

TECHNICAL WHITE PAPER

VSF BEST PRACTICES FOR ARUBA CX 6300 SWITCH SERIES

AOS-CX VERSION 10.4

CONTENTS

Revision History	2
Purpose	3
VSF Deployment	3
General recommendations	3
Aruba CX mobile app provisioning	4
CLI provisioning	7
VSF split detection	8
VSF Maintenance and Troubleshooting	8
Split stack operations	8
Manual failover	9
Removing a VSF member	9
Renumbering VSF members	10
Replacing a VSF member with the same model	11
Replacing a VSF member with a different model	11
Software upgrades	12
Appendix: Timings	13
6300 VSF stack (3 members)	13
Supported Platforms	14

REVISION HISTORY

Document Version	Reason for Change	Revision Date
1.0 (Initial Release)		Dec 2019
1.1	Added instructions for loading a custom template during mobile app provisioning	Dec 2019

PURPOSE

Combining multiple switch chassis into a single logical device can provide increased capacity and improved redundancy for access and distribution workloads, while simplifying configuration and providing a single point of management access. The Aruba CX 6300 Switch Series provides front plane stacking using the Virtual Switching Framework (VSF) feature, utilizing the four front panel SFP ports operating at 10G, 25G, or 50G speeds.

This document contains guidance and best practices for deploying and maintaining 6300 VSF stacks using the latest release of the AOS-CX network operating system.

VSF DEPLOYMENT

The 6300 Switch Series supports the ability to use front panel switch ports to link switches in a ‘virtual’ stack using VSF, with these links being used for intra-stack traffic forwarding and state synchronization between VSF members. A 6300 VSF stack may contain up to 10 members in a *ring* or *chain* topology, supporting any possible combination of 6300F and 6300M switch models.

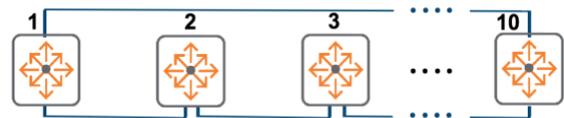
While VSF on AOS-CX is conceptually similar to its AOS-Switch implementation on the 2930F and 5400R switch families, there are a number of substantial implementation and operational differences, which are explained in this document.

General recommendations

To maximize available VSF link bandwidth, it is recommended to use Aruba 50G DAC cables for VSF links. Multiple ports can be aggregated to increase VSF link bandwidth and provide redundancy; for a two-member stack, all four SFP ports can be utilized in a single VSF link for up to 200Gbps of VSF link bandwidth (bidirectional), while larger stacks (up to 10 members) can use up to 2 ports per link for up to 100Gbps of bandwidth.

All VSF link ports in a stack must operate at the same speed (10G, 25G, or 50G). However, VSF links do not all need to use the same *number* of ports; for example, the VSF link between members 1 and 2 may use two SFP ports on each member while the link between members 2 and 3 uses only one SFP port per member. Using less than two ports for specific VSF links may be desirable on members where SFP ports are also being used for stack uplinks. The only limitation this imposes on a stack is reduced VSF link bandwidth between those members using links with only one port assigned per member.

A *ring* is the recommended topology, as it provides redundant links to all members to prevent a failure of any single VSF link or member from causing a stack split. A ring is constructed by configuring two VSF links on all members, and connecting them to provide a continuous path between both ‘ends’ of the stack.



A *chain* topology provides only a single path between any two members of the stack, and should only ever be used as a transitional state when building a stack or when recovering a ring from a link or member failure. A failure of any member or link in the middle of the chain will result in a stack split, a situation described in more detail later in this document.



A *secondary* member should always be defined to assume the VSF-Standby role, and the out-of-band management (OOBM) ports on the *primary* and *secondary* members should be connected to each other — either directly or through a dedicated management network — in order to utilize VSF split detection, which should always be enabled.

For maximum stack resiliency, the *primary* and *secondary* members should be 6300M models with redundant power supplies connected to different circuits, in order to minimize the probability that a single-source power failure will disable both the stack master and standby.

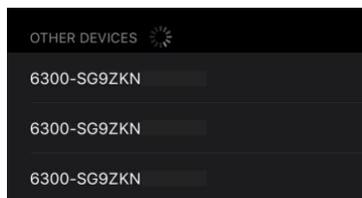
Aruba CX mobile app provisioning

Provisioning a 6300 VSF stack using the Aruba CX mobile app requires a supported Android or iOS device, the latest version of the Aruba CX app (version 2.0 or later), and the USB to Bluetooth adapter included in the box with each 6300 switch model.

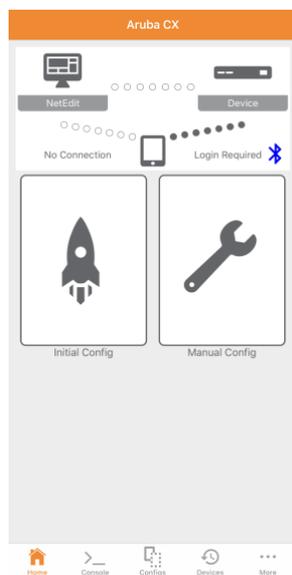
To start the provisioning process, first ensure that all members of the stack are powered up and fully booted in a factory default state, all cables are connected to the ports to be used for VSF links, and the USB to Bluetooth adapter is plugged into the front panel USB-A port on each member.

Note: Spanning Tree is enabled by default on the 6300 Switch Series, which will prevent a loop from forming when VSF link cables are connected prior to the stack being fully provisioned.

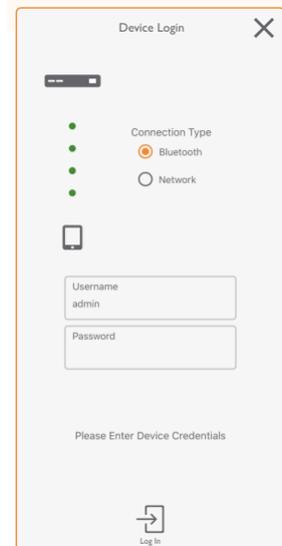
On the mobile device with the Aruba CX mobile app installed, first use Bluetooth to discover and connect to the switch that will be the stack primary; each switch should show up in the device list using the format *6300-SERIAL_NUMBER*.



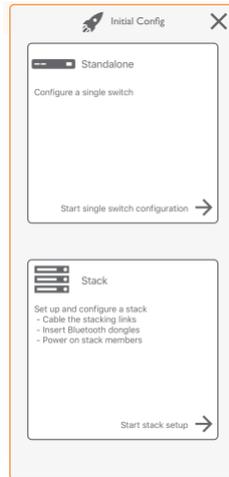
Once the device is connected to the switch, launch the Aruba CX app. Within a few seconds, the app should display an active Bluetooth connection to the switch with the message **Login Required**. Tap the **Initial Config** button to start the stack configuration.



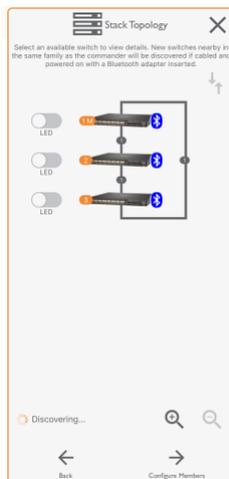
On the **Device Login** screen, leave the **Connection Type** as **Bluetooth**. Enter the username **admin** and leave the password field blank, then tap the **Log In** button at the bottom of the screen.



Next, on the Initial Config screen, tap the **Stack** button to begin stack setup.



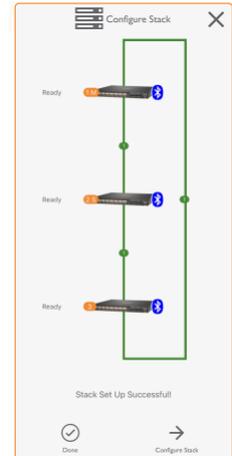
The app will automatically discover all 6300 switches that are connected to the primary via VSF link cables, connect to them via Bluetooth, and will display them in the topology view on the screen. Enabling the **LED** switch for each switch causes the blue UID LED on its front panel to flash; use them to verify that the physical stack layout matches the displayed topology. To change member IDs or assign a member as the stack secondary, tap the switch in the topology view.



With the stack member selected, you can either change the member ID number using the dropdown selector, or designate it as the stack secondary using the checkbox. Once you are done editing the member, tap the **Back** button at the bottom of the screen. After all members are assigned their settings, tap **Configure Members** to continue.



The app will then apply member IDs and secondary configuration to all switches in the stack, which will cause each member other than the primary to reboot. Once all switches have rebooted and joined the stack, the message **Stack Set Up Successful!** will be displayed; tap **Configure Stack** to continue.



If NetEdit will be used to manage the stack, enter the NetEdit server address, username, and password, then tap **Log In**; if not, then tap **Skip**.



Choose the desired switch management interface from the dropdown menu; configure the stack hostname, admin password, and management IP interface (static or DHCP); tap **Next** to continue.



Alternately, you may deploy a custom configuration template (saved on your device or available on a connected file sharing service such as OneDrive, Dropbox, or iCloud Drive) by tapping the interface dropdown menu, and selecting **Import Custom Template...**

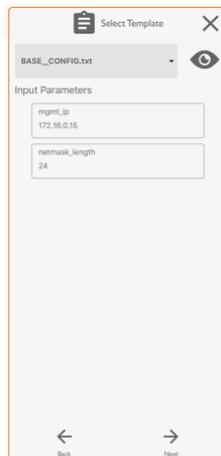


To load the custom template file onto your device, see the appropriate [Apple Support](#) or [Android Help](#) article for your device's operating system.

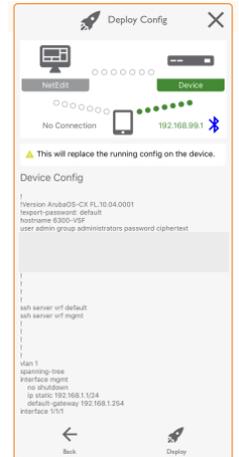
Select the desired configuration template from the file browser, then choose the imported template from the menu.



If the template requires any device-specific parameters, enter the appropriate values and then tap **Next** to continue.



Review the configuration generated by the app or imported from a template; then, tap **Deploy** to apply the configuration to the stack.



Once the configuration has been successfully deployed, the connection between the device and the switch (now the stack primary) will turn green, and the message **Device Deployment Successful!** will be displayed. Tap **Done** to return to the app's main page.



CLI provisioning

For this provisioning method, start with all prospective stack members powered up, with a management console session open (SSH, Telnet, USB-C, or Bluetooth) to the desired *primary* member (ID 1), and all desired VSF link cables connected. As with the Aruba CX mobile app provisioning process, Spanning Tree will prevent loops on the VSF link ports while the stack is being provisioned.

Start by configuring VSF link 1 on the primary switch:

```
switch(config)# vsf member 1
switch(vsf-member-1)# link 1 1/1/49
```

If the stack will contain more than two members, configure VSF link 2 as well:

```
switch(vsf-member-1)# link 2 1/1/50
```

Next, specify the desired *secondary* member ID:

```
switch(config)# vsf secondary-member 2
This will save the configuration and reboot the specified switch.
Do you want to continue (y/n)? y
```

Note: Performing this step *before* adding the desired secondary to the stack eliminates the need for the extra reboot that would occur with an operating stack member; that switch will instead automatically become the secondary when it reboots to join the stack and will assume the VSF-Standby role.

Connect to the next stack member and configure VSF links (one if a 2-member stack, two if more than 2 members). Remember that all stack members start with a member ID of 1, so keep that in mind when assigning ports to the VSF links.

```
switch(config)# vsf member 1
switch(vsf-member-1)# link 1 1/1/49
switch(vsf-member-1)# link 2 1/1/50
```

Once VSF links have been configured on the new member, renumber it to the desired member ID:

```
switch(config)# vsf renumber-to 2
This will save the VSF configuration and reboot the switch.
Do you want to continue (y/n)? y
```

The member will automatically join the stack with its new member ID when it reboots; if it has been designated the secondary, it will also assume the VSF-Standby role.

If a new stack member is running a software version different from that on the VSF-Master, during the boot process it will automatically download the primary and secondary software images from the VSF-Master and reboot a second time.

For stacks of 3 members or greater, when the last member has been added, remember to connect the final VSF link between the ends of the stack to complete the ring topology.

Once stack provisioning is complete, verify the final stack status and topology using the following commands:

```
switch# show vsf
```

```
MAC Address      : e0:07:1b:00:00:01
Secondary       : 2
Topology        : Ring
Status          : No Split
Split Detection Method : mgmt
```

Id	Mac Address	Type	Status
1	e0:07:1b:00:00:01	JL660A	Master
2	e0:07:1b:00:00:02	JL661A	Standby
3	e0:07:1b:00:00:03	JL665A	Member

```
switch# show vsf topology
```

```
Mstr          Stdbby
+----+      +----+      +----+
| 1 |1==2| 3 |1==2| 2 |
+----+      +----+      +----+
  2                1
+=====+
```

VSF split detection

In order to mitigate the effects of a VSF split, a split-detection (also known as Multi-Active Detection, or MAD) method should be configured and utilized. The 6300 Switch Series supports using a connection between the OOBM ports on the primary and secondary members to detect when a split has occurred. The OOBM ports can either be directly connected using an Ethernet cable, or connected to a common Layer 2 broadcast domain via a dedicated OOBM switch or network. In the latter case, the OOBM ports on the primary and secondary can be assigned a global stack OOBM IP address and used normally for remote management.

To enable VSF split detection using the OOBM ports:

```
switch(config)# vsf split-detect mgmt
```

For more details on how split detection operates in a 6300 VSF stack, see **Split stack operations** in the following section.

VSF MAINTENANCE AND TROUBLESHOOTING

Split stack operations

A VSF stack operating as a chain may experience a split stack condition due to the failure of a VSF link or stack member. When a split occurs, the behavior of each resulting stack fragment depends on where the split occurred, whether a secondary member has been designated for the stack, and whether split detection has been enabled.

The behaviors described in the following scenarios presume that VSF split detection has been configured and is functional, and a secondary member has been designated and is operating in the VSF-Standby role.

Scenario A — If a VSF *link* fails and causes a split, but all stack members remain operational, the following occurs:

- The fragment containing the primary (operating as VSF-Master) remains active
 - If the secondary is in a *separate* fragment from the primary, it assumes the VSF-Master role for the fragment, and that fragment is designated *inactive* (all links *except* the OOBM port and VSF link ports are disabled)
 - If the secondary is in the *same* fragment as the primary, all members in the *other* fragment reboot and stay down until a connection to a VSF-Master is re-established

Scenario B — If a VSF *member* fails and causes a split, the following occurs:

- If the failed member was the *primary*, the secondary assumes the VSF-Master role for its fragment and that fragment remains active; all members in the other fragment reboot and stay down until a connection to a VSF-Master (either primary or secondary) is re-established
 - When the primary recovers, it will assume the VSF-Standby role until a subsequent failover occurs, and all other stack members in the isolated fragment will recover and rejoin the stack
- If the failed member was the *secondary*, the fragment containing the primary remains active; all members in the other fragment reboot and stay down until a connection to a VSF-Master is re-established
 - When the secondary recovers, it will reassume the VSF-Standby role, and all other stack members in the isolated fragment will recover and rejoin the stack
- If the failed member was *neither* a primary nor secondary, stack fragment behavior is the same as for a failed VSF link as described in Scenario A

Manual failover

There are a few situations in which you might wish to manually trigger a VSF-Master failover: the primary is to be removed from the stack and replaced; testing for network behavior and potential performance issues during initial deployment; or having the primary reassume the VSF-Master role after a previous failover event.

The preferred method for initiating a manual failover is to use the following command:

```
switch# vsf switchover
```

This will trigger a reboot of the VSF-Master, resulting in the VSF-Standby assuming the VSF-Master role. Once the former VSF-Master reboots, it will rejoin the stack as VSF-Standby.

Removing a VSF member

To remove a VSF stack member, use the following command (replace the member ID with the desired value):

```
switch(config)# no vsf member 4  
The specified switch will be unconfigured and rebooted  
Do you want to continue (y/n)? y
```

This command will remove all stack configuration from the VSF-Master for the specified member. If that member is currently operating as part of the stack, it will be rebooted and restored to factory defaults.

Attempting to remove the stack primary with this command without an operating VSF-Standby will result in the following warning:

```
switch(config)# no vsf member 1  
Unconfiguring the primary switch of the stack without a standby  
will make the stack unusable  
Do you want to continue (y/n)?
```

If an operating member other than the primary or secondary is disconnected from the stack *without* being properly removed from the stack, it will reboot itself then stay down as it attempts to reconnect to the VSF-Master. (This will also occur to *all* stack members with the VSF-Member role if the stack primary is removed using the above command without a secondary member defined and present.)

To restore an isolated VSF-Member switch to factory defaults, you will need to use one of the following two methods via a connection to the USB-C console port on the disconnected stack member:

Method 1: Interrupt the switch boot process using the **Control-C** key combination; this will cause a redirect to an emergency login prompt. Once you have logged in (using credentials from the stack master the member was last connected to), use the following command to reset the switch to factory defaults:

```
switch# vsf-factory-default
```

This command will *immediately* reboot the switch, restoring it to a factory default state with a member ID of 1.

Method 2: Reboot the switch via the front panel Reset push button or via a power-cycle, and select the ServiceOS option from the boot menu when prompted. Login to the ServiceOS using the username **admin**; then, from the SVOS prompt, enter the following command:

```
SVOS> erase zeroize
#####WARNING#####
This will securely erase all customer data and reset the switch
to factory defaults. This will initiate a reboot and render the
switch unavailable until the zeroization is complete.
This should take several minutes to one hour to complete.
#####WARNING#####

Continue (y/n)? y
```

The switch will reboot for zeroization, and will reboot once more after zeroization is complete, at which point it will boot to a factory default state.

Renumbering VSF members

As of the AOS-CX 10.4 release, in order to renumber *operating* VSF members, remove each member to be renumbered from the stack, reconfigure their VSF links, then renumber them to the correct member IDs.

Note: The `vsf renumber-to` command can only be run from the global configuration context on a switch operating in the VSF-Master role with a member ID of 1.

Before performing these steps, back up the VSF configuration for use when reapplying member-specific settings after renumbering is complete.

To back up the stack running configuration in CLI format to an external SFTP server, use the following command as an example:

```
switch# copy running-config sftp://sftpuser@10.10.10.1/6300_running_config.txt cli vrf mgmt
```

In this example, the switches acting as stack members 3 and 4 were connected to the stack in the wrong order during initial provisioning, and need to be renumbered to reflect their physical positions in their rack.

Start by removing the stack members to be renumbered from the configuration on the master; in this example, members 3 and 4 will have their member IDs swapped:

```
switch(config)# no vsf member 3
The specified switch will be unconfigured and rebooted
Do you want to continue (y/n)? y
switch(config)# no vsf member 4
The specified switch will be unconfigured and rebooted
Do you want to continue (y/n)? y
```

Next, connect to the USB-C console port on each removed member, and configure their VSF links using the same physical ports originally assigned (in this example, the same commands are run on *both* members):

```
switch(config)# vsf member 1
switch(vsf-member-1)# link 1 1/1/49
switch(vsf-member-1)# link 2 1/1/50
```

Attach VSF link cables if not left connected, then renumber each member to the correct ID:

```
switch(config)# vsf renumber-to 3
This will save the VSF configuration and reboot the switch.
Do you want to continue (y/n)?
```

```
switch(config)# vsf renumber-to 4
This will save the VSF configuration and reboot the switch.
Do you want to continue (y/n)?
```

If the members' VSF link cables were left connected, each member will rejoin the stack as they reboot; otherwise, they will automatically reboot to join the stack with their new ID when the cables are later reconnected.

Finally, once both members are operating with the correct member IDs, re-apply their member-specific configuration from the backup created earlier, ensuring that the applied configuration is corrected for the new member IDs, any port number differences, and PoE settings.

Replacing a VSF member with the same model

Note: As of the AOS-CX 10.4 release, configuration of a replacement VSF member must be performed via the command line interface.

Replacing a VSF stack member with a switch of the same type (part number) does *not* require removing the existing member from the stack configuration on the master. If the member to be replaced is currently operating as part of the stack, ensure that the stack is operating in a ring topology, then remove the member from service by powering it down. If the stack is operating as a chain due to a previously failed link or member, ensure that the environment can tolerate a temporary split stack condition prior to taking the member out of service.

Once the member has been taken out of service and removed from the stack, configure the replacement member's VSF links, attach VSF link cables, and renumber it to match the existing member's ID. The replacement member will reboot with its new member ID, join the stack, and will assume the previous member's configuration.

Replacing a VSF member with a different model

Note: As of the AOS-CX 10.4 release, configuration of a replacement VSF member must be performed via the command line

interface.

If the member is to be replaced with a *different* model, the member ID must first be removed from the configuration (see **Removing a VSF Member**) before the replacement is added. Failure to do so will result in the replacement member being unable to join the stack when it is renumbered. If pre-provisioning the replacement member's configuration, the following error message will be displayed if the original member has not first been removed:

```
switch(config)# vsf member 3
switch(vsf-member-3)# type j1663a
Cannot change the part number for a configured member. Unconfigure the member first.
```

Once the member to be replaced has been removed from the VSF configuration, configure VSF links on the replacement switch, renumber it using the original member's ID, and connect VSF link cables. Once the replacement has joined the stack, apply any member-specific configuration, adjusting for differences in port numbering and PoE settings.

Software upgrades

Upgrading the software version on a 6300 stack requires loading the software image onto the VSF-Master, then rebooting the stack into the new version. Once the image has been copied, it will automatically be synchronized to all stack members without requiring operator action.

```
switch# copy sftp://sftpuser@10.10.10.1 FL_10_04_0003.swi primary vrf mgmt
```

Once the image has been copied and written to flash, reboot the stack into the new software version using this command:

```
switch# boot system primary
```

The entire VSF stack will simultaneously reboot to load the new software image.

APPENDIX: TIMINGS

Values are in minutes and seconds; all timings are approximate and may vary with software version, features configured, hardware variations, etc.

6300 VSF stack (3 members)

Operation	Estimated time to complete
Stack cold boot (power applied to ports up on all members)	3:39
Failover with standby re-sync (ports down to ports up)	3:12

SUPPORTED PLATFORMS

Virtual Switching Framework (VSF) is supported on the following Aruba CX switch platform:

- Aruba 6300 Switch Series