

CloudVision Mastery Workshop Lab Guide



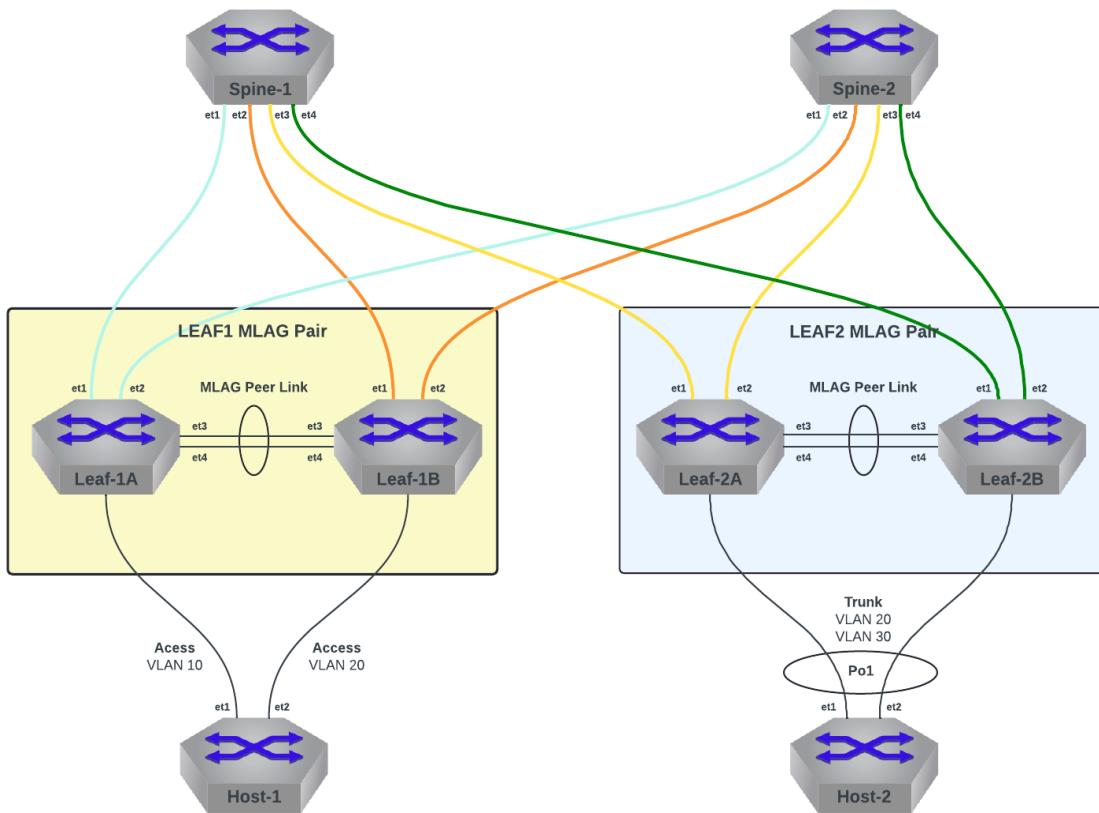
Table of Contents

Lab Topology	2
Connecting to your CVaaS Instance	3
Creating Static Configlets	8
Assigning Configlets to Containers	13
Provisioning Switches	16
Tasks & Change Controls	23
Compliance Updates and Token	29
Images and Image Bundles	30
Image Assignment	35
Create an Action Bundle for the Leafs	39
Create an Action Bundle for the Spines	43
Create an Action Bundle to Cleanup the Flash	47
Create a Template	49
Compliance - Device Configuration	66
Create an Action	72
Creating Custom Dashboards	84

Events and Notifications	89
CloudVision Event Generation	93
CloudVision Event Notifications	95

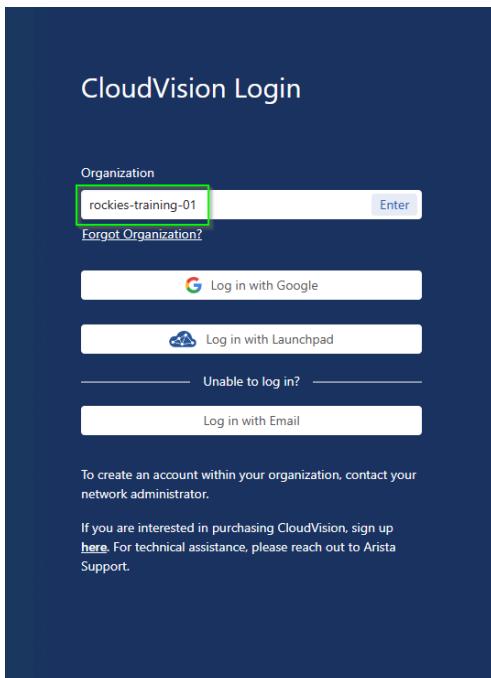
Lab Topology

Each lab environment consists of a Layer 3 Leaf & Spine design. There are 2 Spine switches and 4 Leaf switches. The Leaf switches are connected in MLAG pairs. There is a single host device connected to each MLAG pair. Host-1 is simulating a single-homed device, connected on Leaf-1A and Leaf-1B on different access VLAN's. Host-2 is acting as a dual-homed server using an LACP Port-Channel to allow for active/active sharing of workloads.

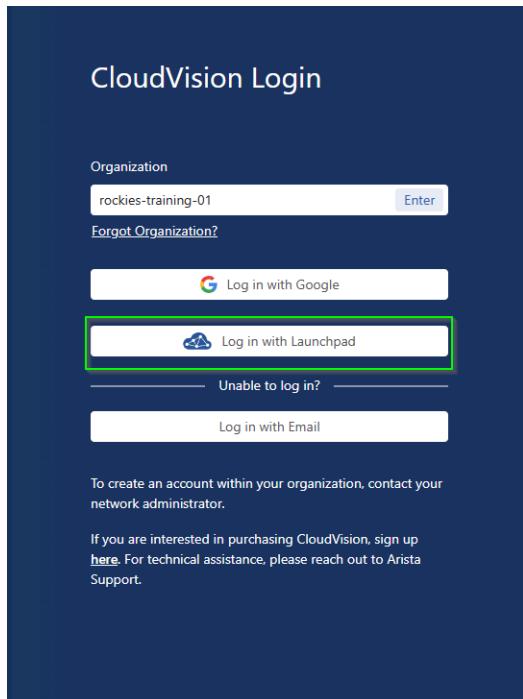


Connecting to your CVaaS Instance

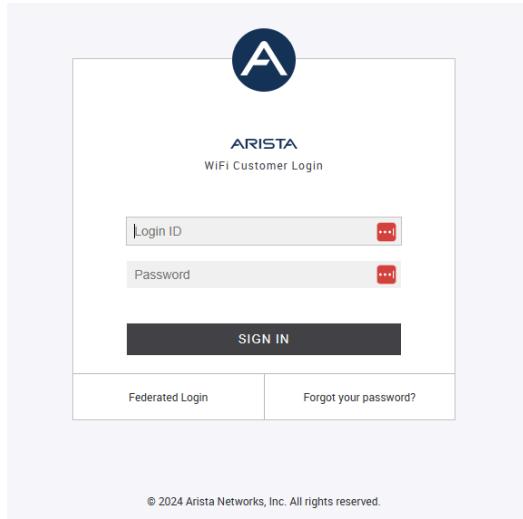
1. Open a browser (preferably Chrome or Firefox) and go to <https://www.arista.io/>.
2. In the “Organization” box, enter your assigned CVaaS instance name. For example, **rockies-training-01**.



3. Click “Log in with Launchpad”



4. Enter in your assigned email address and password



5. Welcome to CloudVision!

Device ID	MAC Address	IP Address	Streaming Agent	Model	Software	Issues	Streaming	Device
44b5:1ea1:01:71		192.168.0.14	vEOS-lab	4.30.6M	1.28.3		● Active	sw-10.18.159.104
44f4:c0:05:b9:b8		192.168.0.18	vEOS-lab	4.30.6M	1.28.3		● Active	sw-10.18.160.13
441b:01:42:21:a8		192.168.0.17	vEOS-lab	4.30.6M	1.28.3		● Active	sw-10.18.160.112
4490:6ccb:b6:5b		192.168.0.12	vEOS-lab	4.30.6M	1.28.3		● Active	sw-10.18.167.203
44d9:35:8b:c3:a3		192.168.0.15	vEOS-lab	4.30.6M	1.28.3		● Active	sw-10.18.167.204
443d:62:1b:7c:2f		192.168.0.16	vEOS-lab	4.30.6M	1.28.3		● Active	sw-10.18.167.205
44ea:5cf5:c1:2d		192.168.0.13	vEOS-lab	4.30.6M	1.28.3		● Active	sw-10.18.167.206

Containers

Containers are a logical entity used to group network devices and to define a hierarchy to which configurations can be applied. A container can contain a device, a group of devices, or other containers.

In this lab we will be creating a Container structure for our virtual data center deployment.

1. Navigate to **Provisioning > Network Provisioning**
2. Right-click on the “**Tenant**” container and select **Add > Container**

The screenshot shows the Arista Network Provisioning interface. On the left is a sidebar with various navigation options like Provisioning, Configlets, Image Repository, Tasks, Actions, Change Control, Studios, Workspaces, Snapshot Configuration, Public Cloud Accounts, Tags, and Zero Touch Provisioning. The main area is titled 'Network Provisioning' and 'Assign devices to containers and manage device-specific configuration'. It shows a table with columns: Name, IP Address, Mac Address, Serial No., Container, and Status. A message at the top right says 'No data to be displayed'. Below the table, there's a 'Tenant' section with details: Associated Configs (0), Associated Switches (0), Created by (cvp system), and Created on (2024-04-02 18:10:14). At the bottom are 'Preview', 'Save', and 'Cancel' buttons. A context menu is open over the 'Container' column, with 'Add > Container' highlighted.

3. Name the new container “DC1” and click OK.
4. Now, right-click on the “DC1” container and select **Add > Container**
5. Name the new container “Spines”

This screenshot is identical to the one above, showing the Arista Network Provisioning interface with the 'Container' context menu open. The steps 4 and 5 described in the list below have not yet been executed, as the table still shows 'No data to be displayed'.

6. Repeat steps 4 & 5 and create a new container named “Leafs”
7. Right-click the “Leafs” container and create 2 new containers named “Leaf1” & “Leaf2” (see example below)

The screenshot shows the Arista Network Provisioning interface. On the left, a sidebar lists various options like Provisioning, Network Provisioning, Configlets, Image Repository, and Change Control. The main area is titled "Network Provisioning" with the sub-section "Leads". A search bar at the top of this section has the text "Search". Below it is a tree view showing a hierarchy: Tenant (0) > Undefined (0) > DC1 (0) > Leads (0). A modal window titled "New Container" is open, prompting for a "Container Name" which is currently "Leaf1". At the bottom of the screen are three buttons: "Preview", "Save", and "Cancel". In the top right corner, there is a user profile icon and the text "hardy Z_ROCKIES-ATD-01".

NOTE: You'll notice that all new containers are highlighted in green. This means there are changes that have yet to be saved.

- Click “**Save**” at the bottom of the screen. Our finished container structure should look something like this:

This screenshot shows the same Arista Network Provisioning interface after saving the container. The tree view now shows Tenant (0) > Undefined (0) > DC1 (0) > Leads (0), with "Leads (0)" expanded to show "Leaf1 (0)" and "Leaf2 (0)". To the right of the tree view, a detailed view of the "Leads" container is shown, including fields for "Associated Configlets" (empty), "Associated Switches" (empty), "Created by" (cv-system), and "Created on" (2024-04-02 18:10:14). The bottom of the screen features the "Save" button, which is now highlighted in green to indicate the changes have been saved.

Configlets

Configlets are a snippet of configuration that CloudVision users code and maintain independently. These configlets can be later applied to devices or containers in the topology. When you apply a configlet to a container, the configlet is automatically applied to all of the devices in the container, including devices in sub-containers. Multiple configlets can be applied to a single device or container. The collection of applied configlets create the “Designed Configuration” on the devices.

Creating Static Configlets

In this lab there will be several pre-built configlets and several that need to be built. In the steps below, we will be creating several new static configlets. We will then assign all configlets to the appropriate containers and devices.

A note on configlet naming convention. Based on our experience helping with many large customer deployments, we are using a naming convention that we found works very well. All configlets that are assigned directly to containers will be in all-caps (i.e. **SPINES**, **LEAF1**, **LEAF2**, etc) and any configlet that is assigned directly to a device will be in all lower-case (i.e. **spine-1**, **spine-2**, **leaf-1A**, **leaf-1B**, etc).

NOTE: In this section we'll be copying code blocks that may span multiple pages. So please ensure you're copying the entire code block before pasting into your configlet. Also, you must use keyboard shortcuts (Ctrl-C, Ctrl-V, Command-C, Command-V) in order to copy/paste inside of CVP.

1. Navigate to **Provisioning > Configlets**. In the upper right-hand corner, click on the “+” icon  and select “**Configlets**”
2. Name the new configlet “**spine-2**” and add the following configuration and click “**Save**”

```
hostname Spine-2
!
interface Ethernet1
    description Leaf-1A_eth2
    no switchport
    ip address 10.242.1.1/30
!
interface Ethernet2
    description Leaf-1B_eth2
    no switchport
```

```

    ip address 10.242.1.5/30
!
interface Ethernet3
    description Leaf-2A_eth2
    no switchport
    ip address 10.242.2.1/30
!
interface Ethernet4
    description Leaf-2B_eth2
    no switchport
    ip address 10.242.2.5/30
!
interface Loopback0
    ip address 10.250.0.2/32
!
interface Management1
    ip address dhcp
    dhcp client accept default-route
!
router bgp 65000
    router-id 10.250.0.2
    neighbor 10.242.1.2 peer group MLAG-1
    neighbor 10.242.1.6 peer group MLAG-1
    neighbor 10.242.2.2 peer group MLAG-2
    neighbor 10.242.2.6 peer group MLAG-2
!
```

3. Let's create another static configlet named “**LEAF2**” and add the following configuration and click “**Save**”

```

vlan 30,40
!
vlan 4093-4094
    trunk group MLAG_Peer_Link
!
interface Port-Channel1
    description Host-2
    switchport mode trunk
    mlag 1
!
interface Port-Channel2000
    description MLAG_Peer_Link
    switchport mode trunk
    switchport trunk group MLAG_Peer_Link
!
interface Ethernet3
    description MLAG-PEER-LINK
    channel-group 2000 mode active
!
interface Ethernet4
```

```

description MLAG-PEER-LINK
channel-group 2000 mode active
!
interface Vlan30
    ip virtual-router address 10.30.30.1
!
interface Vlan40
    ip virtual-router address 10.40.40.1
!
ip virtual-router mac-address 00:1c:73:00:00:01
!
mlag configuration
    domain-id MLAG
    peer-link Port-Channel2000
    reload-delay 60
!
router bgp 65002
    no bgp default ipv4-unicast
    maximum-paths 2
    neighbor MLAG-PEER peer group
    neighbor MLAG-PEER remote-as 65002
    neighbor SPINES peer group
    neighbor SPINES remote-as 65000
    redistribute connected
    !
    address-family ipv4
        neighbor MLAG-PEER activate
        neighbor SPINES activate
!

```

4. Now let's create configlets for each of our devices in the "Leaf2" container (**leaf-2a** & **leaf-2b**). Create a configlet named "**leaf-2a**" and add the following configuration

```

hostname Leaf-2A
!
interface Ethernet1
    description Spine-1_eth3
    no switchport
    ip address 10.241.2.2/30
!
interface Ethernet2
    description Spine-2_eth3
    no switchport
    ip address 10.242.2.2/30
!
interface Ethernet6
    description Host-2_eth1
    channel-group 1 mode active
!
interface Loopback0

```

```

    ip address 10.250.2.1/32
!
interface Management1
    ip address dhcp
    dhcp client accept default-route
!
interface Vlan30
    ip address 10.30.30.2/24
!
interface Vlan40
    ip address 10.40.40.2/24
!
interface Vlan4093
    description iBGP_Peering
    no autostate
    ip address 192.168.0.1/30
!
interface Vlan4094
    description MLAG_Peering
    no autostate
    ip address 192.168.1.1/30
!
mlag configuration
    local-interface Vlan4094
    peer-address 192.168.1.2
!
router bgp 65002
    router-id 10.250.2.1
    neighbor 10.241.2.1 peer group SPINES
    neighbor 10.242.2.1 peer group SPINES
    neighbor 192.168.0.2 peer group MLAG-PEER
!
```

5. Create a configlet named “**leaf-2b**” and add the following configuration

```

hostname Leaf-2B
!
interface Ethernet1
    description Spine-1_eth4
    no switchport
    ip address 10.241.2.6/30
!
interface Ethernet2
    description Spine-2_eth4
    no switchport
    ip address 10.242.2.6/30
!
interface Ethernet6
    description Host-2_eth2
    channel-group 1 mode active
!
```

```

interface Loopback0
    ip address 10.250.2.2/32
!
interface Management1
    ip address dhcp
    dhcp client accept default-route
!
interface Vlan30
    ip address 10.30.30.3/24
!
interface Vlan40
    ip address 10.40.40.3/24
!
interface Vlan4093
    description iBGP_Peering
    no autostate
    ip address 192.168.0.2/30
!
interface Vlan4094
    description MLAG_Peering
    no autostate
    ip address 192.168.1.2/30
!
mlag configuration
    local-interface Vlan4094
    peer-address 192.168.1.1
!
router bgp 65002
    router-id 10.250.2.2
    neighbor 10.241.2.5 peer group SPINES
    neighbor 10.242.2.5 peer group SPINES
    neighbor 192.168.0.1 peer group MLAG-PEER
!
```

6. We should now have the following list of configlets

Configlets

Manage configlets and view configlet details

Search

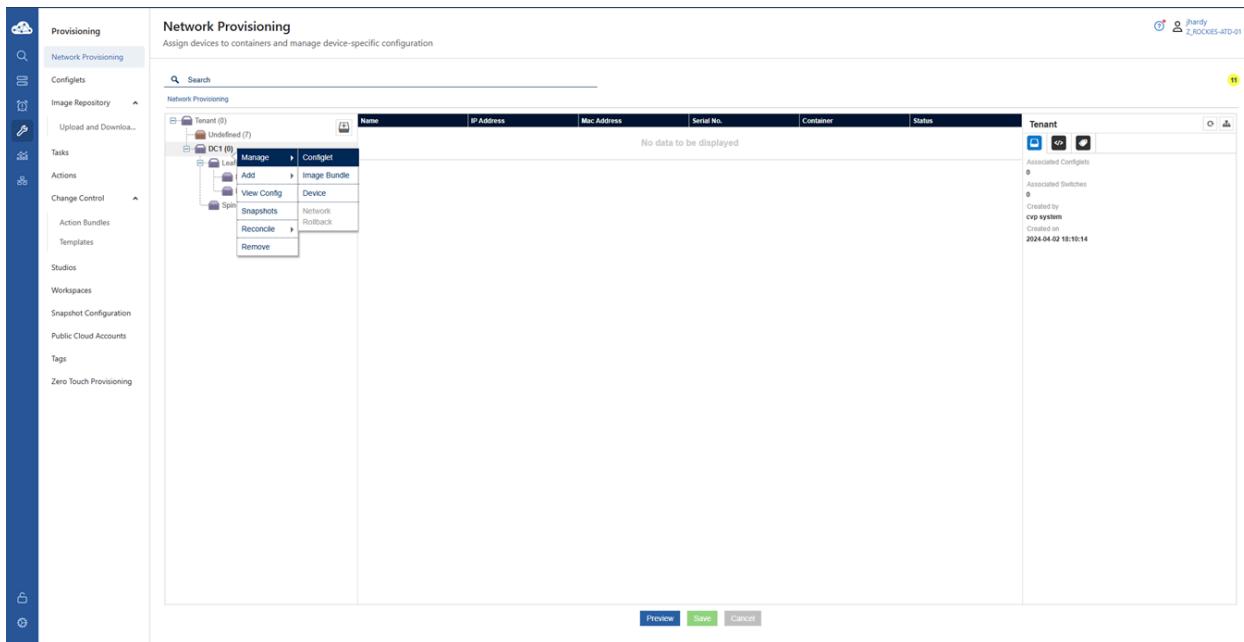
Name
<input type="checkbox"/> BASE
<input type="checkbox"/> LEAF1
<input type="checkbox"/> LEAF2
<input type="checkbox"/> SPINES
<input type="checkbox"/> leaf-1a
<input type="checkbox"/> leaf-1b
<input type="checkbox"/> leaf-2a
<input type="checkbox"/> leaf-2b
<input type="checkbox"/> spine-1
<input type="checkbox"/> spine-2

Assigning Configlets to Containers

Now that we have our container structure and our static configlets built, let's apply the configlets to our containers.

NOTE: Remember, any configlet that is assigned to a container will be inherited by the devices under that container.

1. Let's start by assigning the "**BASE**" configlet to the "**DC1**" container. Navigate to **Provisioning > Network Provisioning**. Right-click on the "**DC1**" container and select **Manage > Configlet**



2. Now select "**BASE**" from the list of configlets. You will see the "**BASE**" configlet show up on the right side of your screen under the "**Proposed Configuration**" section. You can see a preview of the configuration that will be applied by this configlet by clicking the "+" icon on the far right corner of the "**Proposed Configuration**" section for the "**BASE**" configlet. Click "**Update**" at the bottom of the screen.

The screenshot shows the Arista Network Provisioning interface. On the left, a sidebar lists various provisioning and management options like Configuration, Image Repository, Tasks, Actions, Change Control, and Zero Touch Provisioning. The main area is titled "Network Provisioning" and shows a table of configlets under the "Spines" container. The table includes columns for Name, Notes, Type, Created By, Created Date, and Proposed Configuration. One configlet named "SPINES" is selected, and its proposed configuration is displayed in a large text area on the right. The configuration includes basic network settings such as IP, domain name, and management interfaces.

- Next, let's assign the "**SPINES**" configlet to the "**Spines**" container. Right-click on the "**Spines**" container and select **Manage > Configlet**

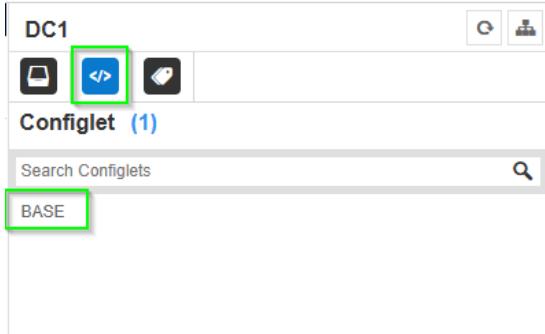
The screenshot shows the "Manage > Configlet" dialog for the "Spines" container. On the left, a sidebar shows the navigation path: Network Provisioning > Spines. The main area displays a list of configlets under the "Spines" container. A context menu is open over the "SPINES" configlet, with options like "Image Bundle", "Device", "View Config", "Synchronizations", "Reconcile", and "Remove". To the right, there is a preview pane showing the configuration details for the "SPINES" configlet, including its associated configlets, switches, and creation information.

- From the list, select the "**SPINES**" configlet and click "**Update**" at the bottom of the screen

5. Next, we're going to assign the “LEAF1” configlet to the “Leaf1” container. Expand the “Leafs” container to expose the individual leafs containers. Right-click on the “Leaf1” container and select **Manage > Configlet**. Select the “LEAF1” configlet and click “Update”

6. Repeat above steps for the “Leaf2” container
7. We've now assigned configlets to the appropriate containers. Don't forget to click “Save” at the bottom of the screen

8. From the Network Provisioning screen, we can verify which configlets are assigned to the containers. Click on a container in the hierarchy, in this example we'll highlight the “DC1” container. From the menu on the right side of the screen, click on the middle icon. This will show you which configlets are assigned to that container



Provisioning Switches

Now that we've assigned configlets to containers, we can move our switches from the “**undefined**” container to their assigned containers. Remember, all container-level configlets will be inherited by the devices under those containers.

In this lab we will provision our devices from ZTP mode, move them into their assigned containers, and apply their device-specific configlets.

1. Navigate to **Provisioning > Network Provisioning** and right-click on the “**Spines**” container. Now select **Add > Device**

Network Provisioning
Assign devices to containers and manage device-specific configuration

No data to be displayed

DC1

Configlet (1)

Search Configlets

BASE

2. Select “**Spine1**” and “**Spine2**” (denoted in the Serial Number) from the list and click “**Add**” at the bottom of the screen

Network Provisioning
Assign devices to containers and manage device-specific configuration

Network Provisioning > Spines > Device Add

Undefined Devices

Name	Serial No.	IP Address	Mac Address	Model	Version
<input type="checkbox"/> snr-192.168.225.242	6A0000C7F0E1028A9-BEAD03071EF715	192.168.225.242	0c:31:a5:66:04		4.29.10.1M
<input checked="" type="checkbox"/> snr-192.168.225.247	SN-PI-Spine1	192.168.225.247	0c:dc:8b:3e:x3:3e		4.29.1M
<input type="checkbox"/> snr-192.168.225.248	SN-PI-Leaf1A	192.168.225.248	0c:ea:82:9e:04:2d		4.29.1M
<input type="checkbox"/> snr-192.168.225.249	SN-PI-Leaf2B	192.168.225.249	0c:08:56:7e:54:43		4.29.1M
<input type="checkbox"/> snr-192.168.225.250	SN-PI-Leaf1B	192.168.225.250	0c:e5:22:fb:44:b7		4.29.1M
<input type="checkbox"/> snr-192.168.225.251	SN-PI-Leaf2A	192.168.225.251	0c:c4:70:00:02:3d		4.29.1M
<input checked="" type="checkbox"/> snr-192.168.225.252	SN-PI-Spine2	192.168.225.252	0c:2f:d7:03:73		4.29.1M

1-7 of 7

Add Back

NOTE: Do not click “**Save**” at this point as we have yet to assign any device-specific configlets to the switches. This is especially important when using static IP addresses to manage the devices. Otherwise we could lose access to the switches after provisioning.

3. Open the container hierarchy and right-click on the “Leaf1” container and select **Add > Device**
4. Select “Leaf1A” and “Leaf1B” from the list and click “Add”

Name	Serial No.	IP Address	Mac Address	Model	Version
sw-192.168.225.242	6A000FCF98E1023A8A8EAD00377EF715	192.168.225.242	0c:31:e3:60:88:04		4.28.10.1M
sw-192.168.225.247	SN-PI-Spine1	192.168.225.247	0c:dc:00:8e:x3:3e		4.29.7M
<input checked="" type="checkbox"/> sw-192.168.225.248	SN-PI-Leaf1A	192.168.225.248	0c:ea:82:8e:94:2d		4.29.7M
<input checked="" type="checkbox"/> sw-192.168.225.249	SN-PI-Leaf1B	192.168.225.249	0c:08:5d:76:54:43		4.29.7M
<input checked="" type="checkbox"/> sw-192.168.225.250		192.168.225.250	0c:e5:22:8e:44:b7		4.29.7M
<input type="checkbox"/> sw-192.168.225.251	SN-PI-Leaf2A	192.168.225.251	0c:ca:c7:80:02:3d		4.29.7M
<input type="checkbox"/> sw-192.168.225.252	SN-PI-Spine2	192.168.225.252	0c:2f:a7:c3:c3:73		4.29.7M

5. Let's repeat the process for container “Leaf2”, adding “Leaf2A” and “Leaf2B”

NOTE: Do not click “Save” at this point as we have yet to assign any device-specific configlets to the switches. This is especially important when using static IP addresses to manage the devices. Otherwise we could lose access to the switches after provisioning.

We are now ready to assign our device-specific configlets to the switches themselves.

6. Click on the “Spines” container and right-click on “Spine1”. Select **Manage > Configlet**

The screenshot shows the Arista Network Provisioning interface. On the left, a sidebar lists various provisioning and management options like Configlets, Image Repository, Tasks, Actions, Change Control, and Workspaces. The main area is titled "Network Provisioning" and shows a tree view of device hierarchies under "Spines". A context menu is open over a selected Spine device, listing options such as Manage, View, Image Bundle, Labels, IP Address, Snapshot, Factory Reset, Move, Replace, and Remove. To the right, there's a "Spines" section with a table showing device details like Name, IP Address, Mac Address, Serial No., Container, and Status. A "Configlet" section is also present.

- Now select the “spine-1” configlet from the list and click “Validate” at the bottom of the screen

This screenshot shows the "Network Provisioning" interface with the "Spines" section selected. A table lists configlets for the "spine-1" device. The "spine-1" configlet is highlighted with a blue selection bar. Below the table, a large panel displays the "Proposed Configuration" for the selected configlet, which includes detailed network configuration commands. At the bottom of this panel are "Validate" and "Cancel" buttons.

The screenshot shows the Arista Network Provisioning interface. On the left, a sidebar lists various management options like Provisioning, Configlets, Image Repository, Tasks, Actions, Change Control, Action Bundles, Templates, Studios, Workspaces, Snapshot Configuration, Public Cloud Accounts, Tags, and Zero Touch Provisioning. The main pane is titled "Network Provisioning" and "Assign devices to containers and manage device-specific configuration". It displays a comparison between "Current Management IP : 10.18.167.205" and "Proposed Management IP : 10.250.0.1". The "Designed Configuration" pane shows 261 total lines, 48 new lines, 3 mismatch lines, and 151 lines to reconcile. The "Running Configuration" pane shows 291 total lines. The configuration code includes commands like `username cpoadmin privilege 15 secret sha1\$12\$0\$0\$Ca0\$tmExPybISLyGK0mN\$g\$Yg\$Yz7ymu\$Crx` and `service configuration session max pending 10`. The interface uses color coding to highlight changes: green for new lines, red for mismatch lines, and blue for lines to reconcile.

- Under the “**Designed Configuration**” pane, we should see approximately **48** New Lines, **03** Mismatch Lines, and **151** Lines To Reconcile.

This screenshot shows the "Designed Configuration" pane with the following statistics: Total Lines : 261, New Lines : 48, Mismatch Lines : 03, To Reconcile : 151. Below these stats are buttons for "Save" and "Select Reconcile Lines".

- Verify the “**Designed Configuration**” against the “**Running Configuration**”. If you’re satisfied with the proposed changes, click “**Save**” at the bottom of the screen.
- Let’s repeat the above process for “**Spine2**” by assigning the “**spine-2**” configlet to that device. We should see the same number of proposed config changes for “**Spine2**” as we did for “**Spine1**”.
- Now let’s assign the device-specific configlets to the Leaf switches. Click on the “**Leaf1**” container and right-click on “**Leaf1A**” and select **Manage > Configlet**.

The screenshot shows the Arista Network Provisioning interface. On the left, a sidebar contains navigation links: Provisioning, Network Provisioning (selected), Configlets, Image Repository, Tasks, Actions, Change Control (selected), Action Bundles, Templates, Studios, Workspaces, Snapshot Configuration, Public Cloud Accounts, Tags, and Zero Touch Provisioning. The main pane is titled "Network Provisioning" and displays a list of devices under "Leaf1". The table includes columns for Name, IP Address, Mac Address, Serial No, Container, and Status. Two devices are listed: "sw-10.18.167.201" (IP 44.90.9c.4b.n6.5b, Serial SIN-P24Leaf1A, Container Leaf1, Status Leaf) and "sw-10.18.167.204" (IP 44.01.35.88.c2.53, Serial SIN-P24Leaf1B, Container Leaf1, Status Leaf). A context menu is open over the first device, showing options: Manage, Conflict, View, Image Bundle, Labels, IPAddress, Snapshot, Rollback, Factory Reset, Move, Replace, and Remove. At the bottom right of the main pane, there are "Preview", "Save", and "Cancel" buttons.

12. Now select “leaf-1a” from the list of configlets and click “Validate” at the bottom of the screen.

The screenshot shows the Arista Network Provisioning interface. The sidebar is identical to the previous screenshot. The main pane is titled "Network Provisioning" and shows a list of configlets for "sw-10.18.167.203.arista.local". The table includes columns for Name, Notes, Type - All, Created By, and Created Date. Several configlets are listed, including "BASE", "LEAF1", "SPINE3", and multiple "leaf" configlets (leaf-1a, leaf-1b, leaf-2a, leaf-2b, spine-1, spine-2). To the right of the table, a "Proposed Configuration" pane displays the configuration code for "leaf-1a". The code includes commands like "hostname Leaf-1A", "interface Ethernet1", "description Spine-1_ether1", "no switchport", "ip address 10.241.1.2/30", and various interface configurations for "leaf-1a" and other interfaces like "Vlan10", "Vlan20", "Vlan4093", and "Vlan4094". At the bottom right of the main pane, there are "Validate" and "Cancel" buttons.

13. Under the “Designed Configuration” pane, we should see approximately **77** New Lines, **03** Mismatch Lines, and **157** Lines To Reconcile.

Proposed Management IP :

Designed Configuration

Total Lines : 305 New Lines : 77 Mismatch Lines : 03 To Reconcile : 157 [Select Reconcile Lines](#)

14. Verify the “**Designed Configuration**” against the “**Running Configuration**”. If you’re satisfied with the proposed changes, click “**Save**” at the bottom of the screen.
15. Let’s repeat the above steps for “**Leaf1B**”, “**Leaf2A**”, & “**Leaf2B**” by assigning all device-level configlets to the appropriate switches.

NOTE: For “**Leaf1B**” we should see the same number of New Lines, Mismatch Lines and Lines To Reconcile as we did for “**Leaf1A**”. For “**Leaf2A & 2B**” we should see approximately **81** New Lines, **03** Mismatch Lines, and **157** Lines To Reconcile.

16. Once all configlets have been assigned to the individual switches, we can go ahead and click “**Save**” at the bottom of the Network Provisioning screen.
17. You should now notice that all of the switches in the Network Provisioning view still show up under the “**Undefined**” container. You’ll now see the switches are all shown in yellow with a “**T**” in the status section, indicating that these devices have available tasks.

The screenshot shows the Arista Network Provisioning interface. On the left is a sidebar with various navigation options: Provisioning, Network Provisioning (selected), Configlets, Image Repository, Upload and Download, Tasks, Actions, Change Control (with a dropdown menu), Action Bundles, Templates, Studios, Workspaces, Snapshot Configuration, Public Cloud Accounts, Tags, and Zero Touch Provisioning. The main area is titled "Network Provisioning" and contains a sub-header "Assign devices to containers and manage device-specific configuration". Below this is a search bar and a breadcrumb trail "Network Provisioning > Undefined". The main content is a table listing 7 switches, each with a yellow "T" icon in the "Status" column, indicating available tasks. The columns in the table are: Name, IP Address, Mac Address, Serial No., Container, and Status. The "Container" column shows all switches are currently in the "Undefined" container. The "Status" column shows all switches as "Available". The table also includes a "Associated Configlets" column (0), "Associated Switches" column (7), and a "Created by" column (hardy). At the bottom of the table are buttons for "Preview", "Save", and "Cancel".

We can execute these tasks using a Change Control - which we'll cover in the next section. Only after the tasks have been executed will the switches show up under their appropriate container structure in the Network Provisioning hierarchy.

Tasks & Change Controls

Tasks in CloudVision are generated whenever there are changes made to configlets, image bundles, etc. Those Tasks are executed using a Change Control operation. A Change Control allows a user to select and execute an individual task or a group of tasks that you want to process simultaneously.

In this lab we will assign our existing Tasks to a Change Control and then execute it so that our switches will finish the Provisioning process.

1. Navigate to **Provisioning > Tasks**. You should see our 6 pending “**Add Device**” Tasks. Under “**Assignable Tasks**” click the top check box next to “**ID**”. This will select all 6 tasks in the list. Now click “**Create Change Control**”

ID	Device	Creator	Type	Updated	Status
49	sw-192.168.225.248 MAC: 0cea8b86042d IP: 192.168.225.248	jhardy	Add Device	2 hours ago	Pending
48	sw-192.168.225.247 MAC: 0cadab8bea3e IP: 192.168.225.247	jhardy	Add Device	2 hours ago	Pending
47	sw-192.168.225.251 MAC: 0ccac7b0023d IP: 192.168.225.251	jhardy	Add Device	2 hours ago	Pending
46	sw-192.168.225.249 MAC: 0c085d7a5443 IP: 192.168.225.249	jhardy	Add Device	2 hours ago	Pending
45	sw-192.168.225.250 MAC: 0ca52fb4467 IP: 192.168.225.250	jhardy	Add Device	2 hours ago	Pending
44	sw-192.168.225.252 MAC: 0c2fafdf3c73 IP: 192.168.225.252	jhardy	Add Device	2 hours ago	Pending

ID	Device	Creator	Type	Updated	Status	Change Control
49	sw-192.168.225.248 MAC: 0cea8b86042d IP: 192.168.225.248	jhardy	Add Device	2 hours ago	Pending	
48	sw-192.168.225.247 MAC: 0cadab8bea3e IP: 192.168.225.247	jhardy	Add Device	2 hours ago	Pending	
47	sw-192.168.225.251 MAC: 0ccac7b0023d IP: 192.168.225.251	jhardy	Add Device	2 hours ago	Pending	
46	sw-192.168.225.249 MAC: 0c085d7a5443 IP: 192.168.225.249	jhardy	Add Device	2 hours ago	Pending	
45	sw-192.168.225.250 MAC: 0ca52fb4467 IP: 192.168.225.250	jhardy	Add Device	2 hours ago	Pending	
44	sw-192.168.225.252 MAC: 0c2fafdf3c73 IP: 192.168.225.252	jhardy	Add Device	2 hours ago	Pending	
43	Host-2 MAC: 0cf3f3ace37a IP: 192.168.225.217	adolson	Update Config	2 weeks ago	Completed	Change 2024-05-09-12-26-00

2. Under “**Select an Arrangement**” click on “**Parallel**” and then click on “**Create Change Control with 6 Tasks**”

Create Change Control > Change 2024-05-22-15-15-53

X

Select an Arrangement

 Series Parallel Template[Create Change Control with 6 Tasks](#)

Assignable Tasks

<input checked="" type="checkbox"/>	ID	Device	Creator	Type	Updated ↓	Status
Filter	Filter	Filter	Filter	Filter	Filter	Filter
<input checked="" type="checkbox"/>	49	sw-192.168.225.248 MAC: 0cea82:8604:2d IP: 192.168.225.248	jhardy	Add Device	2 hours ago	Pending
<input checked="" type="checkbox"/>	48	sw-192.168.225.247 MAC: 0cdad8:83e9:3e IP: 192.168.225.247	jhardy	Add Device	2 hours ago	Pending
<input checked="" type="checkbox"/>	47	sw-192.168.225.251 MAC: 0ccac7:b002:3d IP: 192.168.225.251	jhardy	Add Device	2 hours ago	Pending
<input checked="" type="checkbox"/>	46	sw-192.168.225.249 MAC: 0c08:5d7e:5443 IP: 192.168.225.249	jhardy	Add Device	2 hours ago	Pending
<input checked="" type="checkbox"/>	45	sw-192.168.225.250 MAC: 0ce5:22:fb44:b7 IP: 192.168.225.250	jhardy	Add Device	2 hours ago	Pending
<input checked="" type="checkbox"/>	44	sw-192.168.225.252 MAC: 0c2ffaff:3:c3:73 IP: 192.168.225.252	jhardy	Add Device	2 hours ago	Pending

[Export to CSV](#)

Showing 6 of 6 rows

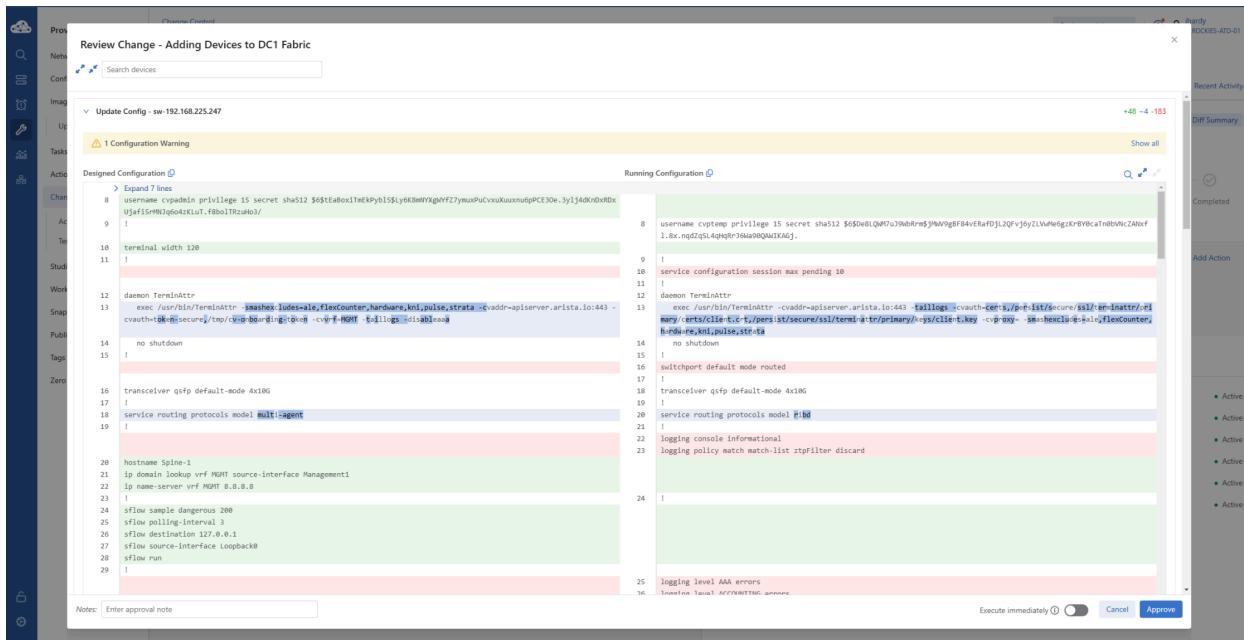
3. We're going to change the name of the Change Control to something more descriptive. This will allow us to quickly locate it later if necessary. Click on the pencil icon next to the name of the Change Control and name it, **"Adding Devices to DC1 Fabric"**

Change Control

Change 2024-05-22-15-15-53

Name	Change 2024-05-22-15-15-53	Description	Schedule Start
	--		Select date
		<input type="text" value="Search actions"/>	

4. In the upper right-hand corner, click on **"Review and Approve"**. Review the changes that are being applied to each device and then click **"Approve"** if you're satisfied with the proposed changes



While reviewing the Change Control, you'll notice that it looks very similar to what we saw when Validating the config changes on each of the devices. Again, you can see how many "**New Lines**", "**Mismatch Lines**", and "**Lines To Reconcile**" are being proposed for each device. You can also click on the ▾ next to each device to collapse its changes - which will provide you with a more summarized view.

Review Change - Change 2024-06-05-16-47-55

Search devices

Update Config - sw-10.18.167.204

Designed Configuration ⓘ

> Expand 7 lines

8 username cvpadmin privilege 15 secret sha512 \$6\$tEaBoxiTmEkPybl5\$Ly6K8mNYXgWYfZ7ymuxPuCvxu
xUjafi5rMNJq6o4zKLuT.f8bolTRzuHo3/

Review Change - Change 2024-06-05-16-47-55

Search devices

> Update Config - sw-10.18.167.204	+77 ~3 -157
> Update Config - sw-10.18.160.112	+81 ~3 -157
> Update Config - sw-10.18.167.206	+48 ~3 -151
> Update Config - sw-10.18.167.203	+77 ~3 -157
> Update Config - sw-10.18.167.205	+48 ~3 -151
> Update Config - sw-10.18.160.13	+81 ~3 -157
> Update Software - common changes on 6 devices	

Notes: Enter approval note

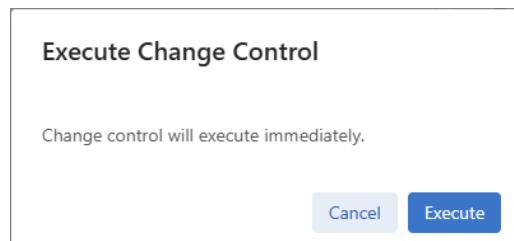
Execute immediately Cancel Approve

5. Once Approved, our Change Control should look something like this:

The screenshot shows the Arista Network's Change Control interface. On the left, there is a sidebar with various provisioning and management options. The main area is titled "Change Control" and shows a list of actions under "Adding Devices to DC1 Fabric". The actions listed are: "sw-192.168.225.252", "sw-192.168.225.250", "sw-192.168.225.249", "sw-192.168.225.251", "sw-192.168.225.247", and "sw-192.168.225.248", all categorized under "Add Device (Task 49)". To the right, there is a "Change Control Summary" section with a timeline showing "Last Edit" (by hardy), "Approved" (by hardy), "In Progress", and "Completed". Below this is an "Action Summary" section showing 6 actions, with a link to "Add Action". At the bottom, there are sections for "Device Status" (6 items), "Image Changes" (6 items), and "Configuration Changes" (6 items). A note at the bottom right indicates that no changes have been applied yet.

NOTE: No changes have actually been applied to the devices up to this point. The only way to push the changes is through the execution of the Change Control.

6. Now it's time to execute our Change Control. Click on “**Execute Change Control**” in the upper right corner. In the pop-up window, we'll click on “**Execute**”.



While the Change Control is running, you can click on “Logs” to get real-time updates as to what actions are being executed within the Change Control.

The screenshot shows the Arista CloudVision interface with the following details:

- Change Control Summary:** Root Execute (Parallel, Series).
- Timeline:** Last Edit (jhardy 48s ago), Approved (jhardy), Started (jhardy 33s ago), Completed.
- Action Summary:** 0% completion, Add Device button.
- Device Status:** 6 items listed: sw-192.168.225.233, sw-192.168.225.234, sw-192.168.225.235, sw-192.168.225.236, sw-192.168.225.237, sw-192.168.225.238. All devices are Active.

The screenshot shows the Arista CloudVision interface with the following details:

- Cloud Provisioning:** Add Devices to DC1 Fabric (Running).
- Change Control Stages:** 8 actions listed under 'Add Device' tasks, all completed successfully.
- Change Control Logs:**
 - Showing 15 logs.
 - Log entries:
 - Mapped device SN-P1-Spine1 to container Spines (sw-192.168.225.233) - May 22, 2024 16:16:09.920 GMT-6
 - Mapped device SN-P1-Leaf1B to container Leaf1 (sw-192.168.225.234) - May 22, 2024 16:16:09.221 GMT-6
 - Waiting for device SN-P1-Leaf1A to be streaming (sw-192.168.225.235) - May 22, 2024 16:16:08.329 GMT-6
 - Waiting for device SN-P1-Leaf2A to be streaming (sw-192.168.225.236) - May 22, 2024 16:16:07.975 GMT-6
 - Waiting for device SN-P1-Leaf2B to be streaming (sw-192.168.225.238) - May 22, 2024 16:16:07.897 GMT-6
 - Waiting for device SN-P1-Spine2 to be streaming (sw-192.168.225.237) - May 22, 2024 16:16:07.821 GMT-6

If any task in the Change Control fails, you'll see a Red “X” next to the task. You can click on the failed task and get more details as to why the action failed.

In the example below, we can see that switch **SN-P1-Spine1** exceeded the timeout after it was rebooted during the provisioning process.

The screenshot shows the Arista Cloud UI interface. On the left, the navigation sidebar includes options like Provisioning, Network Provisioning, Configlets, Image Repository, Tasks, Actions, Change Control (selected), Action Bundles, Templates, Studios, Workspaces, Snapshot Configuration, Public Cloud Accounts, Tags, Filter Management, and Zero Touch Provisioning. The main content area is titled "Change Control" and "Add Devices to DC1 Fabric". It shows a list of tasks under "Change Control Stages": Spine-2, Leaf-1A, Leaf-2A, Leaf-1B, and Leaf-2B. The task for "Leaf-1B" is marked as failed with a red error icon. A detailed view of the failed task for "Leaf-1B" shows the action name "Execute Task (341)", stage ID "QSMsvn2czuu56SxCAqd", and an error message: "Error rebooting device SN-P1-Spine1: rpc error: code = DeadlineExceeded desc = context deadline exceeded". Below this, the "Task Details" section shows the task was created by "hardy" on Jun 6, 2024, at 15:28:31, started at 15:30:09, and failed. The "Associated Device" section shows a summary for "sw-192.168.225.237" which is inactive, with details: vEOS-lab, 4.297M, 192.168.225.237, 16:51, 16:52, 16:53, 16:54, 16:55, Hostname, and sn-192-168-225-237.

If the Change Control completes with no errors or failed Tasks, then **Congratulations**, you have successfully deployed all of the switches in your DC1 fabric!

The screenshot shows the Arista Cloud UI interface, identical to the previous one but with a green success icon next to the task name. The main content area is titled "Change Control" and "Adding Devices to DC1 Fabric". It shows a list of tasks under "Change Control Stages": Spine-2, Leaf-1A, Leaf-2A, Spine-1, Leaf-1B, and Leaf-2B. All tasks are marked as completed with green checkmarks. A detailed view of the completed task for "Leaf-1B" shows the action name "Execute Task (24)", stage ID "QSMsvn2czuu56SxCAqd", and a success message: "CC Adding Devices to DC1 Fabric completed successfully". Below this, the "Change Control Logs" section lists several log entries from Jun 6, 2024, at 14:59:25.818 GMT-4, all indicating successful completion of various sub-tasks like "Add Device Task 19", "Add Device Task 20", "Add Device Task 21", "Add Device Task 22", "Add Device Task 23", and "Add Device Task 24".

**** If your Change Control failed for any reason, please reach out to an Arista team member for assistance. ****

Compliance Updates and Token

Compliance updates automatically keep the compliance settings within CloudVision up to date.

These Compliance Updates include:

- Bug Database
- Software End-of-Life
- Hardware End-of-Life

Let's start by configuring the token for Compliance Updates so that the items above stay current.

1. Navigate to **Settings > Compliance Updates** (Settings is the gear icon in the bottom left corner).

Date	Log Details	Time
May 30, 2024	Update successful (AlertBase released on 2024-05-29) - New updates written	03:27:41
May 29, 2024	Update successful	03:29:55
May 28, 2024	Update successful	04:42:55
May 27, 2024	Update successful	03:30:52
May 26, 2024	Update successful	03:26:08
May 25, 2024	Update successful	04:47:00
May 24, 2024	Update successful	04:47:28
May 23, 2024	Update successful	03:26:35
May 22, 2024	Update successful	03:28:36
May 21, 2024	Update successful	03:24:47
May 20, 2024	Update successful	02:49:32
	Update successful	14:50:21
	Update successful	05:12:49
	Update successful	03:25:07

2. In the “Authentication Token” box, enter the following token

```
4893c46ab1f074b733d48ddf2b60458f
```

3. Select “Save” to save the Token.

Date	Log Details	Time
May 30, 2024	Update successful (AlertBase released on 2024-05-29) + New updates written	03:27:41
May 29, 2024	Update successful	03:29:55
May 28, 2024	Update successful	04:42:55
May 27, 2024	Update successful	03:30:52
May 26, 2024	Update successful	03:26:08
May 25, 2024	Update successful	04:47:00
May 24, 2024	Update successful	04:47:00
May 23, 2024	Update successful	03:26:35
May 22, 2024	Update successful	03:28:36
May 21, 2024	Update successful	02:49:32
May 20, 2024	Update successful	14:50:21
May 20, 2024	Update successful	05:12:49
May 20, 2024	Update successful	03:25:07

NOTE: This token can be generated for customer environments by following the blue hyperlink to your Arista profile. Each token has a lifespan of one year, so keep in mind that it will need to be periodically updated.

Now that the Authentication Token has been supplied, CloudVision will periodically check for compliance updates daily.

Images and Image Bundles

CloudVision simplifies image management for all network devices. A more recent CloudVision feature now allows users to download Arista images and extensions directly from within CloudVision, minimizing the steps required to deploy images to network devices.

NOTE: The Compliance Token added in the previous section is required to be able to download images from within CloudVision

First, we must enable the feature that allows images to be downloaded from within CloudVision.

1. Let's navigate to **Settings > General Settings**
2. The General Settings page has a section called "**Features**" On the right side of your screen. Let's enable the "**Upload and Download Images and Extensions**" feature.

The screenshot shows the Arista General Settings page. On the left is a sidebar with various settings categories like Access Control, Audit Logs, and Developer Tools. The main area is titled "General Settings" with a sub-section "Basic Settings". On the right is a "Features" section containing numerous toggle switches for various cluster management and session management options. One specific feature, "Upload and Download Images and Extensions", is highlighted with a green border around its toggle switch.

3. Now let's Navigate to **Provisioning > Image Repository > Upload and Download Images**

The screenshot shows the Arista Provisioning > Image Repository > Upload and Download Images page. The sidebar includes Network Provisioning, Configuration, and various tasks like Change Control, Action Bundles, and Studios. The main content area is titled "Upload and Download Images" and displays a table of uploaded images. It shows two rows: vEOS-4.29.7M.swi (Filter, SWI, Version 4.29.7M-35648205.4297M) and vEOS-4.31.3M.swi (SWI, Version 4.31.3M-36737551.4313M). There are "Download Images" and "Upload Images" buttons at the top right of the table.

4. In the upper right corner, select “Download Images”
5. A pop-up window will appear showing the available images to be downloaded.
6. Expand **EOS-USA > Active Releases > 4.31 > EOS-4.31.3M**

7. Typically, you would click on the blue cloud icon to download an image, but we don't need to download any images for this lab.

NOTE: The images required for this lab are unique and have already been uploaded to CloudVision. **Please use the provided images in the steps going forward!**

8. Now that the image is downloaded, we need to create the Image Bundle that can be assigned to devices. Head to **Provisioning > Image Repository**.
9. Click on the “+” icon near the top of the right side of the screen.

The screenshot shows the Arista Image Repository interface. On the left is a navigation sidebar with various options like Provisioning, Network Provisioning, Configlets, and Image Repository (which is selected). The main area is titled "Image Repository" and "Manage images and image bundles". It has a search bar and a table titled "Images". The table has columns: Name, Containers, Devices, Notes, Uploaded by, and Uploaded Date. One row is visible: "vEOS-4.29.7M" with 0 containers, 0 devices, and uploaded by "aolson" on 2024-05-30 10:36:29. There are 1-1 of 1 results.

10. In the “Name” field, enter a meaningful name for the image. For this lab, we’ll use “**vEOS-4.31.3M**”.

The screenshot shows the "Create Image Bundle" dialog box. The left sidebar is identical to the previous screenshot. The main area shows a "Create Image Bundle" form with a "Name" field containing "vEOS-4.31.3M". Below the form are two large icons: a large disk icon in the center and a smaller disk icon in the top right corner. At the bottom are "Save" and "Cancel" buttons.

11. To add the Image to the Image Bundle, you can select either the large disk icon in the center of the page or the smaller disk icon on the upper right side of the screen.
12. An “Images” dialog box will appear. Select **vEOS-4.31.3M.swi** and then select the “Add” button.

The screenshot shows the Arista Image Repository interface. On the left, a sidebar lists various provisioning and management options. The 'Image Repository' section is selected. In the main area, a 'Create Image Bundle' form is displayed with the name 'vEOS-4.31.3M'. A modal window titled 'Images' is open, showing a list of files with the following data:

Name	Size	Version	Uploaded by	Uploaded Date	SHA512
vEOS-4.29.7M.swi	519.2 MB	4.29.7M-3564...	solsson	2024-05-30 ...	27b8294c1de...
vEOS-4.31.3M.swi	551.9 MB	4.31.3M-3673...	pfeit	2024-05-23 0...	aa759615c22...

At the bottom of the main page, there are 'Save' and 'Cancel' buttons.

13. You'll be returned to the “Create Image Bundle” page and should see the Image you just added. Click the “Save” button to save the image bundle.

The screenshot shows the 'Create Image Bundle' page after the image has been added. The 'Contents' section now lists the file 'vEOS-4.31.3M.swi'. The 'Save' button is visible at the bottom of the page.

14. The **vEOS-4.31.3M** image bundle will now appear in the Image Repository

Image Assignment

Now that an Image Bundle has been created, we can assign it to the devices in the topology.

1. Let's start by navigating to **Provisioning > Network Provisioning**
2. Right-click on the “DC1” container and select **Manage > Image Bundle**

3. Select the one created in the previous step, **vEOS-4.31.3M**, from the available image bundles.
4. Select the “**Update**” button at the bottom of the screen to assign the Image Bundle to all devices in the DC1 container.

Name	Containers	Notes	Uploaded by	Uploaded Date
vEOS-4.28.7M	0		aristson	2024-05-30 10:36:29
vEOS-4.31.3M	0		aristson	2024-05-30 10:44:23

5. You will now see a green hue surrounding the “**DC1**” container and the switches under the “**DC1**” hierarchy, indicating that changes must be saved. Click the “**Save**” button at the bottom of the screen to save the image assignment changes.

6. You'll now see the switches are all shown in yellow with a "T" in the status section, indicating that these devices have available tasks.

The screenshot shows the Arista Network Provisioning interface. On the left, a sidebar navigation menu includes: Provisioning, Network Provisioning (selected), Confglets, Image Repository, Tasks (selected), Actions, Change Control, Action Bundles, Templates, Studios, Workspaces, Snapshot Configuration, Public Cloud Accounts, Tags, and Zero Touch Provisioning. The main content area is titled "Network Provisioning" with the sub-section "Assign devices to containers and manage device-specific configuration". It displays a table of devices under "Tenant (6)". The table columns are: Name, IP Address, Mac Address, Serial No., Container, and Status. The devices listed are: Leaf-1A, Leaf-1B, Leaf-2A, Leaf-2B, Spine-1, and Spine-2. All devices show a yellow status icon with a 'T'. A message bar at the bottom right indicates: "Associated Configlets: 0", "Associated Switches: 0", "Created by: cvp system", and "Created on: 2024-04-02 18:10:14". At the bottom of the screen are buttons for "Preview", "Save", and "Cancel".

7. On the left side of the screen, select “Tasks.” You should see 6 pending tasks, one for each device in the topology.
 8. Select the 6 “Upgrade Image” tasks in the “Assignable Tasks” section.

The screenshot shows the Arista Tasks interface. On the left, a sidebar navigation menu includes: Provisioning, Network Provisioning, Confglets, Image Repository, Tasks (selected), Actions, Change Control, Action Bundles, Templates, Studios, Workspaces, Snapshot Configuration, Public Cloud Accounts, Tags, and Zero Touch Provisioning. The main content area is titled "Tasks" with the sub-section "View tasks and assign tasks to new change control operations". It shows a button "+ Create Change Control" and a red box "Cancel 6 Tasks". Below this is a section titled "Assignable Tasks" with a table of tasks. The table columns are: ID, Device, Creator, Type, Updated, and Status. The tasks listed are: Spine-1, Leaf-2A, Leaf-1B, Leaf-1A, Leaf-2B, and Spine-2. All tasks are marked as "Pending". At the bottom of the "Assignable Tasks" table is a link "Export to CSV". Below this is a section titled "All Tasks" with a similar table structure. Both sections show 6 rows of tasks, with the last row being "Showing 6 of 6 rows".

9. Now, we need to create a change control with those tasks, so select the “+ Create Change Control” button at the top of the screen.
10. The “Create Change Control” screen should pop up. Ensure all 6 tasks are selected and that “Series” is chosen. Then, choose “Create Change Control with 6 Tasks” on the top left of the pop-up.

The screenshot shows the Arista Network UI with the 'Change Control' section selected in the sidebar. A modal window titled 'Create Change Control > Change 2024-05-30-13-37-08' is open. Inside the modal, under 'Select an Arrangement', 'Series' is selected. Below it, a table lists 'Assignable Tasks' with columns: ID, Device, Creator, Type, Updated, and Status. There are 6 rows of tasks, each with a pending status. At the bottom right of the modal, there is a button labeled 'Create Change Control with 6 Tasks'.

ID	Device	Creator	Type	Updated	Status
85	Spine-1 MAC: 24:6f:d0:dba:0:7f IP: 192.168.0.13	aolsson	Upgrade Image	7 minutes ago	Pending
84	Leaf-2A MAC: 24:ed:2c:17:34:53 IP: 192.168.0.17	aolsson	Upgrade Image	7 minutes ago	Pending
83	Leaf-1B MAC: 24:b8:ba:90:62:ee IP: 192.168.0.11	aolsson	Upgrade Image	7 minutes ago	Pending
82	Leaf-1A MAC: 24:d4:99:e9:e0:f0 IP: 192.168.0.16	aolsson	Upgrade Image	7 minutes ago	Pending
81	Leaf-2B MAC: 24:c6:e0:df:0f:06 IP: 192.168.0.10	aolsson	Upgrade Image	7 minutes ago	Pending
80	Spine-2 MAC: 24:74:37:51:86:7a IP: 192.168.0.12	aolsson	Upgrade Image	7 minutes ago	Pending

11. You'll then be taken to the change control screen. Let's continue by giving the change control a meaningful name. At the top of the change control where it says “Name,” click the little pencil icon and enter “4.31.3 Image Upgrade” and press enter.

As you can see in the change control, every device in our topology will be upgraded in series. While that might look good, it's not an optimal upgrade process for our environment. For example, how do we know that MLAG Health is in a good state so we don't have an outage during the upgrade? How do we know that the MLAG timers have been completed and the switch is back online before taking down its peer for the upgrade? Let's leave the change control as is for now and return to it once we've completed the following steps.

ACTION BUNDLES

An action bundle is a collection of actions that are applied to the stage rule(s) of a change control template. The template is then used in a change control action, to organize the tasks into logical steps as administratively defined where the actions contained in its action bundles are executed.

Each action bundle can contain up to one task action and an unlimited number of non-task actions. You can apply the same action bundle to multiple change control templates.

Create an Action Bundle for the Leafs

Let's create a new action bundle for our Leafs that can help us do MLAG validation as part of the upgrade process.

1. Start by navigating to **Provisioning > Action Bundles**.
2. Select “**New Action Bundle**”.

3. In the “**Bundle Name**” field, enter a meaningful description for this action bundle. This Action Bundle will be used for Leaf upgrades, so we’ll use the name “**Leaf Upgrade Action Bundle**”.

4. Select the “**Add action**” dropdown box and choose the “**Lightweight Check MLAG Health**” action.

The screenshot shows the Arista Network Automation interface. On the left, a sidebar menu includes options like Provisioning, Network Provisioning, Conflicts, Image Repository, Tasks, Actions, Change Control, Action Bundles (which is selected), Templates, Studios, Workspaces, Snapshot Configuration, Public Cloud Accounts, Tags, and Zero Touch Provisioning. The main area is titled "Action Bundles" with the subtitle "Group actions into bundles for use in change control templates". A sub-dialog titled "New Action Bundle" is open on the right. It contains fields for "Bundle Name" (set to "Leaf Upgrade Action Bundle"), "Description (optional)" (empty), and a dropdown for "Lightweight Check MLAG Health" with "Parallel" selected. Below this is a list item "1. Lightweight Check MLAG Health" with "DeviceID" set to "Select a device...". The "checkDuration" field is set to "600". At the bottom right of the dialog are "Cancel" and "Save" buttons.

- In the “DeviceID” section, choose the “Provide via template” option and set the “checkDuration” to 600 (5 Minutes).

This screenshot is similar to the previous one, showing the "Action Bundles" creation dialog. The "DeviceID" dropdown now shows "Provide via template". The "checkDuration" field is still set to "600". The "Add action..." dropdown at the top right is populated with "Leaf Upgrade Action Bundle". The "Series" and "Parallel" buttons are also visible. The "Cancel" and "Save" buttons are at the bottom right.

NOTE: The “Add action” dropdown may stay populated with the name of the last action that was chosen

6. Select the “Add action” dropdown box and select the “Execute Task” action.

Action Bundle: Leaf Upgrade Action Bundle

Bundle Name: Leaf Upgrade Action Bundle

Description (optional):

Add action... Execute Task Parallel

1. Lightweight Check MLAG Health

DeviceID Provide via template

checkDuration 600

2. Task

Run this action with a pre-defined TaskID to execute the specified network changes.

TaskID (assigned by template)

Cancel Save

7. Select the “Add action” dropdown box again and select the “Lightweight Check MLAG Health” action.
8. In the “DeviceID” section, choose the “Provide via template” option and set the “checkDuration” to 600 (5 Minutes).

Action Bundle: Leaf Upgrade Action Bundle

Leaf Upgrade Action Bundle

Description (optional):

Lightweight Check MLAG Health Parallel

1. Lightweight Check MLAG Health

DeviceID Provide via template

checkDuration 600

2. Task

Run this action with a pre-defined TaskID to execute the specified network changes.

TaskID (assigned by template)

3. Lightweight Check MLAG Health

DeviceID Provide via template

checkDuration 600

Cancel Save

9. Select the “**Series**” option so that the actions in this Action Bundle are in series.
10. Select “**Save**” once the Action Bundle looks like the one in the screenshot below.

Action Bundles

Group actions into bundles for use in change control templates

Action Bundle: Leaf Upgrade Action Bundle

Bundle Name: Leaf Upgrade Action Bundle

Description (optional):

Add action... Series

1. Lightweight Check MLAG Health

DeviceID:

checkDuration: 600

2. Task:
Run this action with a pre-defined TaskID to execute the specified network changes.
TaskID:
(assigned by template)

3. Lightweight Check MLAG Health

DeviceID:

checkDuration: 600

Create an Action Bundle for the Spines

Now, let's repeat the process we just went through with a few changes tailored to the Spines. Since the Spines aren't in an MLAG pair, we don't need to do the MLAG Health Check, but we can use BGP Maintenance Mode since all the peers are running BGP and support the BGP GSHUT community.

1. Let's start by creating a new Action Bundle. Click “**New Action Bundle**” on the top right of the screen.

2. In the “**Bundle Name**” field, enter a meaningful description for this action bundle. This Action Bundle will be used for Spine upgrades, so we’ll use the name “**Spine Upgrade Action Bundle**.”

3. Select the “**Add action**” dropdown box and select the “**Enter BGP Maintenance Mode**” action.
4. In the “**DeviceID**” section, choose the “**Provide via template**” option.

NOTE: The “Add action” dropdown may stay populated with the name of the last action that was chosen

5. Select the “Add action” dropdown box and select the “Execute Task” action.

6. Select the “Add action” dropdown box again and select the “Exit BGP Maintenance Mode” action.

7. In the “**DeviceID**” section, choose the “Provide via template” option.

Action Bundles

Group actions into bundles for use in change control templates

Action Bundle: Spine Upgrade Action Bundle

Bundle Name: Spine Upgrade Action Bundle

Description (optional):

Add action... Series Parallel

1. Enter BGP Maintenance Mode

Pair this action with Exit BGP Maintenance Mode to run specific tests detailed in the EOS User Manual before reinserting the device into the network.

DeviceID:

2. Task

Run this action with a pre-defined TaskID to execute the specified network changes.

TaskID:
(assigned by template)

3. Exit BGP Maintenance Mode

Pair this action with Enter BGP Maintenance Mode to run specific tests detailed in the EOS User Manual before reinserting the device into the network.

DeviceID:

Cancel Save

8. Select “**Series**” so that the actions in this Action Bundle are in series.

9. Select “**Save**.”

Action Bundles

Group actions into bundles for use in change control templates

Action Bundle: Spine Upgrade Action Bundle

Bundle Name: Spine Upgrade Action Bundle

Description (optional):

Add action... Series Parallel

1. Enter BGP Maintenance Mode

Pair this action with Exit BGP Maintenance Mode to run specific tests detailed in the EOS User Manual before reinserting the device into the network.

DeviceID:

2. Task

Run this action with a pre-defined TaskID to execute the specified network changes.

TaskID:
(assigned by template)

3. Exit BGP Maintenance Mode

Pair this action with Enter BGP Maintenance Mode to run specific tests detailed in the EOS User Manual before reinserting the device into the network.

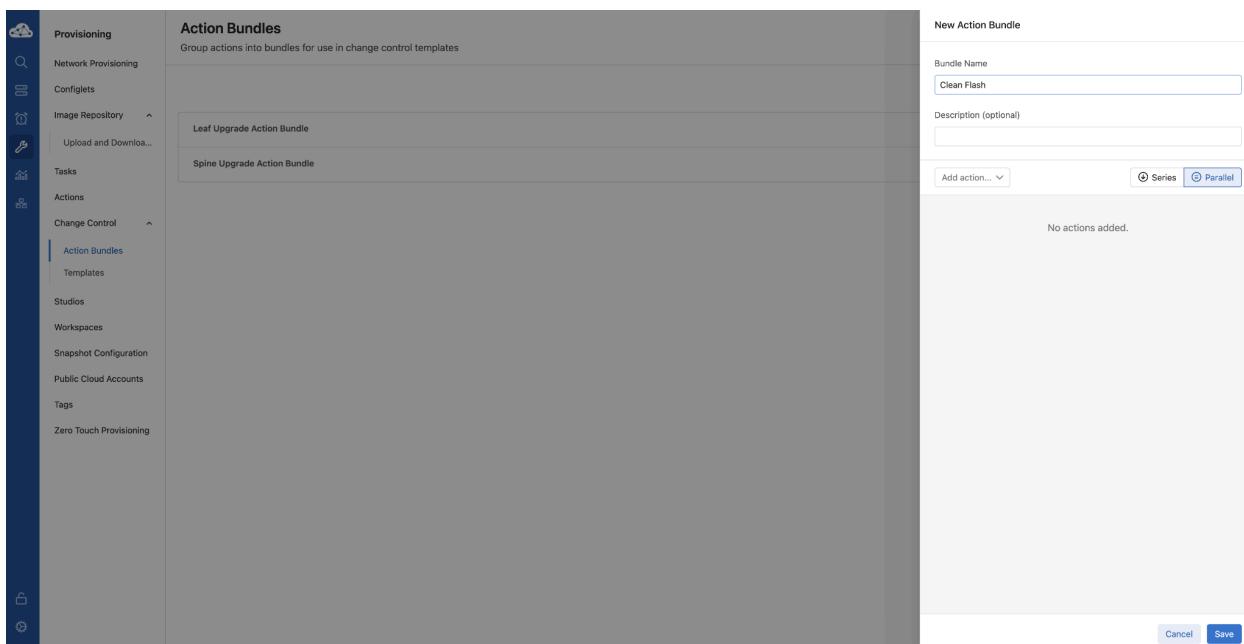
DeviceID:

Cancel Save

Create an Action Bundle to Cleanup the Flash

The final action bundle that we'll create will remove the old images from Flash so we can keep the Flash nice and tidy and not have old images hanging around forever.

1. Select “**New Action Bundle**”
2. In the “**Bundle Name**” field, enter a meaningful description for this action bundle. This Action Bundle will be used to clean up the flash, so we’ll use “**Clean Flash**.”



3. Select the “**Add action**” dropdown box and select the “**Clean Flash**” action.
4. In the “**DeviceID**” section, choose the “**Provide via template**” option.
5. In the “**FileSpecAndGlob**” section, enter:

```
flash:*.swi
```

6. Select “**Save**.”

7. You should now see the three Action Bundles that have been created.

TEMPLATES

A change control template is used as a structure for repeatable change control operations. It enables you to complete common and frequent changes in your network without needing to configure the details of the change control operation each time.

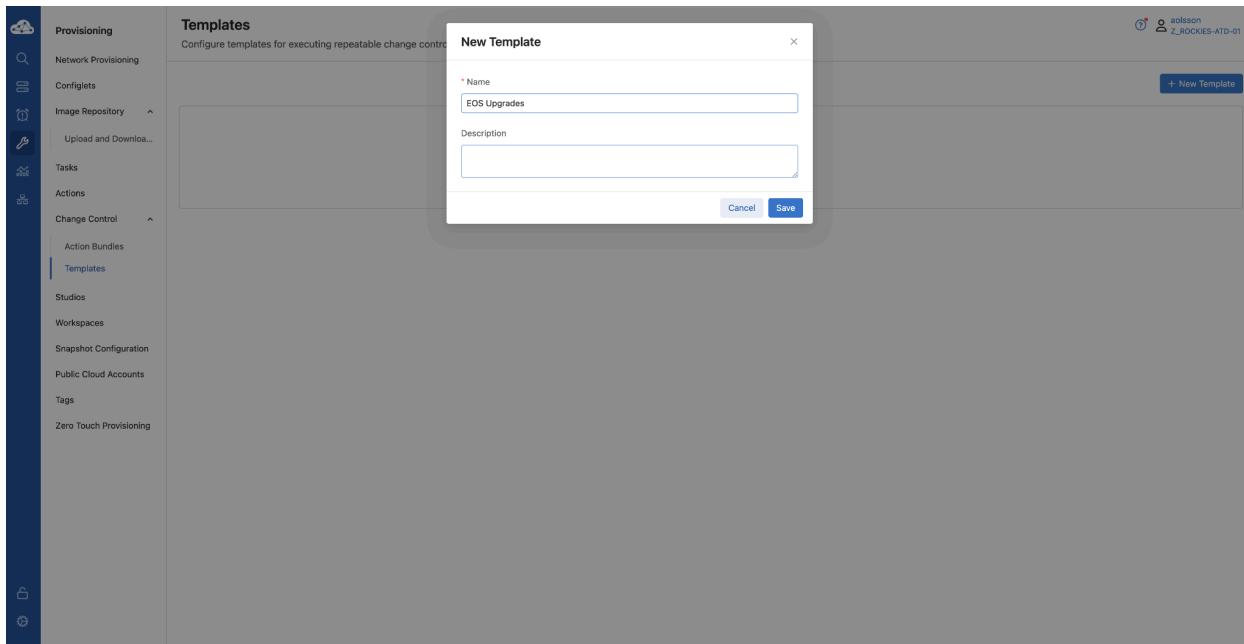
Two elements are used to construct a change control template, action bundles, and templates. Once a template has been created, you can then use it in all future change control operations.

Create a Template

1. The first thing we need to do is navigate to **Provisioning > Templates**.
2. Create a new template by selecting “**+ New Template**”.

The screenshot shows the Arista Provisioning interface. On the left is a sidebar with various navigation options: Network Provisioning, Configlets, Image Repository (with 'Upload and Download' sub-option), Tasks, Actions (with 'Change Control' sub-option), Action Bundles, **Templates** (which is selected and highlighted in blue), Studios, Workspaces, Snapshot Configuration, Public Cloud Accounts, Tags, and Zero Touch Provisioning. The main content area is titled 'Templates' with the subtitle 'Configure templates for executing repeatable change control operations'. It displays a message 'No data' and a small icon of a document with a gear. In the top right corner, there is a user profile icon with the name 'holsson' and the identifier 'Z_ROCKIES-ATD-01', along with a '+ New Template' button.

3. The “**New Template**” dialog box should appear. Enter a name for the new template. Since this template will be for upgrades, let’s call it “**EOS Upgrades**.”
4. Click “**Save**.”



5. Next, you'll be taken to a page where you can define the template options.

6. Now, let's add the first stage to the Template. On the right side of the page, select “**+ Add Stage Rule**”.

NOTE: The Stage rules need to match the order they are in the lab guide. Ensure you are clicking the “**Add Stage Rule**” of the last Stage in the Template.

7. Click on the pencil icon next to the “**Stage Rule Name**” of the first stage rule and name it “**Upgrade B Leafs.**” As you can probably tell from the name, this stage will only be used to upgrade the B Leafs.
8. Since we’ll upgrade the Leafs in this Template stage, choose “**Leaf Upgrade Action Bundle**” in the “**Action Bundle**” dropdown.
9. In the “**Device Filter**” dropdown, change the selection to “**Tag Query.**”
10. In the box just to the right of the “**Tag Query**” dropdown, we must select Leaf-1B and Leaf-2B. To do this, type “**Device:**” and choose Leaf-1B from the dropdown followed by a “,” then select Leaf-2B.

NOTE: Keep in mind that the Tag Key:Value pairs are case sensitive.

11. Change the “**Arrange Bundles**” dropdown from “**Series**” to “**Parallel**” so that this action can be run on both B switches at the same time.

12. Repeat the steps completed for the “**Upgrade B Leafs**” stage, but for “**Upgrade A Leafs.**” Be sure to select both Leaf-1A and Leaf-2A in the Tag Query.

13. Now that the stages for the Leafs are complete, we can create the next stage, which will be used to upgrade the spines. Click on “**+ Add Stage Rule**” once again.
14. Change the name of the stage to “**Upgrade Spines**.”
15. For the “**Action Bundle**,” we need to use the “**Spine Upgrade Action Bundle**”.
16. The Tag Query will be a bit different, we will use the “**Container**” tag for the Spines since both Spines are in the same container in Network Provisioning. Enter “**Container: Spines**” in the Tag Query field.
17. We'll leave the Arrange Bundles dropdown as “**Series**” because we want to upgrade one Spine at a time.

18. Add a final stage to the Template to clean the flash. We'll name it "**Clean Flash**". This will remove the old EOS Image after all the upgrades are completed. Select the "**Clean Flash**" action bundle created in the previous section in the Action Bundle dropdown.
19. The "**Device Filter**" can remain unchanged because we want to clean the flash on all the devices in the change control as part of this stage in the Template.
20. Change the "**Arrange Bundles**" dropdown from "**Series**" to "**Parallel**" so that this action can be run on all devices simultaneously.

21. Change the “**Stage Rules**” section at the top left of the page to “**Series**” so all stages in the template happen in series.
22. Click on “**Save Template**” on the top right of the screen.

Now that the Action Bundles and Upgrade Template have been created, they can be added to the change control created previously.

23. Head over to **Provisioning > Change Control**.
24. Select the “**4.31.3 Image Upgrade**” Change Control.

Recently Executed 2 Days ▾

- 1 Change Control
- Change 2024-05-30-12-27-07
 - Succeeded
 - May 30, 2024 12:28:38 GMT-6
- Change 2024-05-30-12-27-07
 - Started by aolsson
 - May 30, 2024 12:28:28 GMT-6
- Change 2024-05-30-12-27-07
 - Approved by aolsson
 - May 30, 2024 12:28:24 GMT-6
- Change 2024-05-30-12-27-07
 - Created by aolsson
 - May 30, 2024 12:27:12 GMT-6

25. Once in the change control, click the blue “**Select a Template**” dropdown box.
26. In the “**Change Control Template**” dialog box, select the “**EOS Upgrades**” Template created in the previous section.
27. Select “**Apply Template**”.

The screenshot shows the Arista Network's Change Control interface. On the left, a sidebar navigation menu includes: Provisioning, Network Provisioning, Configlets, Image Repository (selected), Upload and Download..., Tasks, Actions, Change Control (selected), Action Bundles, Templates, Studios, Workspaces, Snapshot Configuration, Public Cloud Accounts, Tags, and Zero Touch Provisioning. The main content area is titled "Change Control" and "4.31.3M Image Upgrades". It displays a list of actions under "Change Control Stages": Spine-2, Leaf-2B, Leaf-1A, Leaf-1B, Leaf-2A, and Spine-1. Each stage has an "Upgrade Image" task associated with it. A "Select a Template" dropdown is open, showing "EOS Upgrades". To the right, there is a "Change Control Summary" section with tabs for "Parallel" and "Series" execution, and status indicators for "Edit", "Approval", "In Progress", and "Completed". A "Recent Activity" section shows a log entry from "aoisson" 1m ago. Below the summary is an "Action Summary" section with a count of 6 actions and an "Image" link. At the bottom, there is a "Device Status" section listing devices: Leaf-1A, Leaf-1B, Leaf-2A, Leaf-2B, Spine-1, and Spine-2, all marked as "Active".

The change control will reflect what was defined in the upgrade template. While expanding the newly added stages and looking at the change control, remember that the green circle with the arrow in the center means that the action will be completed in series, and the purple circle with the two parallel lines means that those actions will be completed in parallel.

The screenshot shows the Arista Network's Change Control interface. On the left, a sidebar navigation menu includes: Provisioning, Network Provisioning, Configlets, Image Repository (with sub-options: Upload and Download...), Tasks, Actions, Change Control (selected), Action Bundles, Templates, Studios, Workspaces, Snapshot Configuration, Public Cloud Accounts, Tags, and Zero Touch Provisioning. The main content area is titled "Change Control" and "4.31.3M Image Upgrade". It displays a list of actions under "Change Control Stages": Upgrade B Leafs (6 actions), Upgrade A Leafs (6 actions), Upgrade Spines (6 actions), and Clean Flash (6 actions). To the right, there is a "Change Control Summary" section with tabs for Root Execute (Parallel selected), Last Edit (by aolsson 1m ago), Approval, In Progress, and Completed. Below this is an "Action Summary" section with counts for Image (6), Clean Flash (6), and Misc (12). Further down are sections for Device Status (6) and Image Changes (6), listing devices like Leaf-1A, Leaf-1B, Leaf-2A, Leaf-2B, Spine-1, and Spine-2, all marked as Active. Top right corner shows "Review and Approve" and user info (aolsson, Z_ROCKIES-ATD-01).

28. Expand each stage of the change control by Clicking the “+” next to each stage.
29. Validate that your change control looks similar to the screenshot below.

This screenshot shows the same Arista Network Change Control interface as above, but with the "Change Control Stages" expanded. Under "Upgrade B Leafs", it shows "Leaf-2B Upgrade Image" (3 actions), "Leaf-1B Upgrade Image" (3 actions), and "Leaf-1A Upgrade Image" (3 actions). Under "Upgrade A Leafs", it shows "Leaf-2A Upgrade Image" (3 actions), "Spine-2 Upgrade Image" (3 actions), and "Spine-1 Upgrade Image" (3 actions). Under "Clean Flash", it shows "Spine-2:Clean Flash" (1 action), "Leaf-2B:Clean Flash" (1 action), "Leaf-1A:Clean Flash" (1 action), "Leaf-1B:Clean Flash" (1 action), "Leaf-2A:Clean Flash" (1 action), and "Spine-1:Clean Flash" (1 action). The rest of the interface remains the same, including the summary, device status, and user information.

Now that the change control has been updated using the template, it can be executed as is. However, there's a process that can be used to preload the images on the devices to minimize upgrade times during a maintenance window.

30. On the bottom right section of the change control, click on “**Image Changes**”

The screenshot shows the Arista Cloud interface. The left sidebar has a 'Change Control' section selected. The main content area is titled '4.31.3M Image Upgrade' and lists various actions: Upgrade B Leafs, Upgrade A Leafs, Upgrade Spines, Clean Flash, Leaf-1B Upgrade Image, Leaf-1A Upgrade Image, Leaf-2A Upgrade Image, Spine-2 Upgrade Image, Spine-1 Upgrade Image, Leaf-2B:Clean Flash, Leaf-1A:Clean Flash, Leaf-1B:Clean Flash, Leaf-2A:Clean Flash, and Spine-1:Clean Flash. On the right, there's a 'Change Control Summary' with a timeline showing 'Last Edit' (aoisson 35s ago), 'Approval' (Pending), 'In Progress' (0), and 'Completed' (0). Below it is an 'Action Summary' with counts of 6 Images, 6 Clean Flash, and 12 Misc. The 'Device Status' section shows four devices: Leaf-1A, Leaf-1B, Leaf-2A, and Leaf-2B, each with a 'View Diff' link. At the bottom right is a blue 'Preload Images (6)' button.

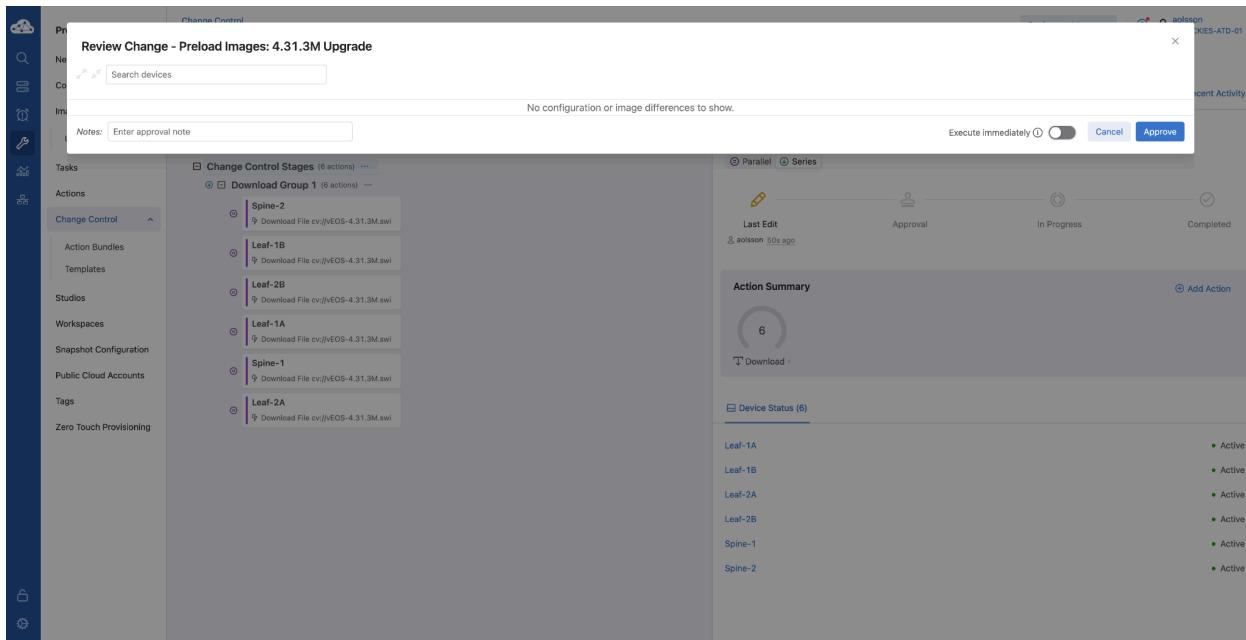
31. Select the blue “**Preload Images**” button.
32. A “**Preload Software Images**” dialog box appears. From here, the number of Parallel Downloads, which can be 6 for the lab, can be specified.
33. Click “**Create Change Control**” to generate a new change control to preload the images on all devices.

The screenshot shows the Arista CloudVision interface. On the left, a sidebar lists various provisioning and management options under 'Change Control'. The main area displays a 'Change Control' section for a '4.31.3M Upgrade' task. A modal window titled 'Preload Software Images' is overlaid, containing a summary of the upgrade process and a button to 'Create Change Control'. In the background, there's a 'Change Control Summary' section with tabs for 'Root Execute', 'Parallel', and 'Series' actions, and a 'Device Status' section showing the status of six devices.

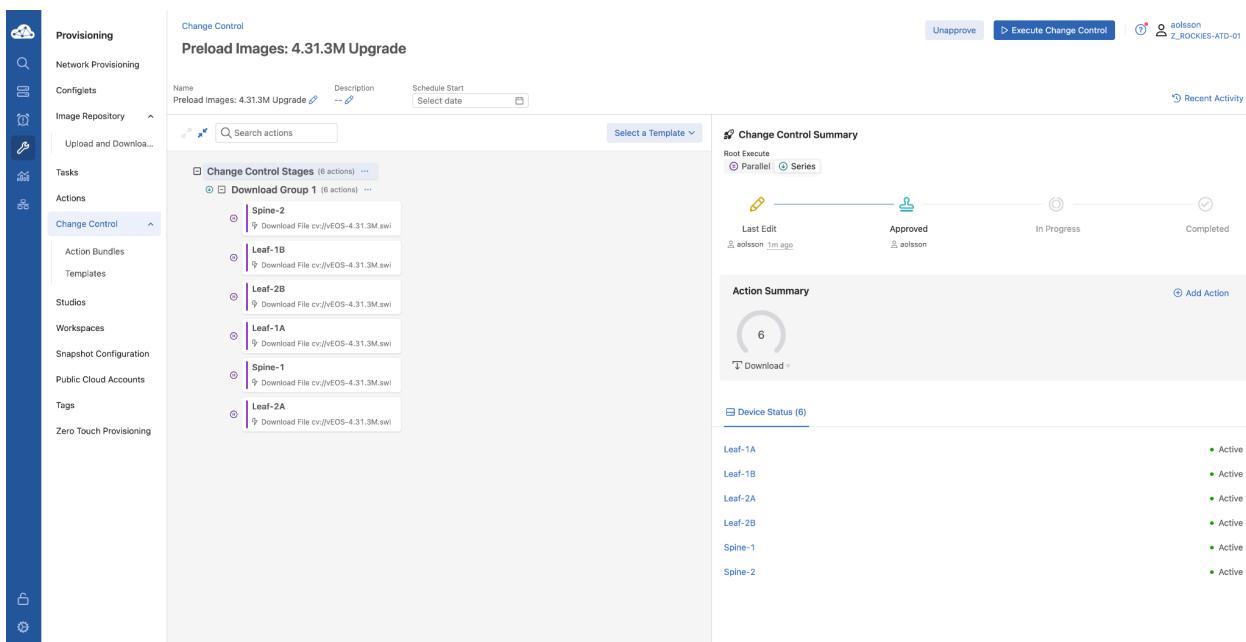
34. CloudVision will redirect you to the new “**Preload Images: 4.31.3M Upgrade**” change control.
35. Select “**Review and Approve**” in the upper right corner to review the change control.

The screenshot shows the Arista CloudVision interface. The sidebar and main 'Change Control' section are identical to the previous screenshot. A modal window titled 'Review Change' is open, displaying the message: 'No changes are being made to the device. This is expected because we’re just transferring the EOS image to the device, so no actual changes are being made.' Below this, there's a 'Change Control Summary' section and a 'Device Status' section.

36. The “**Review Change**” pop-up will appear and show no changes to the device are being made. This is expected because we’re just transferring the EOS image to the device, so no actual changes are being made.
37. Once satisfied with the change control review, select “**Approve**.”



38. Select the blue “**Execute Change Control**” button.



39. An “**Execute Change Control**” pop-up will appear. Select **Execute** to begin the change control execution.

40. The change control animations will appear next to each task in the change control, showing that all tasks are being executed.

41. As the tasks are complete, a green check will appear at the right of each task. Once the change control has been completed, a green “**Success**” message/label will appear to the right of the change control name.

Change Control
Preload Images: 4.31.3M Image Up... Success

Name: Preload Images: 4.31.3M Image Upgrades Description: --

Recent Activity

Change Control Summary

Last Edit: 3m ago Approved: aolsson Started: 45s ago Completed: 19s ago X 26s

Action Summary
100% Download

Device Status (6)

- Leaf-1A Active
- Leaf-1B Active
- Leaf-2A Active
- Leaf-2B Active
- Spine-1 Active
- Spine-2 Active

42. The upgrade change control can be executed now that the EOS images have been preloaded onto all devices. Head to **Provisioning > Change Control** and select the upgrade change control, which is named “**4.31.3M Image Upgrades**.”

Change Control
Manage, review, and execute change control operations

+ Create Change Control aolsson Z_ROCKIES-ATD-01

Date Range: 2024-05-24 → 2024-05-31

Recently Executed 2 Days

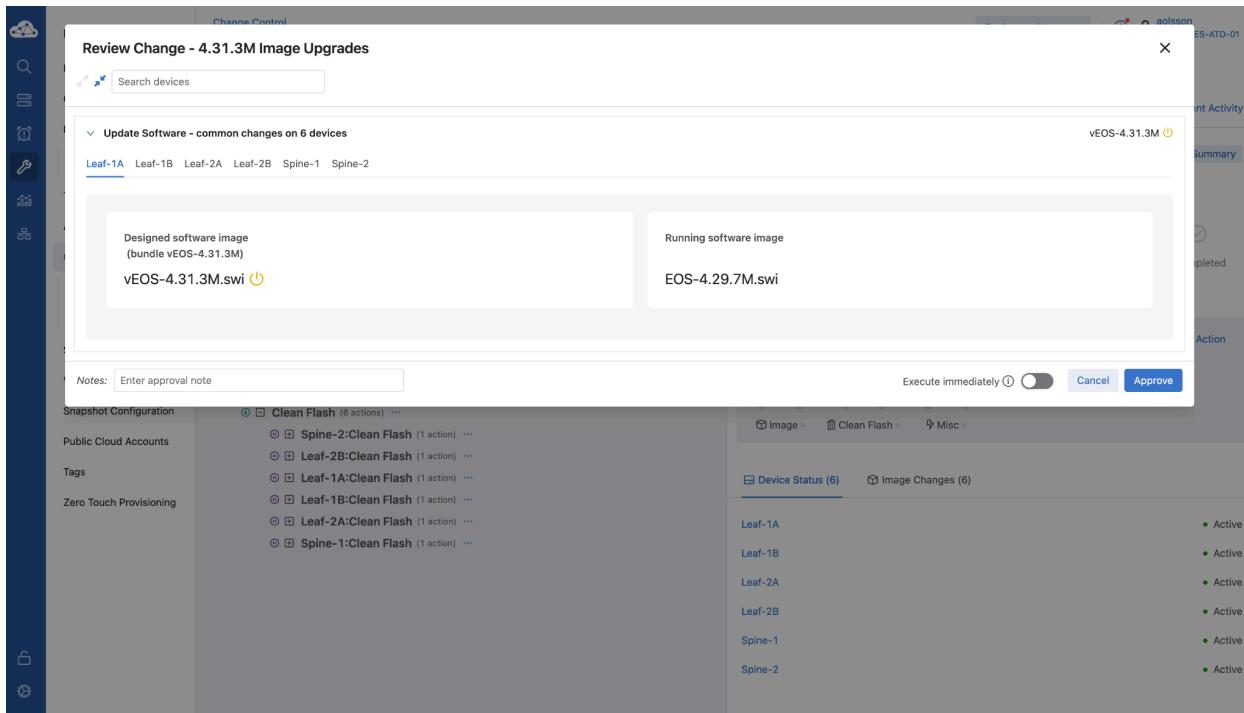
2 Change Controls

- Preload Images: 4.31.3M Image Up... Succeeded May 31, 2024 13:02:31 GMT-6
- Preload Images: 4.31.3M Image Up... Started by aolsson May 31, 2024 13:02:03 GMT-6
- Preload Images: 4.31.3M Image Up... Approved by aolsson May 31, 2024 13:02:03 GMT-6
- Preload Images: 4.31.3M Image Up... Created by aolsson May 31, 2024 12:59:38 GMT-6
- Change 2024-05-30-12-27-07 Succeeded May 30, 2024 12:28:38 GMT-6
- Change 2024-05-30-12-27-07 Started by aolsson May 30, 2024 12:28:28 GMT-6
- Change 2024-05-30-12-27-07 Approved by aolsson May 30, 2024 12:28:24 GMT-6
- Change 2024-05-30-12-27-07 Created by aolsson

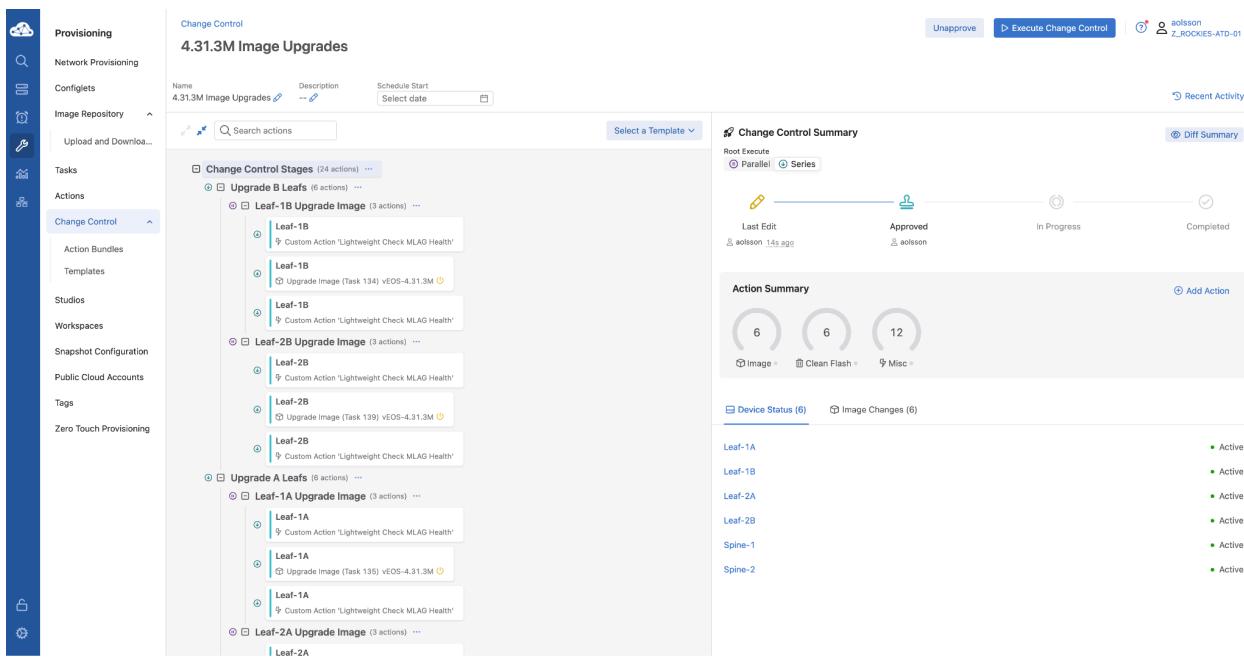
43. Select the blue “**Review and Approve**” button in the upper right corner.

The screenshot shows the Arista Network Management Platform's Change Control interface. A specific change control named "4.31.3M Image Upgrades" is selected. The interface includes a sidebar with various navigation options like Provisioning, Network Provisioning, Configlets, Image Repository, Tasks, Actions, and Change Control. The main content area displays a detailed list of upgrade actions for multiple devices. The timeline indicates the status of the execution, and the action summary provides a breakdown of the types of actions performed. Device status and image changes are also tracked.

44. The “Review Change” pop-up will appear for this change control and should indicate that EOS is being upgraded. The changes for each of the 6 devices should be grouped together because all 6 devices are running the same version of EOS and are being upgraded to the same version of EOS. Select the blue “Approve” button to approve the changes.



45. Select the blue “**Execute Change Control**” button in the upper right corner.



46. An “**Execute Change Control**” pop-up will appear. Select **Execute** to begin the change control execution.

47. Once again, the change control animations will appear next to each task in the change control as it runs. A green check mark will appear just to the right of each task when it is completed.

48. When the change control has been successfully completed, a green “**Success**” message/label will appear next to the change control name.

The screenshot shows the Arista CloudVision Change Control interface. On the left, a sidebar lists various management tasks like Provisioning, Network Provisioning, Configlets, and Change Control. The main area is titled "Change Control" and shows a "4.31.3M Image Upgrades" task with a "Success" status. The task details include the name "4.31.3M Image Upgrades" and a description. A search bar is available to find actions. The central part of the screen shows a hierarchical tree of upgrade stages for "Upgrade B Leafs" and "Upgrade A Leafs", each with multiple sub-actions like "Leaf-1B Upgrade Image". To the right, there's a "Change Control Summary" section with a timeline showing the last edit by "aolsson" 35m ago, the task being approved, started, and completed 34m ago. It also includes an "Action Summary" with three 100% completion circles for Image, Clean Flash, and Misc. Below that is a "Device Status" section listing devices like Leaf-1A, Leaf-1B, Leaf-2A, Leaf-2B, Spine-1, and Spine-2, all marked as Active.

Compliance - Device Configuration

CloudVision continuously computes image and configuration compliances. If a device is either configuration, image, or extension non-compliant, CVP automatically generates a non-compliant event on the Compliance dashboard and flags the device as non-compliant on the Inventory screen.

1. Navigate to **Devices > Inventory**.

The Inventory page will show that all devices are now running 4.31.3M but will also show a red “**Page**” icon in the “**Issues**” column that wasn’t there before the upgrade. This icon indicates that the running configuration on the devices differs from what CloudVision thinks it should be running. This happened because “**ribd**” was the default routing model before EOS 4.30, and “**multi-agent**” was the default for EOS 4.30 and later.

NOTE: Changing the routing model protocol from “**ribd**” to “**multi-agent**” requires a reboot for the changes to take effect. This lab will skip the reboot due to time limitations.

The screenshot shows the Arista CloudVision interface. On the left is a sidebar with icons for Devices, Inventory, Device Registration, Compliance Overview, Endpoint Overview, Connectivity Monitor, Traffic Flows, Endpoint Search, Comparison, Multi-Cloud Dashboard, and Network Segmentation. The main area is titled "Inventory" and displays a table of 7 devices. The columns include Device, Streaming, Issues, Model, Software, Streaming Agent, IP Address, MAC Address, and Device ID. Each device row contains a red "Page" icon next to the device name. The table shows the following data:

Device	Streaming	Issues	Model	Software	Streaming Agent	IP Address	MAC Address	Device ID
Leaf-1A	● Active	⚠️	vEOS-lab	4.31.3M	1.31.3_b9736e1e59c5d63 7	192.168.0.27	64:17:be:ab:ee:62	SN-P1-Leaf1A
Leaf-1B	● Active	⚠️	vEOS-lab	4.31.3M	1.31.3_b9736e1e59c5d63 7	192.168.0.26	64:17:80:b0:63:12	SN-P1-Leaf1B
Leaf-2A	● Active	⚠️	vEOS-lab	4.31.3M	1.31.3_b9736e1e59c5d63 7	192.168.0.28	64:97:69:d4:18:b3	SN-P1-Leaf2A
Leaf-2B	● Active	⚠️	vEOS-lab	4.31.3M	1.31.3_b9736e1e59c5d63 7	192.168.0.25	64:b8:4e:61:9:00	SN-P1-Leaf2B
Spine-1	● Active	⚠️	vEOS-lab	4.31.3M	1.31.3_b9736e1e59c5d63 7	192.168.0.30	64:bb:eb:b0:88:39	SN-P1-Spine1
Spine-2	● Active	⚠️	vEOS-lab	4.31.3M	1.31.3_b9736e1e59c5d63 7	192.168.0.29	64:16:af:1c:47:fc	SN-P1-Spine2
sw-10.18.168.61	● Active	⚠️	vEOS-lab	4.28.10.1M	1.22.3	192.168.0.10	64:9a:a7:52:d2:52	SN-P1-ztpUpdate

[Export to CSV](#)

Showing 7 of 7 rows

- Click on the red "Page" icon next to one of the devices. This will take you to that device's configuration "Compare Config" section.

The screenshot shows the configuration page for the device "Leaf-1A". The left sidebar includes sections for Device Overview, System, Processes, Storage, Log Messages, Hardware Capacity, Configuration (selected), Hardware, Snapshots, CVE and Bug Exp., Tags, Switching, Routing, and IP Tables. The main content area has tabs for Running Config, Designed Config, Compare Config (selected), and Config Sources. The "Compare Config" tab displays two panes: "Designed Configuration" and "Running Configuration". The "Designed Configuration" pane shows a single line of code: "service routing protocols model multi-agent". The "Running Configuration" pane shows a similar line of code with a timestamp: "18 service routing protocols model Field". Below these panes is a timeline from 15:00 to 13:00 on June 3, 2024, with a "Show: Live" button at the bottom right.

NOTE: It's always a good idea to validate device configuration compliance within CloudVision after an upgrade in case EOS default values have changed.

- To bring the device configuration back into compliance, head to **Devices > Compliance Overview**

Bug Exposure
7 devices

CVE Threats
7 devices

Identifier	Type	Summary	Severity	Device Count	Exposed Devices
500322	Bug	Management interfaces may flap with kernel message "transmit timed out, resetting" and should come back up automatically. On rare occasions an interface will remain down after such flap until the interface PCI device is reset or the system reloaded.	High	7	Leaf-1A, Leaf-1B, Leaf-2A, and 4 other devices
543510	Bug	The "show daemon" cli command displays the last PID if the agent has died and not yet restarted	Low	1	sw-10.1B.168.61
613653	Bug	"logging level" CLI in startup config does not work after reload. The workaround is to reconfigure the CLI after reload.	Low	7	Leaf-1A, Leaf-1B, Leaf-2A, and 4 other devices
672269	Bug	ConnectivityMonitor may restart unexpectedly if more than 250 http probes are configured	High	1	sw-10.1B.168.61
678460	Bug	After EOS upgrade on an MLAG pair, one or both peers may show unexpected/incorrect "show mlag config-sanity" output. To workaround, restart the MLAG agent on the first upgraded peer via "agent Mlag terminate", preferably during a maintenance window.	Low	4	Leaf-1A, Leaf-1B, Leaf-2A, and 1 other device
712490	Bug	Any non-EAPOL packet coming from an unknown MAC address will go through the MBA process and will show up under "show dot1x hosts" as successful/failed authentication based on the AAA server settings for that particular MAC address.	Low	1	sw-10.1B.168.61

- Select the “Configuration and Image” tab near the top.

Configuration Compliance
7 devices

Image Compliance
7 devices

Device	Status	Last Compliance Check
Leaf-1A	Configuration out of sync	Jun 3, 2024 10:16:31
Leaf-1B	Configuration out of sync	Jun 3, 2024 10:10:43
Leaf-2A	Configuration out of sync	Jun 3, 2024 10:16:30
Leaf-2B	Configuration out of sync	Jun 3, 2024 10:10:16
Spine-1	Configuration out of sync	Jun 3, 2024 10:31:04
Spine-2	Configuration out of sync	Jun 3, 2024 10:23:46

- Select all 6 devices that need to be brought back into compliance
- Click the blue “Sync Config” button

The screenshot shows the Arista Compliance Overview page. On the left, a sidebar menu includes: Devices, Inventory, Device Registration, Compliance Overview (selected), Endpoint Overview, Connectivity Monitor, Traffic Flows, Endpoint Search, Comparison, Multi-Cloud Dashboard, and Network Segmentation. The main content area has two donut charts: 'Configuration Compliance' (7 devices) and 'Image Compliance' (7 devices). Below each chart is a legend: Non-Compliant (red), Compliant (green), and No Data (grey). A table lists device status with columns for Device, Status, and Last Compliance Check. The table shows 6 rows of data. At the bottom is a timeline from 11:00 to 10:00.

7. You'll be redirected to the change control overview screen for the change control created to bring the devices back into compliance.

The screenshot shows the Arista Change Control Overview page for a task named "Sync Devices 2024-06-03-10-49...". The left sidebar includes: Provisioning, Network Provisioning, Conflicts, Image Repository (selected), Upload and Download, Tasks, Actions, Change Control (selected), Action Bundles, Templates, Studios, Workspaces, Snapshot Configuration, Public Cloud Accounts, Tags, Zero Touch Provisioning. The main content area shows the "Change Control Stages" section with 6 actions: Sync Leaf-1A, Sync Leaf-1B, Sync Leaf-2A, Sync Leaf-2B, Sync Spine-1, and Sync Spine-2. Each action has a status icon (green for Leaf-1A/B, grey for Leaf-2A/B, red for Spine-1/2). To the right, there's a "Change Control Summary" section with a timeline showing "Last Edit" (olson 24s ago), "Approval" (In Progress), and "Completed". Below it is an "Action Summary" (6 Config) and a "Device Status" section listing Leaf-1A, Leaf-1B, Leaf-2A, Leaf-2B, Spine-1, and Spine-2, all marked as "Active".

8. By default, all change controls created using the “Compliance Overview” page will be in series. For this lab, we want to run all changes in parallel, so select the “Change Control Stages” line item at the top of the screen in the section showing the tasks to be

- executed. Once the “**Change Control Stages**” line is highlighted, select the “**Parallel**” button on the right side of the screen just under the “**Change Control Summary**.”
- Select the blue “**Review and Approve**” button on the top right of the screen.

- The “**Review Change**” pop-up will appear, showing a single configuration change on all 6 devices. Select the blue “**Approve**” button to approve the change control.

- Select the blue “**Execute Change Control**” button on the top right.

12. The “Execute Change Control” pop-up will appear. Select the “Execute” button to continue.

13. Once completed, green check marks will appear next to each task, and the green “Success” icon will appear to the right of the change control name.

The screenshot shows the Arista CloudVision interface for Change Control. On the left, a sidebar lists various navigation options like Provisioning, Network Provisioning, Conflicts, Image Repository, Tasks, Actions, and Change Control (which is currently selected). The main content area displays a "Sync Devices 2024-06-03-10-49..." task. This task has a "Name" of "Sync Devices 2024-06-03-10-49-16" and a "Description" of "--". The status is "Success". Below this, there's a search bar and a list of "Change Control Stages" (6 actions) and "Sync Leaf-1A" (1 action), which includes sub-tasks for Leaf-1A, Leaf-1B, Leaf-2A, Leaf-2B, Sync Spine-1 (1 action), Sync Spine-2 (1 action), and Spine-2. Each sub-task shows a timestamp of "Jun 3, 2024 10:49:16" and a status of "+0 -1 -0" with a green checkmark. To the right, there's a "Change Control Summary" section with a timeline showing "Last Edit" (48m ago), "Approved" (48m ago), "Started" (48m ago), and "Completed" (45m ago). Below this is an "Action Summary" showing a 100% completion rate for "Config". At the bottom, there's a "Device Status" section listing six devices: Leaf-1A, Leaf-1B, Leaf-2A, Leaf-2B, Spine-1, and Spine-2, all marked as "Active" with green dots.

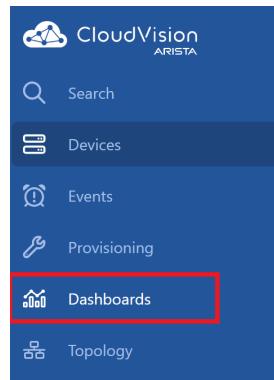
Dashboards

CloudVision allows you to create customizable dashboards consisting of multiple metrics across various datasets in different views. You can quickly resize and drag widgets on the grid to accommodate various custom layouts views. Data is gathered from the real-time telemetry data that the Arista switches are streaming to CVP.

Creating Custom Dashboards

In this lab we will be creating a custom dashboard that will be used as an overall network status page that can be used in a NOC as a visual representation for the overall state of the network.

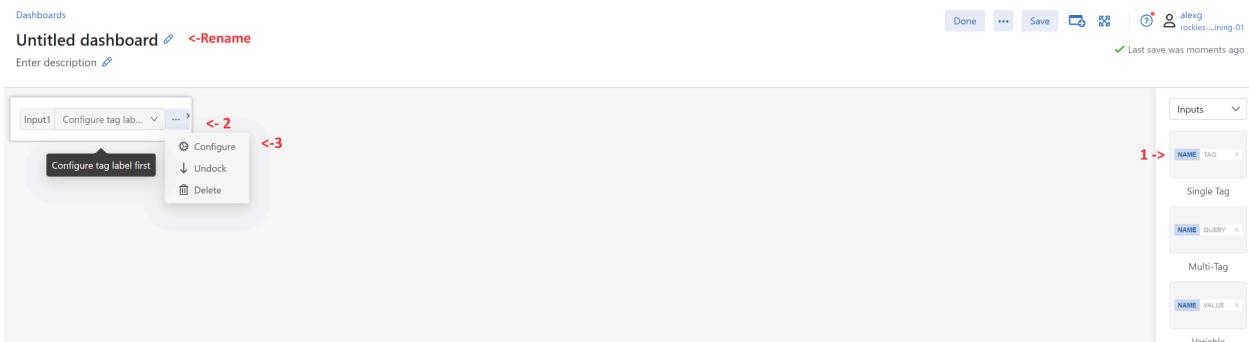
1. Navigate to the Dashboards section of CloudVision, and click on the “New Dashboard” button.



Dashboards

 + New Dashboard

2. Rename your dashboard to “<FirstName_LastInitial_NOC”.
- Select **Inputs > Single tag** from the dropdown menu on the right. Then click on the “3 dots (...) > Configure” for your input field.



3. Configure the Single Tag Panel with the name “Container”. Select “Container” as the “Tag Label”, and select “DC1” as the default value, close the configuration panel.

Configure Single Tag Panel

[Functionality](#)[Appearance](#)

Input Name

Container < Rename

Input Type

Devices

Tag Label ⓘ

Container < 1

Default Value ⓘ

DC1 < 2

Dataset ⓘ

Enter device tags query

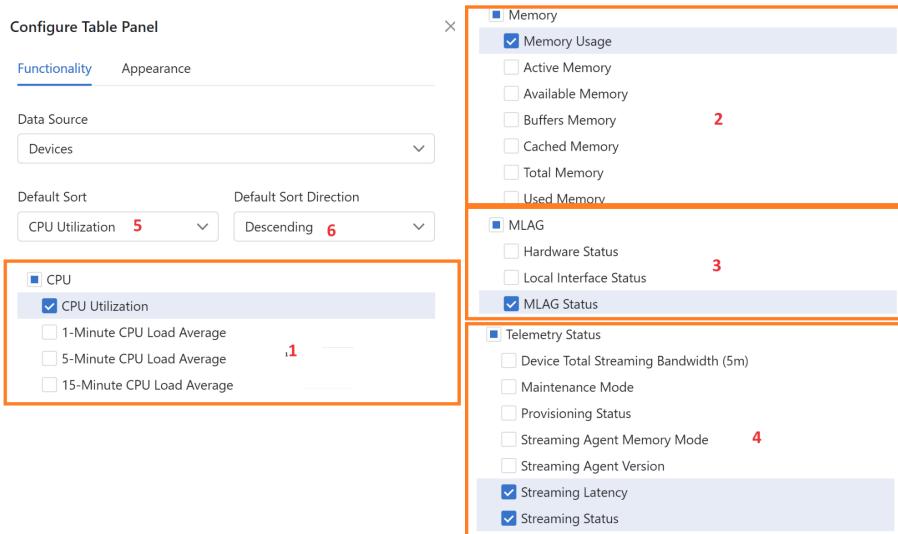
Dashboard Inputs ⓘ

Select

4. Insert a new graph by selecting “Metrics” from the drop down on the right, then select “Table”.



5. Configure this graph with the “CPU > CPU Utilization”, “Memory > Memory Usage”, “MLAG > MLAG Status”, and “Telemetry > Streaming Latency + Streaming Status” options in the picture below. Then sort by “CPU” > “Descending”, close the configuration panel.



6. Insert a new graph by selecting “Summaries” and then “Events”.



7. The preconfigured values for the “Events” dashboard are already correct.
8. Insert a new graph by selecting “Metrics” then “Horizon Graph”.
9. Select “Interfaces” as your Data Source, “Single Metric for Multiple Sources” as the View type, “RX Traffic Rate > Bitrate In” as your Metric, and “Device > *” as your Dataset, close the configuration panel.

Configure Horizon Graph Panel

Functionality Appearance

Data Source
Interfaces <-1

View Type
Single Metric for Multiple Sources <-2

Metric
Bitrate In <-3

Dataset ⓘ
device: * <-4

Dashboard Inputs ⓘ
Select

Dataset Summary
The final dataset includes the defined dataset and any assigned inputs:
device: *

10. Insert a new graph by selecting “**Summaries**” then “**Compliance**”.
11. Select “**Configuration Compliance**” for your Compliance Metric, close the configuration panel.
12. Insert another Compliance graph by selecting “**Summaries**” then “**Compliance**”.
13. Select “**Bug Exposure**” for your Compliance Metric, close the configuration panel.
14. Insert a new graph by selecting “**Metrics**” then “**Table**”
15. Select the “**Routing > ARP Table Size + IPv4 Total Route Count**”, “**Switching > Configured VLANs**”, “**MAC Addresses Learned**”, and “**Total VLANs**”, close the configuration panel

Configure Table Panel

Functionality Appearance

Data Source
Devices

Default Sort
Select Default Sort Direction Ascending

Metrics
Clear All

Routing
 ARP Table Size
 IPv4 Total Route Count
 IPv6 Total Route Count
 Multicast Sparse Mode Total Table Size
 Multicast Static Mode Total Table Size

Switching
 Configured VLANs
 MAC Addresses Learned
 Total VLANs

16. Insert a new graph by selecting “**Metric**” then “**Table**”

17. Select “**Interfaces**” as the Data Source, then select “**LANZ**” and all suboptions, the sort by “**LANZ Queue Length**” and “**Descending**”, close the configuration panel

Configure Table Panel >

Functionality Appearance

Data Source

Interfaces < Needs to be selected for LANZ options to appear ▾

Default Sort Default Sort Direction

LANZ Queue Length <-2 ▾ Descending <-3 ▾

LANZ <- 1

LANZ Queue Drops

LANZ Queue Length

LANZ Transmit Latency

18. The prior “**LANZ**” Dashboard is for instruction and will not be functional due to the lab running on **vEOS**, as this feature can only be run on hardware. We will demonstrate the **LANZ** feature in a later section of the Workshop.

19. Insert a new graph by selecting “**Summaries**” and then “**Traffic Flows**”

20. The default values for the “**Traffic Flows**” dashboard are already correct

21. Organize the dashboard panes as you see fit by dragging and dropping them on the page.

22. Your Dashboard should appear similar to the example below:.

The screenshot displays the Arista CloudVision interface. At the top, there's a navigation bar with tabs like 'Dashboard', 'Devices', 'Logs', 'Metrics', 'Events', 'Notifications', and 'Reports'. Below the navigation is a search bar and a 'Container: DC1' dropdown. The main area is divided into several sections: 'Device Table' showing CPU Utilization, Memory Usage, MLAG Status, Streaming Latency, and Discovery Status for various devices; 'Bridged Int' showing traffic statistics for interfaces like 'Eth1' and 'Eth2'; 'Device Table' showing ARP Table Size, IPsec Total Router Count, Configured VLANs, MAC Addresses Learned, and Total VLANs; 'Top Rows Grouped by Source Host' showing network statistics for hosts like '192.168.1.2', '192.168.1.1', '192.168.2.1', and '192.168.2.2'; 'Configuration Compliance' and 'Req Exposure' sections with circular progress bars; and an 'Interface Table' with columns for Interface, LAN2 Queue Depth, LAN2 Queue Length, and LAN2 Transmit Latency.

Events and Notifications

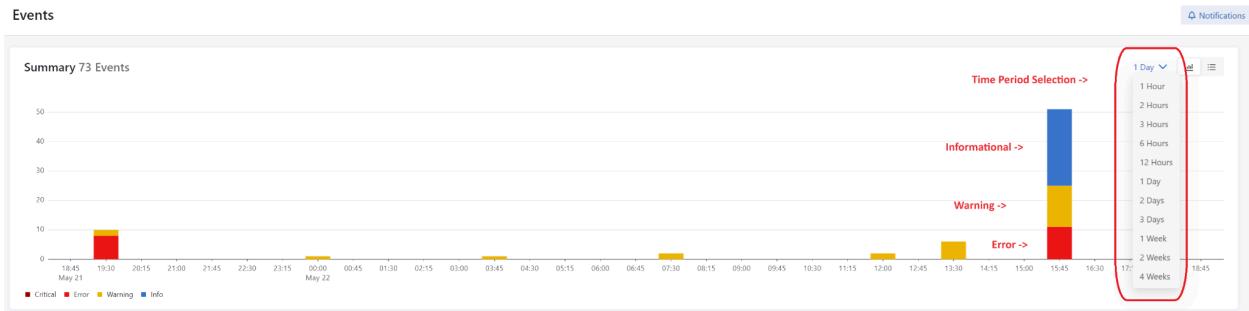
CloudVision events and notifications allow users to configure alerts for network events and receive notifications about them. Users can access the events screens by clicking the Events tab, which provides an overview of network events and allows users to change the time range and filter events. Users can also click on an event to view more details, acknowledge it to hide it, or export the events to a CSV file.

1. Navigate to the “Events” section of CloudVision

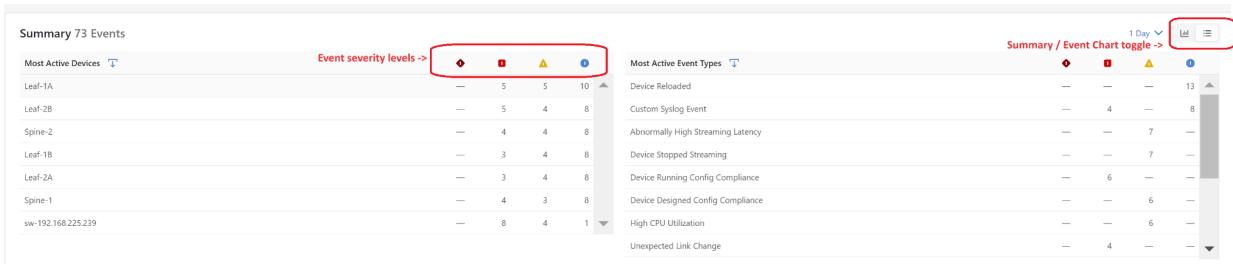
This screenshot shows the 'Events' section of CloudVision. On the left, there's a sidebar with icons for 'Events', 'Logs', 'Metrics', 'Devices', and 'Reports'. A red box highlights the 'Events' icon. The main area has a 'Summary 73 Events' section with a bar chart showing event counts over time. The x-axis represents time from 18:45 to 18:45+1 day. The y-axis represents event count from 0 to 100. A legend indicates event severity: Critical (red), Error (orange), Warning (yellow), and Info (green). Below the summary is an 'Event List' table with columns for Name, Source, Ack, Start Time, and Status. The table lists various events such as 'ECP Peer State Change', 'ECP Notification Detected', and 'ECP Peer State Change'. To the right of the event list is a 'Event Filters' panel with dropdowns for 'Starting before' (set to 'Current Time'), 'Severity' (set to 'Critical'), 'Description' (set to 'Events with any test'), 'Type' (set to 'Events of any type'), 'Source' (set to 'Enter logic query'), 'Acknowledgment State' (set to 'Unacknowledged'), and 'Status' (set to 'All').

By default, CloudVision “Events” are graphically displayed for the selected time period using colors to denote the severity of the event. The time period selection can be changed using the drop down near the top right corner of the graph.

Events



Swap views from the “Event Chart” to the “Summary Tables” using the toggle in the top right corner of the “Summary Events” pane. This allows you to view events by device and severity level for this time period, instead of viewing events by time and severity level.



There are many ways to filter the viewable events in CloudVision by using the “Event Filters” on the right side of the screen:

Time: Custom date and time periods can be selected instead of using only the current time.

Severity: Events can be filtered based on severity level.

Description: Events can be filtered based on matching text.

Type: Events can be filtered based on the pre-built event types within CloudVision.

Tags: Events can be filtered based on pre-built or custom user created Tags.

State: Events can be filtered based on whether they are Acknowledged or Unacknowledged.

Status: Events can be filtered based on whether they are currently Active or Ended.

Event Filters

Starting Before

Select custom time period ->

Severity Critical Error Warning Info

Description

Filter based on severity level ->

Filter based on custom search ->

Dropdown to filter events based on pre-built event types ->

Source

Filter based on Cludvision tags ->

Acknowledgement State

Filter based on Unacknowledged or Acknowledged events ->

Status

Filter based on Active events or historical events ->

Reset Filters

May 2024

<	May 2024	>				
Su	Mo	Tu	We	Th	Fr	Sa
28	29	30	1	2	3	4
5	6	7	8	9	10	11
12	13	14	15	16	17	18
19	20	21	22	23	24	25
26	27	28	29	30	31	1
2	3	4	5	6	7	8

19:40:25

Use current time

2. Reset any existing filters, then filter based on “reboot” using the “Description” field.

Event Filters

Starting Before

Severity Critical Error Warning Info

Description **2 ->**

Type

Source

Acknowledgement State

Status

3. Click on one of the “Device Rebooting” events in the “Event List”. (If your lab doesn’t contain a Device Rebooting event yet, you can follow along using the screenshots in the next section)

Event List				
Name	Source	Ack	Start Time	Status
+ Device Rebooting (12 Times)	(leaf-1B)	—	4h ago	—
+ Device Rebooting (13 Times)	(leaf-1A)	—	4h ago	—
+ Device Rebooting (12 Timed)	(leaf-2B)	—	4h ago	—
+ Device Rebooting (12 Times)	(leaf-2A)	—	4h ago	—
+ Device Rebooting (12 Times)	(Sonic-2)	—	4h ago	—
+ Device Rebooting (3 Times)	(Sonic-1)	—	4h ago	—
+ Device Rebooting (2 Times)	(SN-P1-Host2)	—	6d ago	—
+ Device Rebooting (2 Times)	(SN-P1-Host1)	—	6d ago	—
+ Device Rebooting (2 Times)	(0AAF3995B4D051781986503A1610C1C9C)	—	3w ago	—
+ Device Rebooting (2 Times)	(312F2E23610C96927DD46DE34C1061)	—	3w ago	—

Clicking on an event provides custom details based on the event type:

Time period: The exact time the event occurred is located at the top left. The overall time period from the event becoming active to when it ended is also displayed at the bottom of the screen.

Metrics: Based on the event type, specific metrics and graphs will be displayed providing more details

Logs: Syslog information from the device will be displayed and can be filtered based on search

Events

1 Device Rebooting on Leaf-1B
Happened May 22, 2024 16:19:53 (4h ago) <- Time period event occurred
The same event occurred 1 time in the 24-hour period before this event. Sync Data Visualizations

Event Description
Device SN-P1-Leaf1B is rebooting.
For more information on this event type view the CloudVision Event Guide <- Hyperlink to the Cloudvision Event Guide providing detailed event descriptions

Metric value differences from prior and after rebooting	Software Version	IPv4 Total Route Count	IPv6 Total Route Count	MAC Addresses Learned
4.29.7M updated 2 days ago View details	4.29.7M +512.5% 8 → 49 routes View differences	4.29.7M +50.0% 4 → 6 routes View differences	4.29.7M 0% 4 addrs View differences	
ARP Table Size +150.0% 2 → 5 entries View differences	ARP Table Size 0% 2 interfaces View differences	Port Channels 0% 0 interfaces View details	VXLAN Interfaces -46.2% 13 → 7 VLANs View details	
			Total VLANs	

Memory Overview

Memory Usage	16/19	16/20	16/21	16/22	16/23	16/24	16/25	16/26	16/27	16/28	16/29
Active Memory	4.29.7M										
Available Memory	1.041.7 MB										
Buffers Memory	1.041.7 MB										
Cached Memory	0.000 MB										
Total Memory	1.041.7 MB										
Used Memory	1.041.7 MB										

Output Power

No graphs to display

Fan Speeds

No graphs to display

Log Messages

Showing messages between May 22, 2024 16:18:53 GMT-4 and May 22, 2024 16:29:53 GMT-4
<- Text and graphical representation for the time period this event occurred

CPU Overview

Processor	16/19	16/20	16/21	16/22	16/23	16/24	16/25	16/26	16/27	16/28	16/29
Min CPU Load Average	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%
Min CPU Total Average	4.02%	4.02%	4.02%	4.02%	4.02%	4.02%	4.02%	4.02%	4.02%	4.02%	4.02%
Max CPU Load Average	2.21%	2.21%	2.21%	2.21%	2.21%	2.21%	2.21%	2.21%	2.21%	2.21%	2.21%
Max CPU Total Average	1.04%	1.04%	1.04%	1.04%	1.04%	1.04%	1.04%	1.04%	1.04%	1.04%	1.04%
Util CPU Utilization	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%

CPU Core Utilization

Show Last: 1h 30m 5m 30s

Log Messages

Time period of the provided syslog messages

Showing messages between May 22, 2024 16:18:53 GMT-6 and May 22, 2024 16:29:53 GMT-6

Total: 863

Time ↑

Message

Filter

BIOS

<- Search field to filter log messages, currently filtering using the word 'BIOS'

May 22, 2024 16:20:58 GMT-6	May 22 16:17:51 localhost kernel: [0.00000] BIOS-provided physical RAM map:
May 22, 2024 16:20:58 GMT-6	May 22 16:17:51 localhost kernel: [0.00000] BIOS-e820: [mem 0x0000000000000100-0x000000000009fbff] usable
May 22, 2024 16:20:58 GMT-6	May 22 16:17:51 localhost kernel: [0.00000] BIOS-e820: [mem 0x0000000000000fc00-0x000000000000ffff] reserved
May 22, 2024 16:20:58 GMT-6	May 22 16:17:51 localhost kernel: [0.00000] BIOS-e820: [mem 0x00000000000f0000-0x00000000000ffff] reserved
May 22, 2024 16:20:58 GMT-6	May 22 16:17:51 localhost kernel: [0.00000] BIOS-e820: [mem 0x0000000000100000-0x0000000007ffff] usable
May 22, 2024 16:20:58 GMT-6	May 22 16:17:51 localhost kernel: [0.00000] BIOS-e820: [mem 0x0000000007ffe0000-0x0000000007ffff] reserved
May 22, 2024 16:20:58 GMT-6	May 22 16:17:51 localhost kernel: [0.00000] BIOS-e820: [mem 0x000000000fffc000-0x000000000ffff] reserved
May 22, 2024 16:20:58 GMT-6	May 22 16:17:51 localhost kernel: [0.00000] SMBIOS 2.8 present.
May 22, 2024 16:20:58 GMT-6	May 22 16:17:51 localhost kernel: [0.00000] DMI: QEMU Standard PC (i440FX + PIIX, 1996), BIOS 1.10.2-1ubuntu1 04/01/2014

[Export to CSV](#)

Showing 10 of 863 rows (1 filter active)

Syslog messages from
the selected
timeframe containing
the word 'BIOS'



CloudVision Event Generation

1. Navigate to “Events”, then click on the “Event Generation” button near the top right corner

Events



Notifications Event Generation alej

The “Event Generation” page allows you to edit any of the pre-built events types within CloudVision. Let’s review some of the important fields:

Threshold: The value that must be reached before the event will trigger

Raise Time: The time period the threshold value must be met before the event will trigger

Clear Time: The time period the threshold value must be false before an Active event will end

Event type name <- Search box
IPSec RX Errors Breached Threshold

Event Rules

CloudVision allows you to create custom event rules to get the most useful alerts for your network. Use rules to suppress an event or to reduce its frequency, to generate an event for some devices but not others, or to create events with different severities when different thresholds are breached.

Event rules are checked in the order that they appear here. Rules can be reordered at any time. If there are no custom rules, the default rule or rules below will be used to generate this event.

Currently, events of this type are generated according to the default rule. Click **Add Rule** to create a custom rule.

+ Add Rule

Default

Rule Conditions

Active devices

The rule applies to all devices, unless device tags are selected.

Generate an Event

Time threshold value must be true before event triggers

Severity Threshold Raise Time Clear Time

Error Greater than 0 errors/s 0 sec 900 sec

Ignore Subsequent Events

Time threshold value must be exceeded before event is triggered

Time threshold value must be false before event clears

Rule Label Default rule

<- Prebuilt event types

2. In the search field for “Event type name”, type “CPU” then click on “CPU Utilization Breached Threshold” event type

The pre-built “CPU Utilization Breached Threshold” event type is set to trigger a **Warning** event if the CPU Utilization exceeds 85% for more than 30 seconds. We’re going to create an additional rule for this event that triggers an **Error** event for more serious situations where the CPU Utilization exceeds 90% for more than 60 seconds.

3. Click on “Add Rule”, then perform the following steps:

- a. Check the box for “Generate an Event”.
- b. Change the “Severity” level to “Error”.
- c. Set the “Threshold” value to “90”.
- d. Set the “Raise Time” value to “60”.
- e. Uncheck the box for “Ignore Subsequent Rules”.
- f. Click “Save” in the top right corner.

1 Rule Conditions

Active devices

The rule applies to all devices, unless device tags are selected.

Click here to select device tags

1 -> Generate an Event

Severity: Error

Threshold: Greater than 90% <- 3 %

Raise Time: 60 sec

Clear Time: 0 sec

2 -> Ignore Subsequent Rules

Rule Label: (Optional)

Value must be unique within the event type

+ Add Rule Below

CloudVision Event Notifications

1. Go back to the “Events” page, then click on the “Notifications” button near the top right

Events Notifications

2. Select the “Platforms” option for “Notification Configuration” on the left hand side of your screen.

Events

Notification Configuration

Status	Notification Platforms
Settings	CloudVision supports a range of email, messaging,
Platforms	Email
Receivers	SMTP
Rules	SMTP Host

The “Platforms” configuration is where you can add additional notification platforms for CloudVision to send event notification alerts to when events are triggered. CloudVision

supports many different platforms such as **Email**, **Slack**, **Zoom**, and **PagerDuty**. Additional platform options are regularly added with each new CloudVision update.

Most of these can be added fairly easily via the webhook URL for the specific application platform in question. Let's add two additional platforms this way.

3. Add the following URL for the "**Google Chat**" webhook:

<https://chat.googleapis.com/v1/spaces/AAAAAwXYY1tc/messages?key=AlzaSyDdl0hCZtE6vySjMm-WEfRq3CPzqKqqsHI&token=cKsi1s8vlpL-YTzSObdOgS-eAA2TAdyp-NMz79KdQs>

4. Add the following URL for the "**Slack**" webhook:

<https://hooks.slack.com/services/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>

The screenshot shows the 'Messaging Services' configuration page. It lists three platforms: Google Chat, Microsoft Teams, and Slack. For each, there is a 'URL' input field. The 'Google Chat' URL field contains the provided webhook URL, which is highlighted with a red box. The 'Slack' URL field also contains a valid webhook URL, also highlighted with a red box. The Microsoft Teams URL field is empty.

Now that we have added two new platforms for event notifications, we need to configure the "**Notification Receivers**" that we want to use for our event alerting.

5. Navigate to the "**Receivers**" sections in "**Notification Configuration**".

Notification Configuration

Status
Settings
Platforms
Receivers

Notification Receivers

Set up receivers to create a pool of persons or teams that you can assign to receive event notifications. You can configure each receiver to get notifications via one or more of the platforms listed above.



6. Click on “Add Receiver” and configure three different “Notification Receivers”. One for “Google Chat”, another for “Slack”, and a final one that includes Both. Click “Add Configuration” and select the corresponding items from the drop down menu for each receiver.

Notification Receivers

Set up receivers to create a pool of persons or teams that you can assign to receive event notifications. You can configure each receiver to get notifications via one or more of the platforms that you have enabled.

The screenshot shows three separate configuration panels for notification receivers, each with a red box highlighting specific fields or sections:

- Receiver Name:** Both Google and Slack
- Messaging Services:** Google Chat Configuration
- Configuration Options:** Send notification when events are resolved
- Platform Settings Overrides:** (link)
- Receiver Name:** Google
- Messaging Services:** Google Chat Configuration
- Configuration Options:** Send notification when events are resolved
- Platform Settings Overrides:** (link)
- Receiver Name:** Slack
- Messaging Services:** Slack Configuration
- Configuration Options:** Send notification when events are resolved
- Platform Settings Overrides:** (link)

At the bottom of each panel are buttons for **+ Add Configuration** and **Delete Receiver**.

You've now added two new event "Platforms" - Google Chat and Slack, and you've created three different groups of "Receivers" - Google Chat, Slack, and both Google Chat + Slack. The final step is to add the event "Rules" for which "Receiver" groups will be used for different events.

7. Navigate to the "Rules" section of "Notification Configuration".

Events

Notification Configuration

Status

Settings

Platforms

Receivers

Rules

Notification Rules

Create custom rules to determine which events are sent as notifications to your receivers. Rules are processed in the sequence that you order them.

1



We will now configure three different "Rules" that use the "Receivers" we just created to highlight how this process works during testing later.

8. For the first rule, we are going to select "Spine-1" as our "Device", and "Google" as our "Receiver".

Notification Rules

Create custom rules to determine which events are sent as notifications to your receivers. Rules are processed in the sequence that you order them.

1

Rule Conditions ①

Device	"SN-P1-Spine1" ×	
Add Conditions + Device Tags + Event Type + Interface Tags + Rule Labels + Severity		
Receiver <input style="width: 100%; border: 1px solid #ccc; height: 25px;" type="text" value="Google"/>		
<input type="checkbox"/> Continue Checking Rules ①		
↓ Move Down Delete		

9. For the second rule, we are going to select “**Spine-2**” as our “**Device**”, and “**Slack**” as our “**Receiver**”.

Rule Conditions ⓘ

Device "SN-P1-Spine2" X

Add Conditions + Device Tags + Event Type + Interface Tags + Rule Labels + Severity

Receiver Slack

Continue Checking Rules ⓘ

↑ Move Up ↓ Move Down Delete

10. For the third and last rule, we are going to select all Leafs, and “**Both Google and Slack**” as our “**Receiver**”.

Rule Conditions ⓘ

Device "SN-P1-Leaf1A" X "SN-P1-Leaf1B" X "SN-P1-Leaf2B" X "SN-P1-Leaf2A" X

Add Conditions + Device Tags + Event Type + Interface Tags + Rule Labels + Severity

Receiver Both Google and Slack

Continue Checking Rules ⓘ

↑ Move Up Delete

11. Click “**Save**” in the upper right hand corner.

We've set up everything required for event notifications. We can now test if our notifications are working correctly using the “**Test Notification**” tool in the “**Status**” section of “**Notifications**”.

12. Navigate to the “**Status**” section of “**Notifications**”.

Events

Notification Configuration



Settings

Platforms

Receivers

Rules

Notification System Status

Monitor the health of the notification system by

● Config back-end: **OK**

Last updated 16 hours ago

[Show recent status history](#)

13. We will now generate 3 different test notifications for: “**Spine-1**”, “**Spine-2**”, and any **Leaf**. Include your “<FirstName_LastInitial>” in the “**Rule Label**” field so you can identify your test notifications later in **Slack & Google Chat**.

Notification System Status

Monitor the health of the notification system by sending test notifications to check configurations.

● Config back-end: **OK**

Last updated 17 hours ago

[Show recent status history](#)

Test Notification Sender

Severity

◆ Critical

Event Type

IPSec RX Errors Breached Threshold

Device

No device

Generate 3 different Test Notifications using different Devices:

1: Spine 1

2: Spine2

3: Any Leaf

Interface

Select

Rule Label

Include your name in the ‘Rule Label’ so we can verify the source of Test Notifications

[Send Test Notification](#)

14. The **Google Chat** and **Slack** notifications will be displayed by your instructors.

ACTIONS

Actions can be used within a change control to run a series of commands on devices. There are a number of built-in actions that can be selected when configuring a change control operation or creating a change control template. You can also create custom actions by creating arguments and writing a Python 3 script to manipulate the arguments.

When configuring a change control, you'll select the dynamic values of an argument, like the device the action will run against and the image to install. When creating a custom action, you'll use dynamic arguments, like DeviceID, to enable users to select values when configuring a change control.

Create an Action

1. Navigate to **Provisioning > Actions**.
2. To create a new action, click on the “+ New Action” button at the top of the screen.

The screenshot shows the Arista Provisioning interface. The left sidebar is collapsed, showing navigation options like Provisioning, Network Provisioning, Configlets, Image Repository, Tasks, Actions (which is selected), Change Control, Studios, Workspaces, Snapshot Configuration, Public Cloud Accounts, Tags, and Zero Touch Provisioning. The main content area is titled 'Actions' and has a sub-header 'Create and manage actions for use in action bundles and change control operations'. It includes a search bar and a 'Custom Actions' section containing 'Bounce Interface', 'Generate Data Centers Node IDs', and 'Port Auto Description'. Below this is a 'Built-In Actions' section with 'Capture CLI Snapshot', 'Clean Flash', 'Enter BGP Maintenance Mode', 'Check MLAG Health', 'Download File', and 'Enter ZTP'. At the bottom are buttons for 'Execute Task' and 'Exit BGP Maintenance Mode'. In the top right corner, there is a user profile icon and a link to 'solsson _ROCKIES-ATD-01'. A '+ New Action' button is located in the top right of the main content area.

3. A “New Action” pop-up will appear. In the “Name” field, type your initials followed by “- Clear Counters.”
4. Click the blue “Save” button to save the new action.

The screenshot shows the Arista EOS UI interface. On the left is a sidebar with navigation links: Provisioning, Network Provisioning, Configlets, Image Repository, Tasks, Actions (selected), Change Control, Action Bundles, Templates, Studios, Workspaces, Snapshot Configuration, Public Cloud Accounts, Tags, and Zero Touch Provisioning. The main content area is titled 'Actions' and contains several sections: 'Custom Actions' (Bounce Interface, Generate Data Centers Node IDs), 'Port Auto Description', 'Built-In Actions' (Capture CLI Snapshot, Clean Flash, Enter BGP Maintenance Mode, Execute Task, Exit ZTP), and 'Check MLAG Health', 'Restart Termination Attribute', 'Download File', 'Enter ZTP', 'Exit BGP Maintenance Mode', and 'Interface Cable Test'. A modal window titled 'New Action' is open, prompting for 'Name' (ASE - Clear Counters), 'Description', and 'Action Type' (Change Control). The top right corner shows user information: aolsson, Z_ROCKIES-ATD-01, and a 'Save' button.

5. You'll then be redirected to the action configuration page for the new action.

The screenshot shows the configuration page for the 'ASE - Clear Counters' action. The left sidebar is identical to the previous screenshot. The main area shows the action details: Category (Change Control), Language (Python 3), and Description (Enter description). Below this are tabs for 'Edit Script' (selected) and 'Manage Arguments'. The 'Script' tab displays a code editor with the following Python script:

```

1

```

On the right, there are two sections: 'Static Arguments' (No data) and 'Dynamic Arguments' (No data). The top right corner shows a 'Duplicate Action' button and user information: aolsson, Z_ROCKIES-ATD-01.

6. We'll first want to create some arguments for the Device and Interface for the action to be run against. To do this, click the "**Manage Arguments**" button just below the action's name

The screenshot shows the Arista Cloud interface with the left sidebar expanded. Under the 'Actions' category, the 'ASE - Clear Counters' action is selected. In the main content area, the 'Manage Arguments' tab is active. A note at the top says: "To run an action against a device in a change control, create a dynamic argument with the name DeviceID and set it to required." Below this is a table with columns for Name, Type, Default, Description, Required, and Hidden. A message at the bottom indicates "No data". At the bottom of the table, there are buttons for "Static Argument" and "Dynamic Argument".

7. Create a new argument by selecting the "**+ Dynamic Argument**" at the bottom of the "**Manage Arguments**" section.

NOTE: Keep an eye on the Arguments and Variables as they are case sensitive.

8. A new line for the new argument will appear in the "**Manage Arguments**" section. Enter the following into the new line:

```
Name : DeviceID
Type : Dynamic
Default : Leave Blank
Description : The ID of the device to run this script against
Required : Yes
Hidden : No
```

The screenshot shows the Arista ASE interface. On the left, a sidebar lists various sections like Provisioning, Network Provisioning, Configlets, Image Repository, Tasks, and Actions (which is currently selected). The main area displays a script titled 'ASE - Clear Counters'. Under the 'Actions' tab, the 'Manage Arguments' tab is active. A table lists an argument named 'DeviceID' with the following details:

Name	Type	Description	Required	Hidden
DeviceID	Dynamic	The ID of the device to run this script against	Yes	No

At the bottom of the table, there are buttons for 'Static Argument' and 'Dynamic Argument'. There are also 'Save Changes' and 'Duplicate Action' buttons at the top right.

9. Repeat steps 7 & 8 to create the interface argument using the following values:

```

Name : Interface
Type : Dynamic
Default : Leave Blank
Description : The interface number of the interface to clear counters on
Required : Yes
Hidden : No

```

The screenshot shows the Arista Network's Action Management interface. On the left, a sidebar navigation includes 'Provisioning', 'Network Provisioning', 'Configlets', 'Image Repository', 'Upload and Download...', 'Tasks', 'Actions' (which is selected), 'Change Control', 'Action Bundles', 'Templates', 'Studios', 'Workspaces', 'Snapshot Configuration', 'Public Cloud Accounts', 'Tags', and 'Zero Touch Provisioning'. The main content area is titled 'Actions' and shows an action named 'ASE - Clear Counters'. It has a 'Category' of 'Change Control', 'Language' of 'Python 3', and a placeholder 'Enter description'. Below this are 'Edit Script' and 'Manage Arguments' buttons, with 'Manage Arguments' being the active tab. The 'Manage Arguments' section has a sub-instruction: 'To run an action against a device in a change control, create a dynamic argument with the name DeviceID and set it to required.' A table lists arguments: 'DeviceID' (Dynamic, required) and 'Interface' (Dynamic, required). Buttons for 'Save Changes' and 'Static Argument'/'Dynamic Argument' are also present.

10. Click the blue “**Save Changes**” button on the right side of the screen.
11. Now that the Arguments are created, we can head to the Script section to begin writing the action. Click on the “**Edit Script**” button below the Action name.

The screenshot shows the Arista Network's Action Management interface. The sidebar and action details are identical to the previous screenshot. The 'Edit Script' tab is now active, showing a large text area labeled 'Script' containing the number '1'. To the right of the script area are two panels: 'Static Arguments' (empty) and 'Dynamic Arguments' (listing 'DeviceID (Required)' and 'Interface (Required)'). Buttons for 'Execute' and 'Save Changes' are located at the top right of the script area.

12. On line 1 of the script we'll want to identify who wrote the script. Go ahead and enter the following on line 1, replacing “your-name-here” with your name:

```
# Written by your-name-here
```

13. The next thing we need to do is create a variable in the script that references the “**Interface**” argument that was created. To do that, we skip a line and then add the following on lines 3 & 4.

```
# Create a variable to reflect the "Interface" dynamic argument
interface = ctx.action.args.get("Interface")
```

14. Now, we need to add two different lists of commands that will need to be run on the switch. The first list will show the interface's existing counters, and the second will clear the interface counters. Skip a line and then paste the following into lines 6 - 16:

```
# List of commands to show interface counters
showCountersCmds = [
    'enable',
    f'show interfaces ethernet{interface} counters'
]

# List of commands to clear interface counters
clearCountersCmds = [
    'enable',
    f'clear counters ethernet{interface}'
]
```

15. Validate that the script section matches the following screenshot.

The screenshot shows the Arista Network Management Platform's 'Actions' section. A script named 'ASE - Clear Counters' is selected. The script is written in Python 3 and contains the following code:

```

1 # Written by your-name-here
2
3 # Create a variable to reflect the "Interface" dynamic argument
4 interface = ctx.action.args.get("Interface")
5
6 # List of commands to show interface counters
7 showCountersCmds = [
8     "enable",
9     "show interfaces ethernet{interface} counters"
10 ]
11
12 # List of commands to clear interface counters
13 clearCountersCmds = [
14     "enable",
15     "clear counters ethernet{interface}"
16 ]

```

The right panel displays 'Static Arguments' and 'Dynamic Arguments' sections. The 'Static Arguments' section shows a note 'No data'. The 'Dynamic Arguments' section lists 'DeviceID (Required)' and 'Interface (Required)'. There are also 'Duplicate Action' and user information buttons at the top right.

16. Let's continue editing the script by adding some logging stating that we're getting the values of the current interface counters. Skip another line and then add the following to lines 18 & 19:

```
# Create log entry
ctx.alog(f'Getting counters for Ethernet{interface}')
```

17. Now, we can proceed with running the commands listed in the "showCountersCmds" list on the switch. To do that, add another blank line and the following to lines 21 and 22.

```
# Run the list of commands to get counters
cmdResponse = ctx.runDeviceCmds(showCountersCmds)
```

18. The commands have now been run on the switch and the response has been stored in the variable "**cmdResponse**." Let's create a log entry that outputs the response. On lines 24 - 25 add the following:

```
# Create a log entry of the counters
ctx.alog(f'The current interface counters are : {cmdResponse[1]["response"]}')
```

NOTE: The cmdResponse value returned above is similar to the following:

```
[{'error': '', 'response': {}},  
 {'error': '', 'response': {  
     'interfaces': {  
         'Ethernet1': {  
             'inBroadcastPkts': 1,  
             'inDiscards': 0,  
             'inMulticastPkts': 38,  
             'inOctets': 7529,  
             'inUcastPkts': 0,  
             'lastUpdateTimestamp': 1717697811.8986592,  
             'outBroadcastPkts': 836, 'outDiscards': 0,  
             'outMulticastPkts': 37,  
             'outOctets': 45227, 'outUcastPkts': 0}  
         }  
     }  
}]
```

The cmdResponse list contains a dictionary for each command run on the device. Since we're interested in the "show interfaces ethernet counters" command response value, we need to specify "[1]" to indicate the second item in the list (since list values start at the 0th item in Python) and "[“response”]" for the response data.

19. Now that we have logged what the counters were, let's proceed with clearing the counters. Let's create a blank line, then create another log entry stating that we will clear the counters. On lines 27 - 28 enter the following:

```
# Create log entry before clearing the counters  
ctx.alog(f'Clearing counters for Ethernet{interface}')
```

20. Let's actually clear the counters now. Let's skip another line and then enter the following on lines 30 - 31:

```
# Run the list of commands to clear counters  
cmdResponse = ctx.runDeviceCmds(clearCountersCmds)
```

21. Now that the counters are cleared let's create another log entry stating that the counters have been successfully cleared. To do that, let's skip another line and add the following content to lines 33 - 34:

```
# Create log entry
ctx.alog(f'Counters for Ethernet{interface} Sucessfully Cleared')
```

22. For the final part of the script, let's repeat steps 16 - 18 to get the interface's current counter values and log them.
23. Validate that the script matches the screenshot below, then click the blue “**Save Changes**” button to save the action.

The screenshot shows the Arista EOS UI interface. On the left, there is a sidebar with various navigation options like Provisioning, Network Provisioning, Configlets, Image Repository, Tasks, Actions, Change Control, Action Bundles, Templates, Studios, Workspaces, Snapshot Configuration, Public Cloud Accounts, Tags, and Zero Touch Provisioning. The 'Actions' option is selected. In the main area, there is a card for 'ASE - Clear Counters'. The 'Script' tab is active, displaying the following Python code:

```

1 # Written by your-name-here
2
3 # Create a variable to reflect the "Interface" dynamic argument
4 interface = ctx.action.args.get("Interface")
5
6 # List of commands to show interface counters
7 showCountersCmds = [
8     "enable",
9     "show interfaces ethernet{interface} counters"
10 ]
11
12 # List of commands to clear interface counters
13 clearCountersCmds = [
14     "enable",
15     "clear counters ethernet{interface}"
16 ]
17
18 # Create log entry
19 ctx.alog(f'Getting counters for Ethernet{interface}')
20
21 # Run the list of commands to get counters
22 cmdResponse = ctx.runDeviceCmds(showCountersCmds)
23
24 # Create a log entry of the counters
25 ctx.alog(f'The current interface counters are : {cmdResponse[0]["response"]}')
26
27 # Create log entry prior to clearing the counters
28 ctx.alog(f'Clearing counters for Ethernet{interface}')
29
30 # Run the list of commands to clear counters
31 cmdResponse = ctx.runDeviceCmds(clearCountersCmds)
32
33 # Create log entry
34 ctx.alog(f'Counters for Ethernet{interface} Sucessfully Cleared')
35
36 # Create log entry
37 ctx.alog(f'Getting counters for Ethernet{interface}')
38
39 # Run the list of commands to get counters
40 cmdResponse = ctx.runDeviceCmds(showCountersCmds)
41
42 # Create a log entry of the counters
43 ctx.alog(f'The current interface counters are : {cmdResponse[1]["response"]}')

```

To the right of the script editor, there are sections for 'Static Arguments' (which is currently empty) and 'Dynamic Arguments' (which shows 'DeviceID (Required)' and 'Interface (Required)'). At the top right of the card, there are buttons for 'Duplicate Action', a user icon, and the text 'aoisson Z_ROCKIES-ATD-01'.

24. Now, we can proceed with testing our Action. To do so, click on the light blue “**Execute**” button.
25. An “**Execute Action**” dialog box will appear on the right side of the screen. Click “**Dynamic Argument Values**” to expand the arguments section.

The screenshot shows the Arista EOS CLI interface. On the left, there's a sidebar with various navigation options like Provisioning, Network Provisioning, Configlets, Image Repository, Tasks, Actions, Change Control, Action Bundles, Templates, Studios, Workspaces, Snapshot Configuration, Public Cloud Accounts, Tags, and Zero Touch Provisioning. The 'Actions' option is currently selected. In the main area, there's a title 'ASE - Clear Counters' with a 'Description' field containing 'Enter description'. Below that is a 'Script' section with the following Python code:

```

1 # Written by your-name-here
2
3 # Create a variable to reflect the "Interface" dynamic argument
4 interface = ctx.action.args.get("Interface")
5
6 # List of commands to show interface counters
7 showCountersCmds = [
8     "enable",
9     "show interfaces ethernet{interface} counters"
10]
11
12 # List of commands to clear interface counters
13 clearCountersCmds = [
14     "enable",
15     "clear counters ethernet{interface}"
16]
17
18 # Create log entry
19 ctx.log("Getting counters for Ethernet{interface}")
20
21 # Run the list of commands to get counters
22 cmdResponse = ctx.runDeviceCmds(showCountersCmds)
23
24 # Create a log entry of the counters
25 ctx.log(f"The current interface counters are : {cmdResponse[1]['response']}")
26
27 # Create log entry prior to clearing the counters
28 ctx.log("Clearing counters for Ethernet{interface}")
29
30 # Run the list of commands to clear counters
31 cmdResponse = ctx.runDeviceCmds(clearCountersCmds)
32
33 # Create log entry
34 ctx.log(f"Counters for Ethernet{interface} Successfully Cleared")
35
36 # Create log entry
37 ctx.log("Getting counters for Ethernet{interface}")
38
39 # Run the list of commands to get counters
40 cmdResponse = ctx.runDeviceCmds(showCountersCmds)
41
42 # Create a log entry of the counters
43 ctx.log(f"The current interface counters are : {cmdResponse[1]['response']}")

```

To the right of the script, there's an 'Execute' button. Above the execute button, there are dropdown menus for 'Dynamic Argument Values' (Interface set to 1) and 'DeviceID' (set to Leaf-1A). There's also a 'Select' button.

26. Let's use Ethernet1 for this test, so specify “1” for the “Interface.”

27. In the DeviceID dropdown, select “Leaf-1A”

28. Next, click the blue “Execute” button.

This screenshot is identical to the one above, but the 'DeviceID' dropdown has been changed to 'Leaf-1A'. The rest of the interface, including the script content and the 'Execute' button, remains the same.

29. As the action executes, it will provide the logging output in the pane below the “Execute” button. The oldest log entry will be at the bottom, so you should read the logs from the bottom up. You’ll be able to see that the current interface counters were displayed, they were then cleared, and the new counters were displayed.

The screenshot shows the Arista CloudVision interface. On the left, there's a sidebar with various navigation options like Provisioning, Network Provisioning, Conflicts, Image Repository, Tasks, Actions, Change Control, Action Bundles, Templates, Studios, Workspaces, Snapshot Configuration, Public Cloud Accounts, Tags, and Zero Touch Provisioning. The main area is titled "Actions" and shows a specific action named "ASE - Clear Counters". The "Script" tab is selected, displaying the following Python code:

```

1 # Written by your-name-here
2
3 # Create a variable to reflect the "Interface" dynamic argument
4 interface = ctx.action.args.get("Interface")
5
6 # List of commands to show interface counters
7 showCountersCmds = [
8     "enable",
9     f"show interfaces {interface} counters"
10 ]
11
12 # List of commands to clear interface counters
13 clearCountersCmds = [
14     "enable",
15     f"clear counters {interface}"
16 ]
17
18 # Create log entry
19 ctx.alog(f"Getting counters for {interface}")
20
21 # Run the list of commands to get counters
22 cmdResponse = ctx.runDeviceCmds(showCountersCmds)
23
24 # Create a log entry of the counters
25 ctx.alog(f"The current interface counters are : {cmdResponse[1]['response']}")
26
27 # Create log entry prior to clearing the counters
28 ctx.alog(f"Clearing counters for {interface}")
29
30 # Run the list of commands to clear counters
31 cmdResponse = ctx.runDeviceCmds(clearCountersCmds)
32
33 # Create log entry
34 ctx.alog(f"Counters for {interface} Sucessfully Cleared")
35
36 # Create log entry
37 ctx.alog(f"Getting counters for {interface}")
38
39 # Run the list of commands to get counters
40 cmdResponse = ctx.runDeviceCmds(showCountersCmds)
41
42 # Create a log entry of the counters
43 ctx.alog(f"The current interface counters are : {cmdResponse[1]['response']}")

```

To the right, there's a "Dynamic Argument Values" section with "Interface" set to "1" and "DeviceID" set to "Leaf-1A". Below that is a "Logs" section with a search bar and a "Download All Logs" button. The logs show the execution of the script and the resulting log entries:

- HKJZNogfRIR2aIZeQrmLP 25 seconds ago: The current interface counters are : {'interfaces': {'Ethernet1': {'inBroadcastPkts': 0, 'inOctets': 0, 'lastUpdateTimestamp': 171770194.640033, 'outBroadcastPkts': 0, 'outDiscards': 0, 'outMulticastPkts': 0, 'outOctets': 2243680, 'outUcast... Expand}}
- HKJZNogfRIR2aIZeQrmLP 27 seconds ago: Getting counters for Ethernet1 Jun 6, 2024 13:13:13.614 GMT-6
- HKJZNogfRIR2aIZeQrmLP 27 seconds ago: Counters for Ethernet1 Sucessfully Cleared Jun 6, 2024 13:13:13.615 GMT-6
- HKJZNogfRIR2aIZeQrmLP 29 seconds ago: Clearing counters for Ethernet1 Jun 6, 2024 13:13:11.466 GMT-6
- HKJZNogfRIR2aIZeQrmLP 29 seconds ago: The current interface counters are : {'interfaces': {'Ethernet1': {'inBroadcastPkts': 1, 'inDiscards': 0, 'inMulticastPkts': 7, 'inOctets': 2892, 'inUcastPkts': 16, 'lastUpdateTimestamp': 1717701190.2773674, 'outBroadcastPkts': 0, 'outDiscards': 0, 'outMulticastPkts': 8, 'outOctets': 165612243, 'outUcast... Expand}}
- HKJZNogfRIR2aIZeQrmLP 31 seconds ago: Getting counters for Ethernet1 Jun 6, 2024 13:13:10.033 GMT-6

30. Now, your custom action is available to be used within a change control.

NOTE: Check out this document for more information on custom actions.

<https://www.arista.com/en/support/toi/cvp-2021-3-0/14901-ui-for-custom-action-scripts>

Arista also has a repo with some custom actions you might find useful. That repo can be accessed here: <https://github.com/aristanetworks/cloudvision-python-actions>