once per hour automatically. They can be downloaded from the folder `/mnt/flash/schedule/tech-support`

Root

```
DC1-LEAF1A#dir /schedule/tech-support
```

```
Directory of flash:/schedule/tech-support
```

```

-rw-      104166      Apr 24 21:47
DC1-LEAF1A_tech-support_2021-04-24.2145.log.gz
-rw-      104152      Apr 24 22:47
DC1-LEAF1A_tech-support_2021-04-24.2245.log.gz
-rw-      104236      Apr 24 23:47
DC1-LEAF1A_tech-support_2021-04-24.2345.log.gz
-rw-      104275      Apr 25 00:47
DC1-LEAF1A_tech-support_2021-04-25.0045.log.gz
-rw-      104414      Apr 25 01:47
DC1-LEAF1A_tech-support_2021-04-25.0145.log.gz
-rw-      104479      Apr 25 02:47
DC1-LEAF1A_tech-support_2021-04-25.0245.log.gz
-rw-      104460      Apr 25 03:47
DC1-LEAF1A_tech-support_2021-04-25.0345.log.gz
-rw-      104554      Apr 25 04:47
DC1-LEAF1A_tech-support_2021-04-25.0445.log.gz
-rw-      104598      Apr 25 05:47
DC1-LEAF1A_tech-support_2021-04-25.0545.log.gz
...
-rw-       85202      Apr 29 13:06
DC1-LEAF1A_tech-support_2021-04-29.1305.log.gz
-rw-       85154      Apr 29 14:06
DC1-LEAF1A_tech-support_2021-04-29.1405.log.gz

4093313024 bytes total (2723954688 bytes free)
```

TCPDUMP Control Packet Sniffing

You will only see packets sourced from or destined to the EOS system if you use tcpdump on a front-panel interface!

Therefore, you can run port mirroring on an interface and send a copy of the dataplane traffic to the CPU where the tcpdump utility can be used.

Displaying TCPDUMP Output:

```
Switch-1#tcpdump interface ethernet 3 filter ether dst host 01:80:c2:00:00:0e
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on et3, link-type EN10MB (Ethernet), capture size 262144 bytes
11:34:00.362608 0c:7e:d1:c5:6e:03 > 01:80:c2:00:00:0e, ethertype LLDP (0x88cc),
length 100: LLDP, length 86: Switch-1
11:34:00.679757 0c:7e:d1:aa:4a:03 > 01:80:c2:00:00:0e, ethertype LLDP (0x88cc),
length 96: LLDP, length 82: Leaf-3
<Control-C>
```

2 packets captured
2 packets received by filter
0 packets dropped by kernel

Switch-1#tcpdump interface ethernet 3 verbose filter ether dst host
01:80:c2:00:00:0e

tcpdump: listening on et3, link-type EN10MB (Ethernet), capture size 262144 bytes
11:35:30.367146 0c:7e:d1:c5:6e:03 > 01:80:c2:00:00:0e, ethertype LLDP (0x88cc),
length 100: LLDP, length 86

Chassis ID TLV (1), length 7

Subtype MAC address (4): 0c:7e:d1:fb:b5:24

0x0000: 040c 7ed1 fbb5 24

Port ID TLV (2), length 10

Subtype Interface Name (5): Ethernet3

0x0000: 0545 7468 6572 6e65 7433

Time to Live TLV (3), length 2: TTL 120s

0x0000: 0078

System Name TLV (5), length 8: Switch-1

0x0000: 5377 6974 6368 2d31

System Capabilities TLV (7), length 4

System Capabilities [Bridge, Router] (0x0014)

Enabled Capabilities [Bridge] (0x0004)

0x0000: 0014 0004

Management Address TLV (8), length 14

Management Address length 7, AFI 802 (6): 0c:7e:d1:fb:b5:24

Unknown Interface Numbering (1): 0

0x0000: 0706 0c7e d1fb b524 0100 0000 0000

Organization specific TLV (127), length 6: OUI Ethernet bridged (0x0080c2)

Port VLAN Id Subtype (1)

port vlan id (PVID): 1

0x0000: 0080 c201 0001

Organization specific TLV (127), length 9: OUI IEEE 802.3 Private

(0x00120f)

Link aggregation Subtype (3)

aggregation status [supported, enabled], aggregation port ID 1000010

0x0000: 0012 0f03 0300 0f42 4a

Organization specific TLV (127), length 6: OUI IEEE 802.3 Private

(0x00120f)

Max frame size Subtype (4)

MTU size 9236

0x0000: 0012 0f04 2414

End TLV (0), length 0

11:35:30.684454 0c:7e:d1:aa:4a:03 > 01:80:c2:00:00:0e, ethertype LLDP (0x88cc),
length 96: LLDP, length 82

Chassis ID TLV (1), length 7

Subtype MAC address (4): 0c:7e:d1:4e:b7:ac

0x0000: 040c 7ed1 4eb7 ac

Port ID TLV (2), length 10

Subtype Interface Name (5): Ethernet3

0x0000: 0545 7468 6572 6e65 7433

Time to Live TLV (3), length 2: TTL 120s

0x0000: 0078

System Name TLV (5), length 6: Leaf-3

0x0000: 4c65 6166 2d33

System Capabilities TLV (7), length 4

System Capabilities [Bridge, Router] (0x0014)

Enabled Capabilities [Bridge, Router] (0x0014)

0x0000: 0014 0014

```

Management Address TLV (8), length 12
  Management Address length 5, AFI IPv4 (1): 1.1.1.103
  Interface Index Interface Numbering (2): 5000000
  0x0000: 0501 0101 0167 0200 4c4b 4000
Organization specific TLV (127), length 6: OUI Ethernet bridged (0x0080c2)
  Port VLAN Id Subtype (1)
    port vlan id (PVID): 1
    0x0000: 0080 c201 0001
Organization specific TLV (127), length 9: OUI IEEE 802.3 Private
(0x00120f)
  Link aggregation Subtype (3)
    aggregation status [supported, enabled], aggregation port ID 1000001
    0x0000: 0012 0f03 0300 0f42 41
Organization specific TLV (127), length 6: OUI IEEE 802.3 Private
(0x00120f)
  Max frame size Subtype (4)
    MTU size 9236
    0x0000: 0012 0f04 2414
End TLV (0), length 0
<Control-C>
2 packets captured
2 packets received by filter
0 packets dropped by kernel

```

Switch-1#tcpdump interface ethernet 1 filter src host 1.1.1.1

```

tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on et1, link-type EN10MB (Ethernet), capture size 262144 bytes
14:10:27.647654 50:4e:00:97:55:3a > 50:0f:00:f4:95:e7, ethertype IPv4 (0x0800),
length 66: 1.1.1.1.63748 > 1.1.1.3.bfd-multi-ctl: BFDv1, Multihop, State Up,
Flags: [none], length: 24
14:10:28.769318 50:4e:00:97:55:3a > 50:0f:00:f4:95:e7, ethertype IPv4 (0x0800),
length 66: 1.1.1.1.63748 > 1.1.1.3.bfd-multi-ctl: BFDv1, Multihop, State Up,
Flags: [none], length: 24
14:10:29.890469 50:4e:00:97:55:3a > 50:0f:00:f4:95:e7, ethertype IPv4 (0x0800),
length 66: 1.1.1.1.63748 > 1.1.1.3.bfd-multi-ctl: BFDv1, Multihop, State Up,
Flags: [none], length: 24
14:10:31.070973 50:4e:00:97:55:3a > 50:0f:00:f4:95:e7, ethertype IPv4 (0x0800),
length 66: 1.1.1.1.63748 > 1.1.1.3.bfd-multi-ctl: BFDv1, Multihop, State Up,
Flags: [none], length: 24
14:10:32.065517 50:4e:00:97:55:3a > 50:0f:00:f4:95:e7, ethertype IPv4 (0x0800),
length 66: 1.1.1.1.63748 > 1.1.1.3.bfd-multi-ctl: BFDv1, Multihop, State Up,
Flags: [none], length: 24
14:10:33.218282 50:4e:00:97:55:3a > 50:0f:00:f4:95:e7, ethertype IPv4 (0x0800),
length 66: 1.1.1.1.63748 > 1.1.1.3.bfd-multi-ctl: BFDv1, Multihop, State Up,
Flags: [none], length: 24
14:10:33.411239 50:4e:00:97:55:3a > 50:0f:00:f4:95:e7, ethertype IPv4 (0x0800),
length 178: 1.1.1.1.bgp > 1.1.1.3.42829: Flags [P.], seq 1454174081:1454174185,
ack 2468345303, win 128, options [nop,nop,md5 shared secret not supplied with -M,
can't check - 9b9885a4879d0899c230478fc62c981d], length 104: BGP
14:10:33.465793 50:4e:00:97:55:3a > 50:0f:00:f4:95:e7, ethertype IPv4 (0x0800),
length 143: 1.1.1.1.bgp > 1.1.1.3.42829: Flags [P.], seq 104:173, ack 1, win 128,
options [nop,nop,md5 shared secret not supplied with -M, can't check -
4450d22f7546062217bbe3d55a84904f], length 69: BGP
14:10:33.508861 50:4e:00:97:55:3a > 50:0f:00:f4:95:e7, ethertype IPv4 (0x0800),
length 74: 1.1.1.1.bgp > 1.1.1.3.42829: Flags [.], ack 135, win 127, options
[nop,nop,md5 shared secret not supplied with -M, can't check -
d9dc55c2dce7c43594737e17eb195d89], length 0

```

```

14:10:33.531111 50:4e:00:97:55:3a > 50:0f:00:f4:95:e7, ethertype IPv4 (0x0800),
length 74: 1.1.1.1.bgp > 1.1.1.3.42829: Flags [.], ack 204, win 127, options
[nop,nop,md5 shared secret not supplied with -M, can't check -
4eb1627c10e5bd6b1cdf44ad3597b0ae], length 0
14:10:33.876867 50:4e:00:97:55:3a > 50:0f:00:f4:95:e7, ethertype IPv4 (0x0800),
length 303: 1.1.1.1.bgp > 1.1.1.3.42829: Flags [P.], seq 173:402, ack 204, win
128, options [nop,nop,md5 shared secret not supplied with -M, can't check -
e9a541ab8aaa956b94240e4deb748c4f], length 229: BGP
14:10:33.983064 50:4e:00:97:55:3a > 50:0f:00:f4:95:e7, ethertype IPv4 (0x0800),
length 204: 1.1.1.1.bgp > 1.1.1.3.42829: Flags [P.], seq 402:532, ack 204, win
128, options [nop,nop,md5 shared secret not supplied with -M, can't check -
e9cad0fcd4db3e0a60a3252bd2de8f89], length 130: BGP
14:10:33.998555 50:4e:00:97:55:3a > 50:0f:00:f4:95:e7, ethertype IPv4 (0x0800),
length 74: 1.1.1.1.bgp > 1.1.1.3.42829: Flags [.], ack 338, win 127, options
[nop,nop,md5 shared secret not supplied with -M, can't check -
90c93abade7fba602653edb22606c934], length 0
14:10:34.400592 50:4e:00:97:55:3a > 50:0f:00:f4:95:e7, ethertype IPv4 (0x0800),
length 66: 1.1.1.1.63748 > 1.1.1.3.bfd-multi-ctl: BFDv1, Multihop, State Up,
Flags: [none], length: 24
^C
14 packets captured
14 packets received by filter
0 packets dropped by kernel

```

Sending a packet trace directly to a PCAP File:

This creates a **PCAP file** which is not a plain-text file anymore!

```

Switch-1#tcpdump interface ethernet 12 verbose file flash:eth1.capture
max-file-size 2
tcpdump: listening on et12, link-type EN10MB (Ethernet), capture size 262144 bytes
^C28 packets captured
28 packets received by filter
0 packets dropped by kernel
Switch-1#dir
Directory of flash:/

-rw-          5082          Nov 13 08:09  AsuFastPktTransmit.log
-rw-          1401          Nov 12 15:16  EVPN-L2
drwx          4096          Oct 4 10:56   Fossil
-rw-          3124          Nov 13 08:08  SsuRestore.log
-rw-          3124          Nov 13 08:08  SsuRestoreLegacy.log
-rw-           24          Sep 26 22:56  boot-config
drwx          4096          Oct 14 08:34  debug
-rw-          3786          Nov 13 11:47  eth1.capture
drwx          4096          Oct 4 10:56   fastpkttx.backup
drwx         16384          Sep 26 22:56  lost+found
drwx          4096          Nov 13 11:45  persist
drwx          4096          Oct 4 11:01   schedule
-rw-          1499          Nov 12 15:21  startup-config
-rw-         418919380       Sep 26 22:56  vEOS-lab.swi
-rw-           13          Oct 4 12:37   zerotouch-config

4093313024 bytes total (3226120192 bytes free)

```

TCPDUMP Data Packet Sniffing

This feature is not supported on vEOS-Lab Images and not on all Hardware platforms!

As tcpdump per default only can trace packets sourced from or destined to the EOS system you can run port mirroring on an interface and send a copy of the **dataplane** traffic to the **CPU** where the tcpdump utility can be used.

```
Switch-1(config)#monitor session test1 source ethernet 12 rx
Switch-1(config)#monitor session test1 destination cpu
Switch-1(config)#exit
Switch-1#show monitor session test1
```

```
Session test1
```

```
-----
```

```
Source Ports
```

```
  Rx Only:      Et12
```

```
Destination Ports
```

```
  Cpu:   active (mirror0)
```

```
Switch-1#bash
```

```
Arista Networks EOS shell
```

```
[admin@Switch-1 ~]$ tcpdump -i mirror0
```

Sending TCPDUMP outputs directly to a remote Wireshark station

It is possible to send TCPDUMP outputs from an EOS based system directly to a remote Wireshark station.

On the EOS device you will have to configure a **root password**:

```
Arista-1#conf t
Arista-1(config)#aaa root secret arista
Arista-1(config)#exit
```

You can download Wireshark at <https://www.wireshark.org/#download> and install it on your device.

In this example Wireshark has been installed on a MacOS based system - on other OS platforms it is installed at a different location or could be started directly.

On MacOS you first need to change the correct directory before starting the remote capture.

```
mkrausch@mkrausch ~ % cd /Applications/Wireshark.app/Contents/MacOS
mkrausch@mkrausch MacOS % ls
Wireshark          editcap.dSYM      rawshark.dSYM
Wireshark.dSYM    extcap           reordercap
capinfos          idl2wrs          reordercap.dSYM
capinfos.dSYM    idl2wrs.dSYM     sharkd
capytype         mergecap         sharkd.dSYM
capytype.dSYM    mergecap.dSYM   text2pcap
dfptest         mmdbresolve     text2pcap.dSYM
```

```

dfctest.dSYM      mddbresolve.dSYM  tshark
dumpcap           randpkt           tshark.dSYM
dumpcap.dSYM     randpkt.dSYM
editcap          rawshark

```

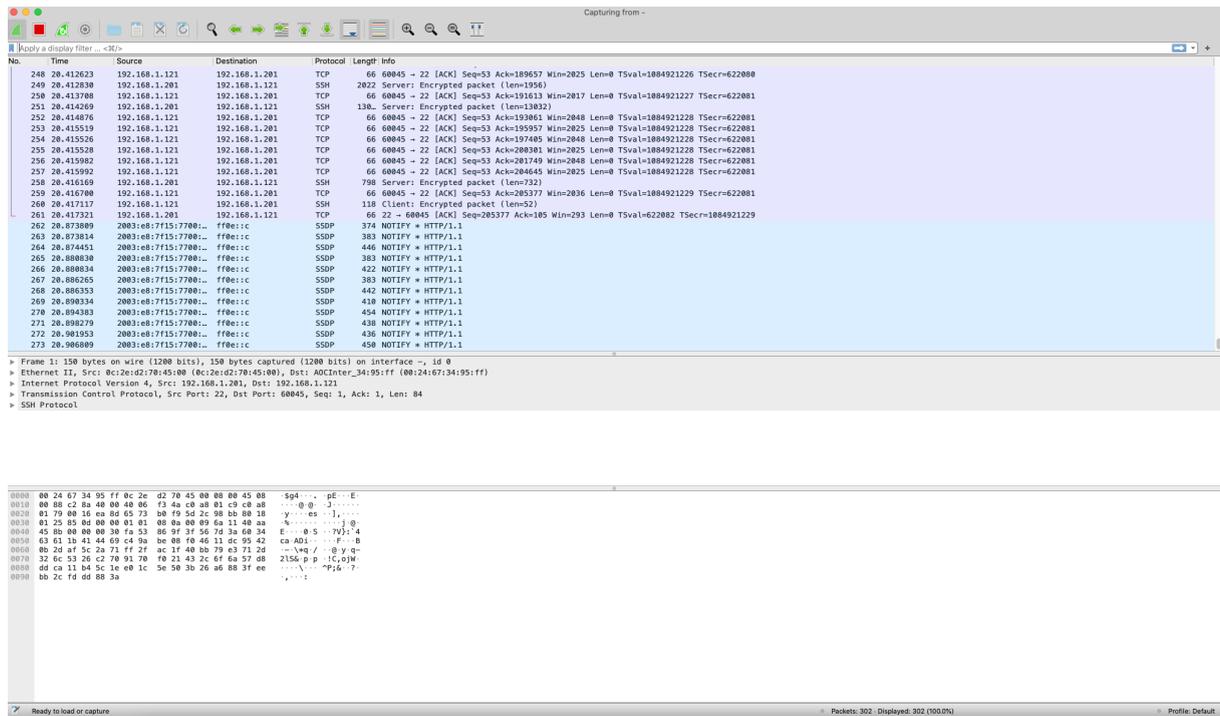
To start the remote capture just type in the following command on your MacOS system using the IP address of the EOS device and the port which you want to capture (e.g. ma1 = Management 1, et1 = Ethernet 1, mirror0 etc.):

```

mkrausch@mkrausch MacOS % ssh root@192.168.1.201 "tcpdump -s0 -U -w - -i ma1" |
./Wireshark -k -i -
The authenticity of host '192.168.1.201 (192.168.1.201)' can't be established.
ECDSA key fingerprint is SHA256:6W+fo59ya0UJ336wV0vHzLz10ILqfbhj9LsnaPQfXn1E.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '192.168.1.201' (ECDSA) to the list of known hosts.
Password: arista
tcpdump: listening on ma1, link-type EN10MB (Ethernet), capture size 262144 bytes

```

Wireshark will be started automatically and the capture will be displayed in real time:



To stop the capture just type "Control + c"

Santa Clara—Corporate Headquarters
5453 Great America Parkway,
Santa Clara, CA 95054
Phone: +1-408-547-5500
Fax: +1-408-538-8920
Email: info@arista.com

Ireland—International Headquarters
3130 Atlantic Avenue
Westpark Business Campus
Shannon, Co. Clare
Ireland
Vancouver—R&D Office
9200 Glenlyon Pkwy, Unit 300
Burnaby, British Columbia
Canada V5J 5J8
San Francisco—R&D and Sales Office
1390
Market Street, Suite 800
San Francisco, CA 94102

India—R&D Office
Global Tech Park, Tower A & B, 11th
Floor
Marathahalli Outer Ring Road
Devarabeesanahalli Village, Varthur
Hobli
Bangalore, India 560103
Singapore—APAC Administrative Office
9 Temasek Boulevard
#29-01, Suntec Tower Two
Singapore 038989
Nashua—R&D Office
10 Tara Boulevard
Nashua, NH 03062

Copyright © 2022 Arista Networks, Inc. All rights reserved. CloudVision, and EOS are registered trademarks and Arista Networks is a trademark of Arista Networks, Inc. All other company names are trademarks of their respective holders. Information in this document is subject to change without notice. Certain features may not yet be available. Arista Networks, Inc. assumes no responsibility for any errors that may appear in this document. May, 2022.

www.arista.com