

# Ansible for Dell EMC Networking Switches

Ansible playbook for leaf-spine architecture

Dell EMC Networking Solutions Engineering  
August 2017

## Revisions

Date	Revision	Description	Author
Aug 2017	1.0	Initial Release	Sambhu Kalaga, S Harish Koushik

THIS WHITE PAPER IS FOR INFORMATIONAL PURPOSES ONLY, AND MAY CONTAIN TYPOGRAPHICAL ERRORS AND TECHNICAL INACCURACIES. THE CONTENT IS PROVIDED AS IS, WITHOUT EXPRESS OR IMPLIED WARRANTIES OF ANY KIND.

Copyright © 2017 Dell Inc. All rights reserved. Dell and the Dell EMC logo are trademarks of Dell Inc. in the United States and/or other jurisdictions. All other marks and names mentioned herein may be trademarks of their respective companies.

# Table of contents

Revisions.....	2
Table of contents .....	3
1 Introduction.....	4
1.1 Dell EMC Networking Ansible solutions .....	4
1.2 Dell EMC Networking Ansible roles .....	5
1.2.1 Structure of Dell EMC Networking Ansible roles .....	6
1.2.2 Tasks .....	7
1.2.3 Templates .....	8
1.2.4 Variables .....	9
1.3 Dell EMC Networking Ansible Roles installation .....	10
1.4 Basic Dell EMC Networking switch configurations .....	11
1.5 RSA-SSH authentication passwordless login.....	12
2 Playbooks.....	14
2.1 Inventory .....	14
2.2 Playbook .....	15
2.3 Ansible Playbook execution.....	16
3 Leaf-spine configuration through Ansible.....	17
3.1 Introduction to the leaf-spine architecture .....	17
3.2 Ansible configuration for leaf-spine architecture .....	18
3.2.1 Inventory File .....	25
3.2.2 Playbook .....	32
3.2.3 Dell EMC Networking Ansible role: Link Aggregation Group (LAG).....	33
3.2.4 Dell EMC Networking Ansible role: interface.....	35
3.2.5 Dell EMC Networking Ansible role: VLT .....	38
3.2.6 Dell EMC Networking Ansible Networking role: prefix-list.....	38
3.2.7 Dell EMC Networking Ansible role: BGP .....	41
3.2.8 Dell EMC Networking Ansible role: ECMP .....	43
3.2.9 Execution of Dell leaf-spine playbook .....	43
A Additional Information.....	54
B Support and Feedback .....	55

# 1 Introduction

The goal of this document is to provide readers insight into the configuration of Dell EMC Networking switches using Dell EMC Networking Ansible roles. The Dell EMC Networking Ansible roles facilitate the provisioning of devices running Dell EMC Networking Operating Systems. Dell EMC Networking Ansible roles support configuration of all three Dell EMC Networking operating systems, i.e. OS6, OS9 and OS10. However, this document focuses on providing Ansible configuration examples for Dell EMC Networking OS9 switches, specifically for leaf-spine architectures.

[Section 1.2](#) describes the structure of the Dell EMC Networking Ansible roles by providing an example of the dellos-interface role. [Section 2](#) introduces the creation and execution of a basic playbook using Ansible. [Section 3](#) discusses the configuration of a leaf-spine architecture using Dell EMC Networking Ansible Roles.

## 1.1 Dell EMC Networking Ansible solutions

Dell EMC Networking Solutions are based on an open ecosystem, enabling organizations to choose from industry-standard network applications, network operating systems, and network hardware. These open networking solutions enable IT managers to build an application-agnostic infrastructure and simplify data center management with standard automation tools, like Ansible and standards-based open platforms.

The use of Ansible to provision and manage Dell EMC Networking switches enables rapid new device deployment and network configuration changes. It also enables the tracking of running network device configurations against a known baseline for both Dell EMC Networking and third-party operating systems.

The Ansible modules for Dell EMC Networking Solutions allow organizations to reduce the time and effort required to design, provision, and manage networks in the following ways:

- **Agentless:** No new software to install on your switches.
- **Powerful:** End-to-end automation of the configuration of bare metal switches using the Dell EMC Networking Open Automation framework.
- **Easy:** Dell EMC Networking modules ship with the Ansible distribution. There is nothing extra to install.
- **Best Practice:** Leverages CLI user authentication to centralize and monitor session management.
- **Flexible:** Full support of the Ansible model for networking: command, configuration and facts.

With the introduction of Ansible 2.2, Ansible core supports Dell EMC Networking modules. Use these modules to manage and automate Dell EMC Networking switches running Dell EMC Networking OS6, OS9 and OS10. All modules are currently executed in local connection mode using the CLI and SSH(Secure Shell) transport. The following Dell EMC Networking modules are integrated into the Ansible core:

- OS6 Modules
  - dellos6\_command - Run commands on remote devices running OS6
  - dellos6\_config - Manage OS6 configuration sections
  - dellos6\_facts - Collect facts from remote devices running OS6
- OS9 Modules
  - dellos9\_command - Run commands on remote devices running OS9

- dellos9\_config - Manage OS9 configuration sections
- dellos9\_facts - Collect facts from remote devices running OS9
- OS10 Modules
  - dellos10\_command - Run commands on remote devices running OS10
  - dellos10\_config - Manage OS10 configuration sections
  - dellos10\_facts - Collect facts from remote devices running OS10

For additional details about the Dell EMC Networking modules, see the [Ansible Network Module documentation](#).

Dell EMC Networking modules have been packaged and distributed through Ansible Galaxy. Ansible Galaxy is a hub for finding, reusing and sharing the Ansible content. Galaxy provides pre-packaged units known in Ansible as roles.

## 1.2 Dell EMC Networking Ansible roles

Ansible roles allow you to automatically load certain files for variables, tasks and handlers based on a known file structure. Grouping content by roles allows the roles to be easily shared with other users. These roles are abstracted for OS6, OS9 and OS10. The following Dell EMC Networking Ansible Roles are currently available in [Ansible Galaxy](#):

- **dellos-acl:**  
The dellos-acl role facilitates the configuration of an access control list (ACL). It supports the configuration of different types of ACLs (standard and extended) for both IPv4 and IPv6, and assigns the access-class to the line terminals. Currently, this role is only supported for OS9.
- **dellos-bgp:**  
The dellos-bgp role facilitates the configuration of Border Gateway Protocol (BGP) attributes. It supports the configuration of router ID, networks, neighbors, maximum path and other BGP attributes.
- **dellos-dcb:**  
The dellos-dcb role facilitates the configuration of Data Center Bridging (DCB). It supports the configuration of DCB map and DCB buffer and assigns them to interfaces. Currently, this role is only supported for OS9.
- **dellos-interface:**  
The dellos-interface role facilitates the configuration of interface attributes. It supports the configuration of administrative state, description, MTU, IP address, IP helper, port mode and other interface attributes.
- **dellos-lag:**  
The dellos-lag role facilitates the configuration of Link Aggregation Group (LAG) attributes. It supports the creation and deletion of a LAG and its member ports. It also supports the configuration of type (static/dynamic), hash scheme, minimum required link and other lag attributes.

- **dellos-system:**  
The dellos-system role facilitates the configuration of global system attributes. It supports the configuration of hostname, SNMP server, NTP server, logging servers, management route and CLI users.
- **dellos-vlan:**  
The dellos-vlan role facilitates configuration of virtual LAN (VLAN) attributes. It supports the creation and deletion of a VLAN and its member ports.
- **dellos-vlt:**  
The dellos-vlt role facilitates the configuration of Virtual Link Trunking (VLT) attributes. It supports the creation and deletion of VLT links LAG and VLTi link. Currently, this role is only supported for OS9.
- **dellos-vrf:**  
The dellos-vrf role facilitates configuration of virtual routing and forwarding (VRF) that helps in the partition of physical routers to multiple virtual routers. Currently, this role is only supported for OS9.
- **dellos-xstp:**  
The dellos-xstp role facilitates the configuration of xSTP attributes. It supports multiple versions of Spanning Tree Protocol (STP): Rapid Spanning Tree (RSTP), Rapid Per-VLAN Spanning Tree (Rapid PVST+), Multiple Spanning Tree (MST) and Per-VLAN Spanning Tree (PVST). It supports the configuration of bridge priority, enabling and disabling spanning tree, creating and deleting instances and mapping virtual LANs (VLANs) to instances.
- **dellos-ecmp:**  
The dellos-ecmp role facilitates the configuration of ECMP group attributes in devices running Dell EMC Networking Operating Systems.
- **dellos-prefix-list:**  
The dellos-prefixlist role facilitates the configuration of prefix list attributes in devices running Dell EMC Networking Operating Systems.

Visit [Dell EMC Networking Ansible Roles](#) to get more detailed information and examples.

### 1.2.1 Structure of Dell EMC Networking Ansible roles

This section explains the basic file structure of Dell EMC Networking Ansible roles. In the following sections, the Dell-Networking.dellos-interface role structure is explained in detail. All the Dell EMC Ansible Networking roles follow the same file structure, which is defined by Ansible Galaxy.

The detailed structure of the Dell EMC Dell-Networking.dellos-interface role is shown below.

```
$ sudo tree Dell-Networking.dellos-interface
Dell-Networking.dellos-interface
|-- README.md
|-- defaults
|   |-- main.yml
|-- handlers
|   |-- main.yml
|-- meta
|   |-- main.yml
|-- tasks
|   |-- main.yml
|-- templates
|   |-- dellos10_interface.j2
|   |-- dellos6_interface.j2
|   |-- dellos9_interface.j2
|-- tests
|   |-- inventory
|   |-- main.yml
|   |-- test.yml
`-- vars
    |-- main.yml
```

7 directories, 12 files

- The README file provides details about the installation, requirements, different role variables, connection variables, dependencies, licenses and examples of a playbook.
- If dellos-interface/tasks/main.yml exists, then tasks listed therein will be added to the play.
- If dellos-interface/handlers/main.yml exists, then handlers listed therein will be added to the play.
- If dellos-interface/meta/main.yml exists, then any role dependencies listed therein will be added to the list of roles.
- If dellos-interface/vars/main.yml exists, then variables listed therein will be added to the play.
- A copy, script, template or include tasks (in the role) can reference files in dellos-interface/{*templates, tasks etc.*} without having to path them relatively or absolutely. For example, if a playbook calls for the dellos-interface role, then all the tasks, handlers, variables, templates are added to the play and executed.

## 1.2.2 Tasks

During the execution of an Ansible playbook, when a particular role is called, all the tasks within the role are added to the play. As mentioned in the previous section, for the **OS9** operating system, the dellos9\_interface.j2 template file is executed and rest of the tasks are skipped. If the operating system selected is **OS10** or **OS6**, then the corresponding jinja2 template files are selected for execution, i.e. dellos10\_interface.j2 or dellos6\_interface.j2. A sample interface template file is shown below.

```
---
- name: "Provision Interface configuration for OS9"
  dellos9_config:
```

```

        src: dellos9_interface.j2
        provider: "{{ cli }}"
#   notify: save config os9
    when: ansible_net_os_name == "dellos9"
    register: output

- name: "Provision Interface configuration for OS10"
  dellos10_config:
    src: dellos10_interface.j2
    provider: "{{ cli }}"
#   notify: save config os10
  when: ansible_net_os_name == "dellos10"
  register: output

- name: "Provision Interface configuration for OS6"
  dellos6_config:
    src: dellos6_interface.j2
    provider: "{{ cli }}"
#   notify: save config os6
  when: ansible_net_os_name == "dellos6"
  register: output

```

### 1.2.3 Templates

All Dell EMC Networking Ansible role templates are developed in the Jinja2 format. Jinja2 is a modern and designer-friendly templating language for Python. Jinja2 templating is used to enable dynamic expressions and access to variables.

**Note:** The templates keep evolving as new updates and features are added to the role. Please read the README files for complete details about the updates and new features.

All the templating occurs on the Ansible controller before the task is sent and executed on the target machine. A sample jinja2 template for dellos9-interface.j2 is shown below.

```

#jinja2: trim_blocks: True,lstrip_blocks: True
{#####
Purpose:
Configure interfaces on OS9 Devices.

dellos_interface:
  OS9:
    fortyGigE 0/36:
      desc: "OS9 intf"
      portmode: hybrid
      mtu: 2000
      switchport: False

```



```

    admin: up
    ip_and_mask: "192.168.13.1/24"
    suppress_ra: present
    ip_type_dynamic: true
    ipv6_and_mask: 2001:4898:5808:ffa2::9/126
    ipv6_reachabletime: 60000
    ip_helper:
        - ip: 10.0.0.33
          state: present
    class_vendor_identifier: present
    option82: true
    remote_id: hostname
    fanout: True

#####}
{% if (dell_interface is defined and dell_interface) and (hostname in
dell_interface) %}
{% set vars = dell_interface[hostname] %}
{% if vars %}
{% for key in vars.keys() %}
{% set intf_vars = vars[key] %}

{% set intf = key.split(" ") %}
{% set port = intf[1].split('/') %}
{% if intf_vars.fanout is defined %}
    {% if intf_vars.fanout %}
stack-unit {{ port[0] }} port {{ port[1] }} portmode quad
    {% else %}
no stack-unit {{ port[0] }} port {{ port[1] }} portmode quad
    {% endif %}
{% endif %}

{% if (intf_vars.fanout is defined and not intf_vars.fanout) or
(intf_vars.fanout is not defined)%}
interface {{ key }}
    {% if intf_vars.desc is defined %}
    {% if intf_vars.desc %}
description {{ intf_vars.desc }}
    {% else %}
no description
    {% endif %}

```

## 1.2.4 Variables

Variables are mainly used to define the switch configurations in Ansible. Many Dell EMC Networking switches have common configurations. There may also be differences in configurations between the Dell EMC Networking switches based on their topological location. Variables are used to define these common

configurations through the use of group variable files or differences in configurations through host variable files. Variables are also used as part of playbook definitions, command line arguments, inventory definitions etc. For more inputs on variables, please go through the Ansible documentation.

**Note:** The common configuration variables are at vars/main.yml file. All host-specific configurations are in the host\_vars/<host\_name>.yml configuration file. Please read the README files for complete details about the feature 'keys' to be used for configurations. The supported key:value pairs with examples are listed in the README files.

The var/main.yml file in the Dell EMC Networking Ansible roles is used to define the common interface variables for each host. For example, if interface fo1/25 is a layer 2 interface in all switches, it is a common configuration variable and can be placed in vars/main.yml as shown below.

```
---
# vars file for Dell-Networking.dellos-interface
dellos_interface:
  Fo 1/25:
    admin: up
    desc: "Rack 1 Leaf-A Connection"
    switchport: True
    portmode: "hybrid"
```

The host\_vars/<host\_name>.yml file is used to define the variables specific to a host. For example, the ip address on interface fo 1/26 is specific to a host and it needs to be placed in host\_vars/<host\_name>.yml. A sample interface host specific configuration in host\_vars/Leaf1A.yml is shown below.

```
---
hostname : Leaf1A
cli :
  host : "{{ ansible_host }}"
  username : admin
  password : xxxx
  authorize : yes
  auth_pass : false
  transport : cli
dellos_interface:
  Fo 1/26:
    admin: up
    desc: "Rack 1 Leaf-1A Connection"
    ip_and_mask: 192.168.1.2/30
```

## 1.3 Dell EMC Networking Ansible Roles installation

Ansible can be installed on any Linux machine with Python 2.6 or 2.7 installed. Linux flavors include Red Hat, Debian, Ubuntu, CentOS, OS X, any of the BSDs and so on. Install Ansible 2.2 or later to use the Dell EMC Networking Ansible modules.

In the example below, Ansible 2.2 was installed on Ubuntu 16.04 by following the steps listed below:

1. Add the Ansible PPA to Ubuntu 16.04 by executing the following command:

```
$ sudo apt-add-repository ppa:ansible/ansible
```

2. Refresh the system's package index by executing the following command:

```
$ sudo apt-get update
```

3. Install the Ansible package by executing the following command:

```
$ sudo apt-get install ansible
```

4. Install the Dell EMC Networking Ansible roles, by executing the following command:

```
$ sudo ansible-galaxy install Dell-Networking.dellos-bgp
- downloading role 'dellos-bgp', owned by Dell-Networking
- downloading role from https://github.com/Dell-Networking/ansible-role-dellos-bgp/archive/v1.0.5.tar.gz
- extracting Dell-Networking.dellos-bgp to /etc/ansible/roles/Dell-Networking.dellos-bgp
- Dell-Networking.dellos-bgp was installed successfully
```

5. To install more than one Dell EMC Networking Ansible Roles, execute the following commands:

```
$ sudo ansible-galaxy install -r dellemc_roles.txt
$ cat dellemc_roles.txt
Dell-Networking.dellos-bgp
Dell-Networking.dellos-interface
Dell-Networking.dellos-lag
Dell-Networking.dellos-vlan
```

## 1.4 Basic Dell EMC Networking switch configurations

To configure the Dell EMC Networking switches using Dell EMC Networking Ansible roles, the following commands should be executed first (for OS9).

1. Create a username and password for Ansible.

```
Leaf1A(conf)#username ansible password xxxx privilege 15
```

2. Configure the management interface (static or dynamic ip address).

```
Leaf1A(conf)#interface managementethernet 0/0
Leaf1A(conf-if-ma-0/0)#ip address 100.67.170.31/24
Leaf1A(conf-if-ma-0/0)#no shutdown
Leaf1A(conf-if-ma-0/0)#exit
```

3. Enable the SSH server on the switch.

```
Leaf1A(conf)#ip ssh server enable
```

```
Leaf1A(conf)#ip ssh connection-rate-limit 60
```

## 1.5 RSA-SSH authentication passwordless login

SSH is required between the Ansible control machine (Ubuntu 16.04) and the Dell EMC Networking switches for executing the Dell EMC Networking Ansible Roles. Dell EMC Networking switches support SSH version 2.0 in combination with RSA (Rivest-Shamir-Adleman) encryption for authentication. RSA-SSH authentication permits the user to establish a SSH session without entering a password. This allows the user using an Ansible script to automatically launch a SSH session to a Dell EMC Networking switch, eliminating the potential security vulnerability of including a password in the script.

Using RSA-SSH authentication requires creating a public key/private key pair on the host where the SSH session is launched. The host's public key is configured on the Dell EMC Networking switch and the host's private key helps in authenticating the login to the switch.

The following procedure outlines the steps necessary to configure RSA-SSH login from an Ubuntu 16.04 host to Dell EMC Networking OS9 switches.

1. First generate the public/private key pair on the Ansible control machine (Ubuntu 16.04 or any Linux machine) by executing the following command:

```
$ ssh-keygen -t rsa
```

2. Enable the RSA-SSH authentication feature on the Dell EMC Networking switch by executing the following command:

```
Leaf1A(conf)#ip ssh rsa-authentication enable
```

3. Configure the username and password on the Dell EMC Networking switch. The RSA-authentication public key will be added to the admin.

```
Leaf1A(conf)#username admin password xxxx privilege 15
```

4. Login to the Dell EMC Networking switch using SSH and username admin. Copy the public key of the Ansible control machine to the Dell EMC Networking switch by executing the following commands:

```
Leaf1A#copy scp: flash:
Address or name of remote host []: 100.67.1.10
Source file name []: .ssh/id_rsa.pub
User name to login remote host: root
Password to login remote host:
Destination file name [id_rsa.pub]:
!
224 bytes successfully copied
Leaf1A#
```

5. Login into the Dell EMC Networking switch through SSH and configure the public key on the Dell EMC Networking switch by executing the following command:

```
Leaf1A#ip ssh rsa-authentication my-authorized-keys id_rsa.pub
```

```
RSA keys added to admin's list of authorized-keys.  
Delete the file id_rsa.pub : (yes/no) ? yes  
Leaf1#
```

6. Verify the SSH connection by executing the following command:

```
$ ssh 100.67.170.31 -i .ssh/id_rsa -l admin
```

## 2 Playbooks

In Ansible, tasks can be executed on remote devices through Ad-Hoc commands or through playbooks. Ad-Hoc commands are only used when there are minimal tasks (one or two) to execute and there is no need to save them for later. An example of an ad-hoc command is powering off all the machines using a single command. However, for configuration management, deployment and orchestration, Ansible extensively uses playbooks. An Ansible playbook takes inventory and playbook files as arguments and maps the group of hosts listed in the inventory files to the tasks listed in the playbook file.

### 2.1 Inventory

Ansible can configure multiple machines through playbooks. Multiple machine details are passed to the Ansible playbooks through an inventory file. By default, Ansible playbooks use `/etc/ansible/hosts` as the default inventory file. To specify a different inventory file, use the `-i <path of file>` as an option when executing the Ansible playbook.

Example inventory file for a playbook [Inventory 1 – inventory.yml]:

```
[Spine-switches]
Spine1 ansible_host=100.67.170.32 ansible_net_os_name=dellos9

[Leaf-switches]
Leaf1A ansible_host=100.67.170.31 ansible_net_os_name=dellos9
Leaf1B ansible_host=100.67.170.27 ansible_net_os_name=dellos9
```

The headings in the bracket are the group names. The playbooks use these group names to classify which roles are executed against which remote machines. A single machine can be in one or more groups. **Spine1** is the name used by Ansible to identify the host with IP address 100.67.170.32.

Once an inventory file is defined, the host specific files which provide the login credentials to Ansible playbook, must be created. A host specific file `<host_name>.yml` will also have host specific configuration variables as well. The sample `Spine1.yml` is shown below:

```
---
hostname: Spine1
cli :
  host: "{{ ansible_host }}"
  username: admin
  password: xxxx
  authorize: yes
  auth_pass: false
  transport: cli
```

`Leaf1A.yml` and `Leaf1B.yml` also have similar `<host_name>.yml` files.

The link aggregation configuration variables are common to the leaf switches. In this sample playbook [Playbook1 – Dell-EMC\_lag.yml], the link aggregation configurations are applied only to leaf switches. Hence,

the common configuration variables are placed in vars/main.yml file. The sample vars/main.yml file for this configuration is shown below:

```
---
# vars file for Dell-Networking.dellos-lag
dellos_lag:
  Po 1:
    type: dynamic
    description: "To R730-2 100.67.170.2"
    admin: up
    min_links:
    hash:
    channel_members:
      - port: TenGigabitEthernet 1/1
        state: present
    state: present
```

## 2.2 Playbook

Playbooks are the basis for simple configuration management and multi-machine deployments. Playbooks are designed to be human-readable and are expressed in YAML format. Each playbook is composed of one or more plays in a list. By composing multiple plays in a single playbook, Ansible can deploy or orchestrate multi-machine deployments. The goal of the play is to map a group of hosts to Dell EMC Networking Ansible roles, represented by tasks.

**Note:** For more information about the YAML format see [www.yaml.org/start.html](http://www.yaml.org/start.html)

Example of a single play in a single playbook [Playbook 1 – Dell-EMC\_lag.yml]:

```
---
- hosts : Leaf-switches
  connection : local
  gather_facts : no
  roles :
    - Dell-Networking.dellos-lag
```

Example of multiple plays in a single playbook [Playbook 2 – Dell-EMC.yml]:

```
---
- hosts : Spine-switches
  connection : local
  gather_facts : no

  roles :
    - Dell-Networking.dellos-interface
    - Dell-Networking.dellos-vlan
```

```

- Dell-Networking.dellos-bgp

- hosts : Leaf-switches
  connection : local
  gather_facts : no

  roles :
    - Dell-Networking.dellos-interface
    - Dell-Networking.dellos-vlan

```

## 2.3 Ansible Playbook execution

Ansible playbook needs an inventory file and a playbook for execution. To check the syntax of a playbook, use the command `ansible-playbook` with the `--syntax-check` flag. This will run the playbook file through the parser to ensure its included files, roles, etc. have no syntax problems. The following shows the Ansible playbook to configure link aggregation (Playbook 1 – Dell-EMC\_lag.yml) being executed:

```
$ ansible-playbook -i inventory.yml Dell-EMC_lag.yml
```

```

PLAY [Leaf-switches] *****

TASK [Dell-Networking.dellos-lag : Provision LAG configuration for OS9] *****
changed: [Leaf1B]
changed: [Leaf1A]

TASK [Dell-Networking.dellos-lag : Provision LAG configuration for OS10] *****
skipping: [Leaf1A]
skipping: [Leaf1B]

TASK [Dell-Networking.dellos-lag : Provision LAG configuration for OS6] *****
skipping: [Leaf1A]
skipping: [Leaf1B]

PLAY RECAP *****
Leaf1A                : ok=1    changed=1    unreachable=0    failed=0
Leaf1B                : ok=1    changed=1    unreachable=0    failed=0

```

The inventory file used in this case has two groups: Spine-switches and Leaf-switches. The Spine-switches group consists of one switch and the Leaf-switches group contains two switches running OS9. The Dell-EMC\_lag.yml playbook has one play with the Dell-Networking.dellos-lag role, which will be executed on the Leaf-switches. The Dell-EMC\_lag.yml playbook calls the Dell-Networking.dellos-lag role in OS9. The other tasks in OS6 and OS10 are skipped. The PLAY RECAP section confirms that link aggregation has been configured on the Leaf1A and Leaf1B switches, which belong to the Leaf-switches group.



## 3 Leaf-spine configuration through Ansible

This section gives a brief introduction of the leaf-spine architecture and its configuration through Ansible playbook.

### 3.1 Introduction to the leaf-spine architecture

In a leaf-spine architecture, a series of access layer switches form the leaf layer. These switches are connected to a series of spine switches. Each leaf connects to each of the spine switches. The spines do not connect to one another when using the routed leaf-spine example in this chapter. The total number of connections is equal to the number of leaf switches multiplied by the number of spine switches. This equation represents the base number of interconnects. Bandwidth of the fabric may be improved by increasing the number of interconnects as long as the spine layer can support the additional connections. This is often useful in 10GbE environments.

**Note:** The routed leaf-spine example used in this chapter does not have an inter-switch link. In a bridged leaf-spine environment, spine switches have an inter-switch link in the form of a Virtual Link Trunking interconnect (VLTi).

The design ensures that leaf switches are no more than one hop away from one another, minimizing latency and the likelihood of bottlenecks between them. Given any single-link failure scenario, all leaf switches retain connectivity to one another through the remaining links.

The connections between spine switches and leaf switches can be Layer 2 or Layer 3. The deployment scenario in this guide uses Layer 3 connections. The leaf layer is the Layer 2 / Layer 3 boundary for this topology.

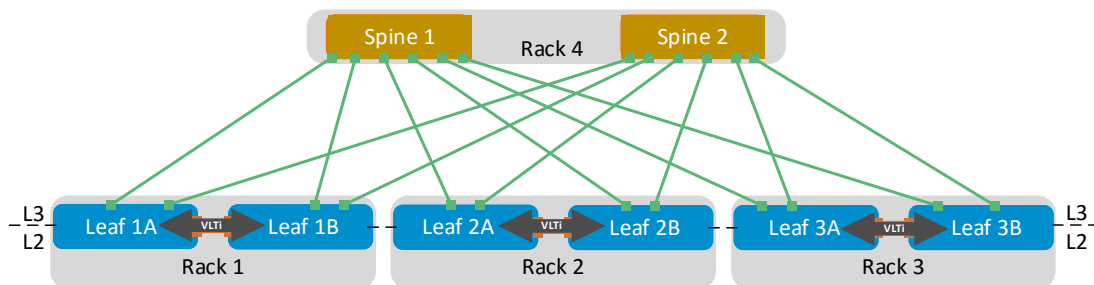


Figure 1 Leaf-spine topology example

Figure 1 shows a high-level diagram of the spine and leaf topology used in this guide. As a best practice, when administrators add racks to the data center, two leaf switches are added to each new rack. These two switches are joined using a VLTi so they are seen as a single logical switch to the rest of the Layer 2 network downstream. Administrators may add spine switches as bandwidth requirements increase.

The following physical concepts apply to all routed leaf-spine topologies:

- Each leaf switch connects to every spine switch in the topology.
- Spine switches only connect to leaf switches.
- Leaf switches connect to spine switches and other devices such as servers, storage arrays and edge routers.
- Servers, storage arrays, edge routers and other non-leaf-switch devices never connect to spine switches.
- Configuring leaf switch pairs for VLTi is a best practice that provides redundancy at the Layer 2 level for devices that use an active-active LAG to attach to both switches.

## 3.2 Ansible configuration for leaf-spine architecture

The following assumptions and limitations have been made in this document for leaf-spine architecture.

- Ansible 2.2 version has been installed on Ubuntu server 16.04. The required Ansible roles are installed in the system.
- All Dell EMC Networking switches are using OS9 operating systems.
- BGP is selected as the routing protocol for the leaf-spine architecture. This can be extended to OSPF or other routing protocols by using existing Dell EMC Networking Ansible roles or developing new ones.
- Limited the leaf-spine Ansible configurations to a single spine and two leaf switches. However, this design can be easily extended to include multiple spine and leaf switches.
- Minimum OS version required for Dell EMC Networking switches for OS6 are 6.3.1 and above; OS9 are 9.10.0.1P13 and above; OS10 are 10.2 and above.
- SSH RSA Authentication Password-less login procedure has been executed on all the switches and the Ansible control machine.

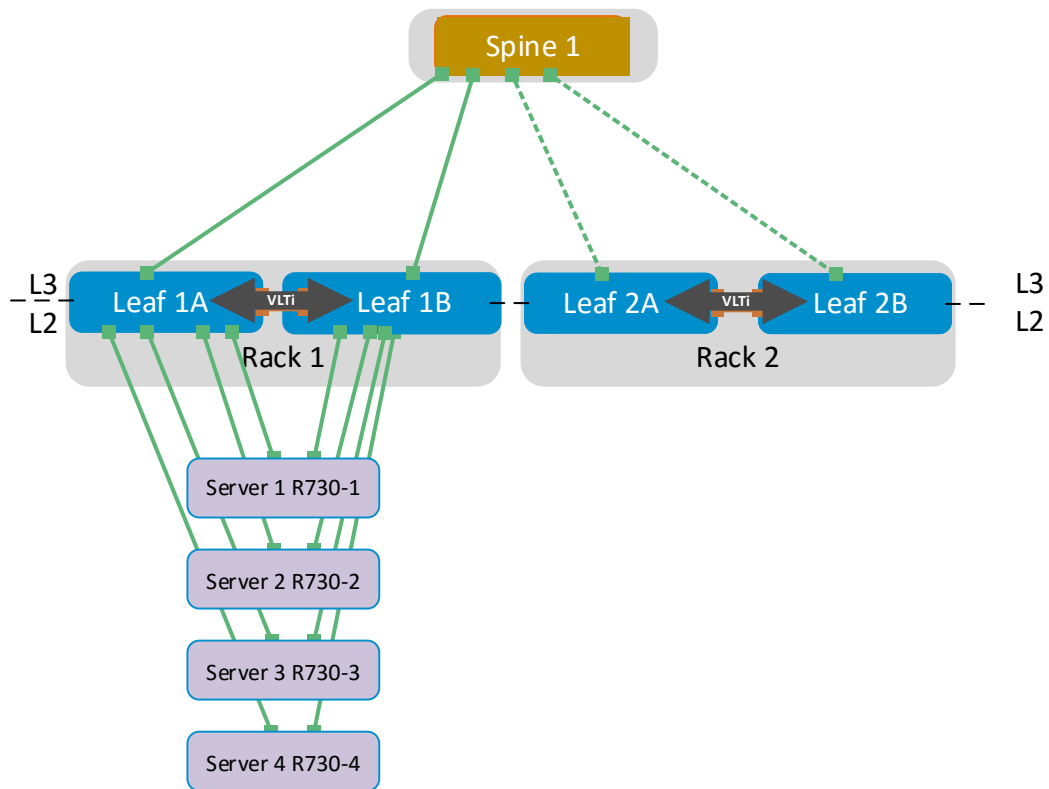


Figure 2 Leaf-spine topology example

Figure 2 shows Ansible being used to configure high-level leaf-spine topology. This guide provides sample configurations for switches including Spine1 and Rack1 switches - Leaf1A and Leaf1B. This can be easily extended to multiple racks based on the requirement.

The following brief configurations have been targeted for the leaf-spine architecture through Ansible:

Table 1 lists the Dell EMC Networking switch management details for Spine1, Leaf1A, and Leaf1B. Execute the basic switch configurations [Section 1.4](#) and RSA-SSH authentication passwordless login configurations [Section 1.5](#) using the management information provided below.

Table 1 Switch management details

Switch Management Details		
Switches	Interface Name	IP address
Spine 1	Management 1/1	100.67.170.32
Leaf 1A	Management 1/1	100.67.170.31
Leaf 1B	Management 1/1	100.67.170.27

Configuration of LAG parameters on leaf switches for VLT and server connections is done through the Dell EMC Networking Ansible role Dell-Networking.dellos-lag. This role is executed only on leaf switches. Table 2 provides the LAG configuration details for Leaf1A and Leaf1B.

Table 2 LAG configuration

Link Aggregation Group Configurations				
Switches	LAG Name	Description	LAG Type	Channel Member(s)
Leaf1A	Port-Channel 1	To R730-1 100.67.170.1	dynamic	TenGigabitEthernet 1/1
	Port-Channel 2	To R730-2 100.67.170.2	dynamic	TenGigabitEthernet 1/2
	Port-Channel 3	To R730-3 100.67.170.3	dynamic	TenGigabitEthernet 1/3
	Port-Channel 4	To R730-4 100.67.170.4	dynamic	TenGigabitEthernet 1/4
	Port-Channel 127	VLTi	Static	fortyGigE 1/53, fortyGigE 1/54
Leaf1B	Port-Channel 1	To R730-1 100.67.170.1	dynamic	TenGigabitEthernet 1/1
	Port-Channel 2	To R730-2 100.67.170.2	dynamic	TenGigabitEthernet 1/2
	Port-Channel 3	To R730-3 100.67.170.3	dynamic	TenGigabitEthernet 1/3
	Port-Channel 4	To R730-4 100.67.170.4	dynamic	TenGigabitEthernet 1/4
	Port-Channel 127	VLTi	Static	fortyGigE 1/53, fortyGigE 1/54

Configuration of interfaces (Physical, LAG, VLAN etc.) for IP addresses, MTU parameters, admin up, assigning layer 2 or hybrid mode etc. are done through the Dell EMC Networking Ansible role Dell-Networking.dellos-interface. This role is executed on both leaf and spine switches. Switchport assignments to LAG interfaces are done in this role. The dellos-interface role is executed after the LAG. Table 3 provides the interface configuration details for Spine1, Leaf1A, and Leaf1B.

Table 3 Interface configuration

Interface Configuration Details					
Switches	Interface	Description	VLT Peer Config	Switchport mode	IP address
Spine 1	FortyGigE 1/1	Rack1_Leaf_1A			192.168.1.0/31
	FortyGigE 1/2	Rack1_Leaf_1B			192.168.1.2/31
Leaf1A		To R730-1			
	TenGigabit Ethernet 1/1	100.67.170.1			
		To R730-2			
	TenGigabit Ethernet 1/2	100.67.170.2			
		To R730-3			
	TenGigabit Ethernet 1/3	100.67.170.3			
		To R730-4			
	TenGigabit Ethernet 1/4	100.67.170.4			
	fortyGiGE 1/49	Connected to spine-1			192.168.1.3/31
	fortyGigE 1/53	VLTi			
	fortyGigE 1/54	VLTi			
		To R730-1			
	Port-channel 1	100.67.170.1	1	Hybrid, True	
		To R730-2			
	Port-channel 2	100.67.170.2	2	Hybrid, True	
Leaf1B		To R730-3			
	Port-channel 3	100.67.170.3	3	Hybrid, True	
		To R730-4			
	Port-channel 4	100.67.170.4	4	Hybrid, True	
	Port-channel 127	VLTi			
		To R730-1			
	TenGigabit Ethernet 1/1	100.67.170.1			
		To R730-2			
	TenGigabit Ethernet 1/2	100.67.170.2			
		To R730-3			
	TenGigabit Ethernet 1/3	100.67.170.3			
		To R730-4			
	TenGigabit Ethernet 1/4	100.67.170.4			
	fortyGiGE 1/49	Connected to spine-1			192.168.1.1/31
	fortyGigE 1/53	VLTi			
	fortyGigE 1/54	VLTi			
		To R730-2			
	Port-channel 1	100.67.170.2	1	Hybrid, True	
		To R730-2			
	Port-channel 2	100.67.170.2	2	Hybrid, True	
		To R730-4			
	Port-channel 3	100.67.170.4	3	Hybrid, True	
		To R730-4			
	Port-channel 4	100.67.170.4	4	Hybrid, True	
	Port-channel 127	VLTi			

Configuration of VLT on leaf switches by creating a VLTi domain, assigning priority, etc. is done through the Dell EMC Networking Ansible role Dell-Networking.dellos-vlt. Dell-Networking.dellos-lag role must be executed before executing Dell-Networking.dellos-vlt. This role is executed only for leaf switches. Table 4 provides the VLTi configuration details for Leaf1A and Leaf1B.

Table 4 VLTi configurations

VLTi Configurations						
Switches	Domain	Backup Destination	Destination Type	Port Channel ID	Unit ID	Peer routing enabled
Leaf1A	1	100.67.170.27	ipv4	127	0	Yes
Leaf1B	1	100.67.170.31	ipv4	127	1	Yes

Configuration of the prefix list for BGP route-advertisements is done through the Dell EMC Networking Ansible role Dell-Networking.dellos-prefix-list. This role is executed for all leaf and spine switches. Table 5 provides the prefix list configuration details for Spine1, Leaf1A, and Leaf1B.

Table 5 Prefix list configurations

Prefix List configurations									
Switches	Type	Name	Description	Seq List Number	isPermit	Network Number	Mask	Condition	PreLen
Spine1	ipv4	leaf-spine	Redistribute loopback and leaf networks	5	TRUE	10.0.0.0	23	GE	32
	ipv4	leaf-spine	Redistribute loopback and leaf networks	10	TRUE	10.0.0.0	8	GE	26
Leaf1A	ipv4	leaf-spine	Redistribute loopback and leaf networks	5	TRUE	10.0.0.0	23	GE	32
	ipv4	leaf-spine	Redistribute loopback and leaf networks	10	TRUE	10.0.0.0	8	GE	24
Leaf1A	ipv4	leaf-spine	Redistribute loopback and leaf networks	5	TRUE	10.0.0.0	23	GE	32
	ipv4	leaf-spine	Redistribute loopback and leaf networks	10	TRUE	10.0.0.0	8	GE	24

Configuration of BGP parameters such as AS number, neighbor configurations, redistribution parameters, maximum path configurations, etc. are done through the Dell EMC Networking Ansible role Dell-Networking.dellos-bgp. This role is executed for leaf and spine switches. Table 6, Table 7, and Table 8 provide the BGP configuration details for Spine1, Leaf1A, and Leaf1B.

Table 6 Spine1 BGP configuration

BGP Configuration Details (Spine1)	
Parameters	Values
ASN	64601
Router ID	192.168.3.100
Max Path EBGp	2
Graceful Restart	TRUE
Best Path: AS Path	multipath-relax
Best Path: Ignore Router ID	TRUE
IPv4 Network	102.1.1.0/30
Neighbor: Name	leaf-spine
Neighbor: Type	peer group
Neighbor: Type Options	fall-over, advertisement-interval, nshut down
Neighbor: IP	192.168.1.1
Neighbor: Type	ipv4
Neighbor: Remote ASN	64704
Neighbor: Peer Group	leaf-spine
Neighbor: Type Options	nshut down
Neighbor: IP	192.168.1.3
Neighbor: Type	ipv4
Neighbor: Remote ASN	64703
Neighbor: Peer Group	leaf-spine
Neighbor: Type Options	nshut down
Redistribute: Route Type	connected
Redistribute: Address Type	ipv4
Redistribute: Route Map Name	spine-leaf

Table 7 Leaf1A BGP configuration

BGP Configuration Details (Leaf1A)	
Parameters	Values
ASN	64703
Router ID	192.168.3.101
Max Path EGBP	2
Graceful Restart	TRUE
Best Path: AS Path	multipath-relax
Best Path: Ignore Router ID	TRUE
Neighbor: Name	leaf-spine
Neighbor: Type	peergroup
Neighbor: Type Options	fall-over, advertisement-interval, nshutdown
Neighbor: IP	192.168.1.2
Neighbor: Type	ipv4
Neighbor: Remote ASN	64601
Neighbor: Peer Group	leaf-spine
Neighbor: Type Options	nshutdown
Redistribute: Route Type	connected
Redistribute: Address Type	ipv4
Redistribute: Route Map Name	leaf-spine

Table 8 Leaf1B BGP configuration

BGP Configuration Details (Leaf1B)	
Parameters	Values
ASN	64704
Router ID	192.168.3.102
Max Path EGBP	2
Graceful Restart	TRUE
Best Path: AS Path	multipath-relax
Best Path: Ignore Router ID	TRUE
Neighbor: Name	leaf-spine
Neighbor: Type	peergroup
Neighbor: Type Options	fall-over, advertisement-interval, nshutdown
Neighbor: IP	192.168.1.0
Neighbor: Type	ipv4
Neighbor: Remote ASN	64601
Neighbor: Peer Group	leaf-spine
Neighbor: Type Options	nshutdown
Redistribute: Route Type	connected
Redistribute: Address Type	ipv4
Redistribute: Route Map Name	leaf-spine



Configuration of Equal-Cost Multi-Path (ECMP) parameters is done through the Dell EMC Networking Ansible role Dell-Networking.dellos-ecmp. This role is executed on the leaf switches. Table 9 provides the ECMP configuration details for Leaf1A and Leaf1B.

Table 9 ECMP configuration

ECMP Configurations			
Switches	ECMP Group No	Interfaces	Link Bundle Monitor
Leaf1A	1	fortyGigE 1/49, fortyGigE 1/51	Enable
Leaf1B	1	fortyGigE 1/49, fortyGigE 1/51	Enable

The following sections provide detailed Ansible playbook, inventory and role variable details for this leaf-spine architecture.

### 3.2.1 Inventory File

The inventory file defines the leaf and spine switches, and their groups. The `ansible_net_os_name` is set to **dellos9**, which means Dell EMC Networking OS9. Ansible uses SSH to login into these switches using the IP address mentioned in `ansible_host`.

```
[Spine-switches]
Spine1 ansible_host=100.67.170.32 ansible_net_os_name=dellos9

[Leaf-switches]
Leaf1A ansible_host=100.67.170.31 ansible_net_os_name=dellos9
Leaf1B ansible_host=100.67.170.27 ansible_net_os_name=dellos9
```

The login credentials and the host specific configuration variables are configured in the host specific inventory files. The sample files for the leaf-spine architecture are listed below.

Spine1.yml inventory file for the spine switch in the Spine-switches group.

```
---
hostname : Spine1
cli :
  host : "{{ ansible_host }}"
  username : admin
  password : xxxxxx
  authorize : yes
  auth_pass : false
  transport : cli

dellos_bgp:
  asn: 64601
  router_id: 192.168.3.100
```

```

maxpath_ebgp: 2
graceful_restart:
  state: present
best_path:
  as_path: multipath-relax
  ignore_router_id: true
ipv4_network:
  - address: 102.1.1.0/30
    state: present
neighbor:
  - name: leaf-spine
    type: peergroup
    type_options:
      fall_over: present
      adv_interval: 1
    admin: up
    state: present
  - ip: 192.168.1.1
    type: ipv4
    remote_asn: 64704
    peergroup: leaf-spine
    peergroup_state: present
    admin: up
    state: present
  - ip: 192.168.1.3
    type: ipv4
    remote_asn: 64703
    peergroup: leaf-spine
    peergroup_state: present
    admin: up
    state: present
redistribute:
  - route_type: connected
    state: present
    address_type: ipv4
    route_map: present
    route_map_name : leaf-spine
dellos_interface:
  fortyGigE 1/1:
    desc: "Rack1_Leaf_A"
    admin: up
    ip_and_mask: 192.168.1.0/31
  fortyGigE 1/2:
    desc: "Rack1_Leaf_B"
    admin: up
    ip_and_mask: 192.168.1.2/31
dellos_prefixlist:

```

```

- type: ipv4
  name: leaf-spine
  description: Redistribute loopback and leaf networks
  entries:
    - number: 5
      permit: true
      net_num: 10.0.0.0
      mask: 23
      condition_list:
        - condition: ge
          prelen: 32
    - number: 10
      permit: true
      net_num: 10.0.0.0
      mask: 8
      condition_list:
        - condition: ge
          prelen: 26

```

Leaf1A.yml inventory file for the first leaf switch in the Leaf-switches group.

```

---
hostname : Leaf1A
cli :
  host : "{{ ansible_host }}"
  username : admin
  password : xxxxxx
  authorize : yes
  auth_pass : false
  transport : cli
delllos_bgp:
  asn: 64703
  router_id: 192.168.3.101
  maxpath_ebgp: 2
  graceful_restart:
    state: present
  best_path:
    as_path: multipath-relax
    ignore_router_id: true
  neighbor:
    - name: leaf-spine
      type: peergroup
      type_options:
        fall_over: present
      adv_interval: 1
      admin: up

```

```

    state: present
- ip: 192.168.1.2
  type: ipv4
  remote_asn: 64601
  peergroup: leaf-spine
  peergroup_state: present
  admin: up
  state: present
redistribute:
- route_type: connected
  state: present
  address_type: ipv4
  route_map: present
  route_map_name: leaf-spine
dellos_interface:
  TenGigabitEthernet 1/1:
    desc: "To R730-2 100.67.170.2"
    admin: up
  TenGigabitEthernet 1/2:
    desc: "To R730-2 100.67.170.2"
    admin: up
  TenGigabitEthernet 1/3:
    desc: "To R730-4 100.67.170.4"
    admin: up
  TenGigabitEthernet 1/4:
    desc: "To R730-4 100.67.170.4"
    admin: up
  fortyGigE 1/49:
    desc: "Connected to spine-1"
    mtu:
    portmode:
    admin: up
    ip_and_mask: 192.168.1.3/31
  fortyGigE 1/53:
    desc: "VLTi"
    admin: up
  fortyGigE 1/54:
    desc: "VLTi"
    admin: up
  Port-channel 1:
    desc: "To R730-2 100.67.170.2"
    switchport: True
    portmode: "hybrid"
    vlt_peer: 1
    admin: up
  Port-channel 2:
    desc: "To R730-2 100.67.170.2"

```

```

    switchport: True
    portmode: "hybrid"
    vlt_peer: 2
    admin: up
Port-channel 3:
    desc: "To R730-4 100.67.170.4"
    switchport: True
    portmode: "hybrid"
    vlt_peer: 3
    admin: up
Port-channel 4:
    desc: "To R730-4 100.67.170.4"
    switchport: True
    portmode: "hybrid"
    vlt_peer: 4
    admin: up
Port-channel 127:
    desc: "VLTi"
    admin: up
dellos_vlt:
    domain: 1
    backupdestination: 100.67.170.27
    destination_type: ipv4
    priority:
    channelid: 127
    unitid: 0
    vrfname:
    state: present
    peerrouting: True
dellos_prefixlist:
  - type: ipv4
    name: leaf-spine
    description: Redistribute loopback and leaf networks
    entries:
      - number: 5
        permit: true
        net_num: 10.0.0.0
        mask: 23
        condition_list:
          - condition: ge
            prelen: 32
      - number: 10
        permit: true
        net_num: 10.0.0.0
        mask: 8
        condition_list:
          - condition: ge

```

prelen: 24

Leaf1B.yml inventory file for the second leaf switch in the Leaf-switches group.

```
---
hostname : Leaf1B
cli :
  host : "{{ ansible_host }}"
  username : admin
  password : xxxxxx
  authorize : yes
  auth_pass : false
  transport : cli
interface : TenGigabitEthernet 1/19

dellos_bgp:
  asn: 64704
  router_id: 192.168.3.102
  maxpath_ebgp: 2
  graceful_restart:
    state: present
  best_path:
    as_path: multipath-relax
    ignore_router_id: true
  neighbor:
    - name: leaf-spine
      type: peergroup
      type_options:
        fall_over: present
        adv_interval: 1
        admin: up
        state: present
    - ip: 192.168.1.0
      type: ipv4
      remote_asn: 64601
      peergroup: leaf-spine
      peergroup_state: present
      admin: up
      state: present
  redistribute:
    - route_type: connected
      state: present
      address_type: ipv4
      route_map: present
      route_map_name: leaf-spine
dellos_interface:
```

```

TenGigabitEthernet 1/1:
  desc: "To R730-2 100.67.170.2"
  admin: up
TenGigabitEthernet 1/2:
  desc: "To R730-2 100.67.170.2"
  admin: up
TenGigabitEthernet 1/3:
  desc: "To R730-4 100.67.170.4"
  admin: up
TenGigabitEthernet 1/4:
  desc: "To R730-4 100.67.170.4"
  admin: up
fortyGigE 1/49:
  desc: "Connected to spine-1"
  mtu:
  portmode:
  admin: up
  ip_and_mask: 192.168.1.1/31
fortyGigE 1/53:
  desc: "VLTi"
  admin: up
fortyGigE 1/54:
  desc: "VLTi"
  admin: up
Port-channel 1:
  desc: "To R730-2 100.67.170.2"
  switchport: True
  portmode: "hybrid"
  vlt_peer: 1
  admin: up
Port-channel 2:
  desc: "To R730-2 100.67.170.2"
  switchport: True
  portmode: "hybrid"
  vlt_peer: 2
  admin: up
Port-channel 3:
  desc: "To R730-4 100.67.170.4"
  switchport: True
  portmode: "hybrid"
  vlt_peer: 3
  admin: up
Port-channel 4:
  desc: "To R730-4 100.67.170.4"
  switchport: True
  portmode: "hybrid"
  vlt_peer: 4

```

```

    admin: up
  Port-channel 127:
    desc: "VLTi"
    admin: up
dellos_vlt:
  domain: 1
  backupdestination: 100.67.170.31
  destination_type: ipv4
  priority:
  channelid: 127
  unitid: 1
  vrfname:
  state: present
  peerrouting: True
dellos_prefixlist:
  - type: ipv4
    name: leaf-spine
    description: Redistribute loopback and leaf networks
    entries:
      - number: 5
        permit: true
        net_num: 10.0.0.0
        mask: 23
        condition_list:
          - condition: ge
            prelen: 32
      - number: 10
        permit: true
        net_num: 10.0.0.0
        mask: 8
        condition_list:
          - condition: ge
            prelen: 24

```

### 3.2.2 Playbook

The playbook shown below is used to configure the leaf-spine architecture. There are two plays in this playbook. The first play is applied on the spine switches and the second play is applied on leaf switches. The first play maps the spine switches to tasks using Ad-hoc commands like `route-map leaf-spine permit 10, match ip address leaf-spine` and the roles including interface, prefix list and BGP. The second play maps the leaf switches to tasks using Ad-hoc commands similar to the first play and roles including link aggregation(LAG), interface, VLT, prefix list, BGP and ECMP.

```

---
- hosts : Spine-switches
  connection : local

```



```

gather_facts : no
tasks :
- name : "Basic Configuration"
  dellos9_config :
    commands:
      - route-map leaf-spine permit 10
      - match ip address leaf-spine
    provider : "{{cli}}"
roles :
- Dell-Networking.dellos-interface
- Dell-Networking.dellos-prefix-list
- Dell-Networking.dellos-bgp

- hosts : Leaf-switches
  connection : local
  gather_facts : no
  tasks :
- name : "Basic Configuration"
  dellos9_config :
    commands:
      - route-map leaf-spine permit 10
      - match ip address leaf-spine
    provider : "{{cli}}"
roles :
- Dell-Networking.dellos-lag
- Dell-Networking.dellos-interface
- Dell-Networking.dellos-vlt
- Dell-Networking.dellos-prefix-list
- Dell-Networking.dellos-bgp
- Dell-Networking.dellos-ecmp

```

### 3.2.3 Dell EMC Networking Ansible role: Link Aggregation Group (LAG)

Dell-Networking.dellos-lag role in Ansible is used to configure the link aggregation protocol in Dell EMC Networking switches. This role is applied to the leaf switches as shown in the leaf-spine playbook. Since the link aggregation configurations are common in both the switches, the configuration variables are placed in the vars/main.yml file. The link aggregation attributes are defined here in the key value pairs. Based on the values to the key attributes, the Ansible playbook executes the dellos\_lag template and configures the switches.

For example, the port channel 1 (Po 1) type attribute is defined as dynamic. This means the port channel is configured as a dynamic link aggregation, i.e. by configuring the LACP protocol. For port channel 127 (Po 127) the type attribute is defined as static, i.e. port channel 127 is configured as a static port channel. The port channel 1 members include interface TenGigabitEthernet 1/1 and the port channel 127 members include FortyGigabitEthernet 1/53 and FortyGigabitEthernet 1/54.

---

# vars file for Dell-Networking.dellos-lag

```

dellos_lag:
  Po 1:
    type: dynamic
    description: "To R730-1 100.67.170.1"
    admin: up
    min_links:
    hash:
    channel_members:
      - port: TenGigabitEthernet 1/1
        state: present
    state: present
  Po 2:
    type: dynamic
    description: "To R730-2 100.67.170.2"
    admin: up
    min_links:
    hash:
    channel_members:
      - port: TenGigabitEthernet 1/2
        state: present
    state: present
  Po 3:
    type: dynamic
    description: "To R730-3 100.67.170.3"
    admin: up
    min_links:
    hash:
    channel_members:
      - port: TenGigabitEthernet 1/3
        state: present
    state: present
  Po 4:
    type: dynamic
    description: "To R730-4 100.67.170.4"
    admin: up
    min_links:
    hash:
    channel_members:
      - port: TenGigabitEthernet 1/4
        state: present
    state: present
  Po 127:
    type: static
    min_links:
    hash:
    channel_members:

```

```

- port: fortyGigE 1/53
  state: present
- port: fortyGigE 1/54
  state: present
state: present

```

### 3.2.4 Dell EMC Networking Ansible role: interface

Dell-Networking.dellos-interface role is used to configure the interfaces in Dell EMC Networking switches. This role is present in both the plays of the playbook (both spine switches and leaf switches). The interface configuration attributes include assigning an IP address/network mask, configuring the interface in switchport mode, hybrid mode, administrative shutdown/no shutdown, configuring MTU, etc. Based on the key pair values of the interface variables, the dellos\_interface role configures the Dell EMC Networking switches. These interface variables are specific to each host, hence the interface configuration variables are placed in host.yml file for each switch.

The sample interface configuration of Leaf1B switch [Refer to Leaf1B.yml, [Inventory File section](#) ]:

```

dellos_interface:
  TenGigabitEthernet 1/1:
    desc: "To R730-1 100.67.170.1"
    admin: up
  TenGigabitEthernet 1/2:
    desc: "To R730-2 100.67.170.2"
    admin: up
  TenGigabitEthernet 1/3:
    desc: "To R730-3 100.67.170.3"
    admin: up
  TenGigabitEthernet 1/4:
    desc: "To R730-4 100.67.170.4"
    admin: up
  fortyGigE 1/49:
    desc: "Connected to spine-1"
    mtu:
    portmode:
    admin: up
    ip_and_mask: 192.168.1.1/31
  fortyGigE 1/53:
    desc: "VLTi"
    admin: up
  fortyGigE 1/54:
    desc: "VLTi"
    admin: up
  Port-channel 1:
    desc: "To R730-1 100.67.170.1"
    switchport: True
    portmode: "hybrid"

```

```

    vlt_peer: 1
    admin: up
Port-channel 2:
    desc: "To R730-2 100.67.170.2"
    switchport: True
    portmode: "hybrid"
    vlt_peer: 2
    admin: up
Port-channel 3:
    desc: "To R730-3 100.67.170.3"
    switchport: True
    portmode: "hybrid"
    vlt_peer: 3
    admin: up
Port-channel 4:
    desc: "To R730-4 100.67.170.4"
    switchport: True
    portmode: "hybrid"
    vlt_peer: 4
    admin: up
Port-channel 127:
    desc: "VLTi"
    admin: up

```

The sample interface configuration of Leaf1A switch [Refer to Leaf1A.yml, [Inventory File section](#)]:

```

dellos_interface:
  TenGigabitEthernet 1/1:
    desc: "To R730-1 100.67.170.1"
    admin: up
  TenGigabitEthernet 1/2:
    desc: "To R730-2 100.67.170.2"
    admin: up
  TenGigabitEthernet 1/3:
    desc: "To R730-3 100.67.170.3"
    admin: up
  TenGigabitEthernet 1/4:
    desc: "To R730-4 100.67.170.4"
    admin: up
  fortyGigE 1/49:
    desc: "Connected to spine-1"
    mtu:
    portmode:
    admin: up
    ip_and_mask: 192.168.1.3/31
  fortyGigE 1/53:

```

```

    desc: "VLTi"
    admin: up
  fortyGigE 1/54:
    desc: "VLTi"
    admin: up
  Port-channel 1:
    desc: "To R730-2 100.67.170.2"
    switchport: True
    portmode: "hybrid"
    vlt_peer: 1
    admin: up
  Port-channel 2:
    desc: "To R730-2 100.67.170.2"
    switchport: True
    portmode: "hybrid"
    vlt_peer: 2
    admin: up
  Port-channel 3:
    desc: "To R730-4 100.67.170.4"
    switchport: True
    portmode: "hybrid"
    vlt_peer: 3
    admin: up
  Port-channel 4:
    desc: "To R730-4 100.67.170.4"
    switchport: True
    portmode: "hybrid"
    vlt_peer: 4
    admin: up
  Port-channel 127:
    desc: "VLTi"
    admin: up

```

The sample interface configuration of Spine1 switch [Refer to Spine1.yml, [Inventory File section](#)]:

```

dellos_interface:
  fortyGigE 1/1:
    desc: "Rack1_Leaf_A"
    admin: up
    ip_and_mask: 192.168.1.0/31
  fortyGigE 1/2:
    desc: "Rack1_Leaf_B"
    admin: up
    ip_and_mask: 192.168.1.2/31

```

### 3.2.5 Dell EMC Networking Ansible role: VLT

Dell-Networking.dellos-vlt role is used to configure the VLT parameters in Dell EMC Networking switches. This role is only supported for OS9. In the leaf-spine playbook, this role is present in the second play of the playbook (mapped only to leaf switches).

For the leaf-spine architecture, the following values are configured for VLT. Port channel 127 must be configured before VLT is configured, so Dell-Networking.dellos-lag must be executed before the VLT role. The configurations of VLT are specific to each switch, hence the configuration variables must be placed in the host.yml file.

The sample VLT configuration of Leaf1A switch [Refer to Leaf1A.yml, [Inventory File section](#)]:

```
dellos_vlt:
  domain: 1
  backupdestination: 100.67.170.27
  destination_type: ipv4
  priority:
  channelid: 127
  unitid: 1
  vrfname:
  state: present
  peerrouting: True
```

The sample VLT configuration of Leaf1B switch [Refer to Leaf1B.yml, [Inventory File section](#)]:

```
dellos_vlt:
  domain: 1
  backupdestination: 100.67.170.31
  destination_type: ipv4
  priority:
  channelid: 127
  unitid: 0
  vrfname:
  state: present
  peerrouting: True
```

### 3.2.6 Dell EMC Networking Ansible Networking role: prefix-list

The Dell-Networking.dellos-prefix-list role is used to configure the prefix-list parameters in Dell EMC Networking switches. This role is supported only for OS9. In the leaf-spine playbook, this role is present for both the plays in the playbook. For the leaf-spine architecture the prefix-list configurations are specific to host switches, hence the configurations are placed in host\_name.yml file. The prefix list configurations are shown below.

The sample prefix-list configuration of Spine1 switch [Refer to Spine1.yml, [Inventory File section](#)]:

```
dellos_prefixlist:
```

```

- type: ipv4
  name: leaf-spine
  description: Redistribute loopback and leaf networks
  entries:
    - number: 5
      permit: true
      net_num: 10.0.0.0
      mask: 23
      condition_list:
        - condition: ge
          prelen: 32
    - number: 10
      permit: true
      net_num: 10.0.0.0
      mask: 8
      condition_list:
        - condition: ge
          prelen: 26

```

The sample prefix-list configuration of Leaf1A switch [Refer to Leaf1A.yml, [Inventory File section](#)]:

```

dellos_prefixlist:
- type: ipv4
  name: leaf-spine
  description: Redistribute loopback and leaf networks
  entries:
    - number: 5
      permit: true
      net_num: 10.0.0.0
      mask: 23
      condition_list:
        - condition: ge
          prelen: 32
    - number: 10
      permit: true
      net_num: 10.0.0.0
      mask: 8
      condition_list:
        - condition: ge
          prelen: 24

```

The sample prefix-list configuration of Leaf1B switch [Refer to Leaf1B.yml, [Inventory File section](#)]:

```

dellos_prefixlist:
- type: ipv4
  name: leaf-spine
  description: Redistribute loopback and leaf networks
  entries:

```

```
- number: 5
  permit: true
  net_num: 10.0.0.0
  mask: 23
  condition_list:
    - condition: ge
      prelen: 32
- number: 10
  permit: true
  net_num: 10.0.0.0
  mask: 8
  condition_list:
    - condition: ge
      prelen: 24
```



### 3.2.7 Dell EMC Networking Ansible role: BGP

The Dell-Networking.dellos-bgp role is used to configure the BGP parameters in Dell EMC Networking switches. In the leaf-spine playbook, this role is present in both the plays of the playbook (mapped only to both leaf switches and spine switches). The router ID, bgp neighbor configurations, etc. are specific to host switches, hence the BGP configuration variables are placed in the host\_name.yml file.

The sample BGP configuration of the Spine1 switch [Refer to Spine1.yml, [Inventory File section](#)]:

```
dellos_bgp:
  asn: 64601
  state: present
  router_id: 192.168.3.100
  #maxpath_ibgp: 2
  maxpath_ebgp: 2
  graceful_restart:
    state: present
  best_path:
    as_path: multipath-relax
    ignore_router_id: true
  ipv4_network:
    - address: 102.1.1.0/30
      state: present
  neighbor:
    - name: leaf-spine
      type: peergroup
      fall_over: present
      adv_interval: 10
      admin: up
      state: present
    - ip: 192.168.1.1
      type: ipv4
      remote_asn: 64704
      peergroup: leaf-spine
      peergroup_state: present
      admin: up
      state: present
    - ip: 192.168.1.3
      type: ipv4
      remote_asn: 64703
      peergroup: leaf-spine
      peergroup_state: present
      admin: up
  redistribute:
    - route_type: connected
      state: present
      address_type: ipv4
      route_map: present
```

```
route_map_name: leaf-spine
```

The sample BGP configuration of Leaf1A switch [Refer to Leaf1A.yml, [Inventory File section](#)]:

```
dellos_bgp:
  asn: 64703
  state: present
  router_id: 192.168.3.101
  #maxpath_ibgp: 2
  maxpath_ebgp: 2
  graceful_restart:
    state: present
  best_path:
    as_path: multipath-relax
    ignore_router_id: true
  neighbor:
    - name: leaf-spine
      type: peergroup
      fall_over: present
      adv_interval: 10
      admin: up
      state: present
    - ip: 192.168.1.2
      type: ipv4
      remote_asn: 64601
      peergroup: leaf-spine
      peergroup_state: present
      admin: up
      state: present
  redistribute:
    - route_type: connected
      state: present
      address_type: ipv4
      route_map: present
      route_map_name: leaf-spine
```

The sample BGP configuration of Leaf1B switch [Refer to Leaf1B.yml, [Inventory File section](#)]:

```
dellos_bgp:
  asn: 64704
  state: present
  router_id: 192.168.3.102
  #maxpath_ibgp: 2
  maxpath_ebgp: 2
  graceful_restart:
    state: present
```

```

best_path:
  as_path: multipath-relax
  ignore_router_id: true
neighbor:
  - name: leaf-spine
    type: peergroup
    fall_over: present
    adv_interval: 10
    admin: up
    state: present
  - ip: 192.168.1.0
    type: ipv4
    remote_asn: 64601
    peergroup: leaf-spine
    peergroup_state: present
    admin: up
    state: present
redistribute:
  - route_type: connected
    state: present
    address_type: ipv4
    route_map: present
    route_map_name: leaf-spine

```

### 3.2.8 Dell EMC Networking Ansible role: ECMP

The Dell-Networking.dellos-ecmp role is used to configure the prefix-list parameters in Dell EMC Networking switches. This role is supported only for OS9. In the leaf-spine playbook, this role is present in second play of the playbook (mapped only to leaf switches). The ECMP configuration is common to both the leaf switches, hence this configuration is placed in the vars/main.yml file.

The sample configuration for ECMP is shown below:

```

---
dellos_ecmp:
  ecmp 1:
    interface:
      - fortyGigE 1/49
      - fortyGigE 1/51
    link_bundle_monitor: enable
    state: present

```

### 3.2.9 Execution of Dell leaf-spine playbook

All inventory and playbook file configurations are explained in the previous sections. To execute the leaf-spine playbook and configure the leaf-spine, execute the command shown below.

```

root@dell-PowerEdge-R620:~/automation/ansible# ansible-playbook -i inventory
bgp-leaf-spine_config.yml

PLAY [Spine-switches] *****

TASK [Dell-Networking.dellos-interface : Provision Interface configuration for
OS9] ***
ok: [Spine1]

TASK [Dell-Networking.dellos-prefix-list : Provision prefix-list configuration
for OS9] *
ok: [Spine1]

TASK [Dell-Networking.dellos-bgp : Provision BGP configuration for OS9] *****
changed: [Spine1]

TASK [Basic Configuration] *****
changed: [Spine1]

PLAY [Leaf-switches] *****

TASK [Dell-Networking.dellos-lag : Provision LAG configuration for OS9] *****
changed: [Leaf1B]
changed: [Leaf1A]

TASK [Dell-Networking.dellos-lag : Provision LAG configuration for OS10] *****
skipping: [Leaf1A]
skipping: [Leaf1B]

TASK [Dell-Networking.dellos-lag : Provision LAG configuration for OS6] *****
skipping: [Leaf1A]
skipping: [Leaf1B]

TASK [Dell-Networking.dellos-interface : Provision Interface configuration for
OS9] ***
changed: [Leaf1B]
changed: [Leaf1A]

TASK [Dell-Networking.dellos-vlt : Provision vlt configuration for OS9] *****
changed: [Leaf1B]
changed: [Leaf1A]

TASK [Dell-Networking.dellos-prefix-list : Provision prefix-list configuration
for OS9] *
ok: [Leaf1B]
ok: [Leaf1A]

```

```
TASK [Dell-Networking.dellos-bgp : Provision BGP configuration for OS9] *****
changed: [Leaf1B]
changed: [Leaf1A]
```

```
TASK [Dell-Networking.dellos-ecmp : Provision ecmp configuration for OS9] *****
changed: [Leaf1A]
changed: [Leaf1B]
```

```
TASK [Basic Configuration] *****
changed: [Leaf1B]
changed: [Leaf1A]
```

```
PLAY RECAP *****
Leaf1A                : ok=11    changed=6    unreachable=0    failed=0
Leaf1B                : ok=11    changed=6    unreachable=0    failed=0
Spine1                 : ok=8     changed=2    unreachable=0    failed=0
```

Spine switch configuration after executing the Ansible-playbook:

```
Spine1#
Spine1#show running-config
Current Configuration ...
! Version 9.11(0.0)
! Last configuration change at Sat Apr 22 02:57:50 2017 by admin
!
hostname Spine1
!
protocol lldp
!
redundancy auto-synchronize full
!
username admin password 7 7fa786736049da17 privilege 15
!
stack-unit 1 quad-port-profile
2,4,6,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,26,28,30,32
!
stack-unit 1 provision S6010-ON
!
interface fortyGigE 1/1
  description Rack1_Leaf_1A
  ip address 192.168.1.0/31
  no shutdown
!
interface fortyGigE 1/2
  description Rack1_Leaf_1B
```

```

    ip address 192.168.1.2/31
    no shutdown
!
interface fortyGigE 1/3
    no ip address
    shutdown
!
    .
    .
    .
!
interface fortyGigE 1/32
    no ip address
    shutdown
!
interface ManagementEthernet 1/1
    ip address 100.67.170.32/24
    no shutdown
!
interface Vlan 1
!
route-map leaf-spine permit 10
    match ip address leaf-spine
!
router bgp 64601
    bgp router-id 192.168.3.100
    network 102.1.1.0/30
    bgp bestpath as-path multipath-relax
    bgp bestpath router-id ignore
    maximum-paths ebgp 2
    redistribute connected
    bgp graceful-restart
    neighbor leaf-spine peer-group
    neighbor leaf-spine shutdown
    neighbor 192.168.1.1 remote-as 64704
    neighbor 192.168.1.1 peer-group leaf-spine
    neighbor 192.168.1.1 shutdown
    neighbor 192.168.1.3 remote-as 64703
    neighbor 192.168.1.3 peer-group leaf-spine
    neighbor 192.168.1.3 shutdown
!
management route 0.0.0.0/0 100.67.170.254
!
ip ssh server enable
!
Spine1#

```

Leaf1A switch configuration after executing the Ansible-playbook:

```
Leaf1A#show running-config
Current Configuration ...
! Version 9.11(0.0P2)
! Last configuration change at Fri Apr 21 12:06:28 2017 by admin
!
hostname Leaf1A
!
protocol lldp
!
redundancy auto-synchronize full
!
username admin password 7 7fa786736049da17 privilege 15
!
vlt domain 1
  peer-link port-channel 127
  back-up destination 100.67.170.27
  unit-id 1
  peer-routing
!
stack-unit 1 provision S4048-ON
!
interface TenGigabitEthernet 1/1
  description To R730-1 100.67.170.1
  no ip address
!
  port-channel-protocol LACP
  port-channel 1 mode active
  no shutdown
!
interface TenGigabitEthernet 1/2
  description To R730-2 100.67.170.2
  no ip address
!
  port-channel-protocol LACP
  port-channel 2 mode active
  no shutdown
!
interface TenGigabitEthernet 1/3
  description To R730-3 100.67.170.3
  no ip address
!
  port-channel-protocol LACP
  port-channel 3 mode active
```

```

    no shutdown
!
interface TenGigabitEthernet 1/4
    description To R730-4 100.67.170.4
    no ip address
!
    port-channel-protocol LACP
    port-channel 4 mode active
    no shutdown
        .
        .
        .

interface fortyGigE 1/49
    description Connected to spine-1
    ip address 192.168.1.3/31
    no shutdown
        .
        .
        .

interface fortyGigE 1/53
    description VLTi
    no ip address
    no shutdown
!
interface fortyGigE 1/54
    description VLTi
    no ip address
    no shutdown
!
interface ManagementEthernet 1/1
    ip address 100.67.170.31/24
    no shutdown
        .
        .
        .

interface Port-channel 1
    description To R730-1 100.67.170.1
    no ip address
    portmode hybrid
    switchport
    vlt-peer-lag port-channel 1
    no shutdown
!
interface Port-channel 2
    description To R730-2 100.67.170.2

```



```

no ip address
portmode hybrid
switchport
vlt-peer-lag port-channel 2
no shutdown
!
interface Port-channel 3
description To R730-3 100.67.170.3
no ip address
portmode hybrid
switchport
vlt-peer-lag port-channel 3
no shutdown
!
interface Port-channel 4
description To R730-4 100.67.170.4
no ip address
portmode hybrid
switchport
vlt-peer-lag port-channel 4
no shutdown
!
interface Port-channel 127
description VLTi
no ip address
channel-member fortyGigE 1/53,1/54
no shutdown
!
interface Vlan 1
!untagged Port-channel 1-4,127
!
route-map leaf-spine permit 10
match ip address leaf-spine
!
router bgp 64703
bgp router-id 192.168.3.101
bgp bestpath as-path multipath-relax
bgp bestpath router-id ignore
maximum-paths ebgp 2
redistribute connected
bgp graceful-restart
neighbor leaf-spine peer-group
neighbor leaf-spine shutdown
neighbor 192.168.1.2 remote-as 64601
neighbor 192.168.1.2 peer-group leaf-spine
neighbor 192.168.1.2 shutdown
!

```

```

management route 0.0.0.0/0 100.67.170.254
!
ip ssh rsa-authentication enable
ip ssh server enable
!
ecmp-group 1
    interface fortyGigE 1/49
    interface fortyGigE 1/51
    link-bundle-monitor enable
Leaf1A#

```

Leaf1B switch configuration after executing the Ansible-playbook:

```

Leaf1B#show running-config
Current Configuration ...
! Version 9.11(0.0P4)
! Last configuration change at Fri Apr 21 18:39:29 2017 by admin
!
hostname Leaf1B
!
protocol lldp
!
redundancy auto-synchronize full
!
username admin password 7 7fa786736049da17 privilege 15
!
vlt domain 1
    peer-link port-channel 127
    back-up destination 100.67.170.31
    unit-id 0
    peer-routing
!
stack-unit 1 provision S4048-ON
!
interface TenGigabitEthernet 1/1
    description To R730-1 100.67.170.1
    no ip address
!
    port-channel-protocol LACP
    port-channel 1 mode active
    no shutdown
!
interface TenGigabitEthernet 1/2
    description To R730-2 100.67.170.2

```

```

    no ip address
    !
    port-channel-protocol LACP
    port-channel 2 mode active
    no shutdown
    !
interface TenGigabitEthernet 1/3
    description To R730-3 100.67.170.3
    no ip address
    !
    port-channel-protocol LACP
    port-channel 3 mode active
    no shutdown
    !
interface TenGigabitEthernet 1/4
    description To R730-4 100.67.170.4
    no ip address
    !
    port-channel-protocol LACP
    port-channel 4 mode active
    no shutdown
    .
    .
    .
interface fortyGigE 1/49
    description Connected to spine-1
    ip address 192.168.1.1/31
    no shutdown
    .
    .
    .
interface fortyGigE 1/53
    description VLTi
    no ip address
    no shutdown
    !
interface fortyGigE 1/54
    description VLTi
    no ip address
    no shutdown
    !
interface ManagementEthernet 1/1
    ip address 100.67.170.27/24
    no shutdown
    .
    .
    .
interface Port-channel 1

```

```

description To R730-1 100.67.170.1
no ip address
portmode hybrid
switchport
vlt-peer-lag port-channel 1
no shutdown
!
interface Port-channel 2
description To R730-2 100.67.170.2
no ip address
portmode hybrid
switchport
vlt-peer-lag port-channel 2
no shutdown
!
interface Port-channel 3
description To R730-3 100.67.170.3
no ip address
portmode hybrid
switchport
vlt-peer-lag port-channel 3
no shutdown
!
interface Port-channel 4
description To R730-4 100.67.170.4
no ip address
portmode hybrid
switchport
vlt-peer-lag port-channel 4
no shutdown
!
interface Port-channel 127
description VLTi
no ip address
channel-member fortyGigE 1/53,1/54
no shutdown
!
interface Vlan 1
!untagged Port-channel 1-4,127
!
route-map leaf-spine permit 10
match ip address leaf-spine
!
router bgp 64704
bgp router-id 192.168.3.102
bgp bestpath as-path multipath-relax
bgp bestpath router-id ignore

```

```
maximum-paths ebgp 2
redistribute connected
bgp graceful-restart
neighbor leaf-spine peer-group
neighbor leaf-spine shutdown
neighbor 192.168.1.0 remote-as 64601
neighbor 192.168.1.0 peer-group leaf-spine
neighbor 192.168.1.0 shutdown
!
management route 0.0.0.0/0 100.67.170.254
!
ip ssh rsa-authentication enable
ip ssh server enable
!
ecmp-group 1
    interface fortyGigE 1/49
    interface fortyGigE 1/51
    link-bundle-monitor enable
```

## A Additional Information

Complete documentation about Ansible installation, getting started, playbooks and inventory can be found at the following link: <http://docs.ansible.com/>

Documentation about the Dell EMC Networking can be found at the following link: <http://ansible-dellos-docs.readthedocs.io/en/latest/>

Detailed documentation about the Dell EMC Networking Ansible roles can also be found at the following link: <https://galaxy.ansible.com/Dell-Networking/>

Detailed examples of Dell EMC Networking Ansible roles can be found at this github location: <https://github.com/Dell-Networking/ansible-dellos-examples>

## B Support and Feedback

### **Contacting Technical Support**

Support Contact Information Web: <http://Support.Dell.com>

Telephone: USA: 1-800-945-3355

### **Feedback for this document**

We encourage readers of this publication to provide feedback on the quality and usefulness of this deployment guide by sending an email to [Dell\\_Networking\\_Solutions@dell.com](mailto:Dell_Networking_Solutions@dell.com)

## About Dell EMC

Dell EMC is a worldwide leader in data center and campus solutions, which includes the manufacturing and distribution of servers, network switches, storage devices, personal computers, and related hardware and software. For more information on these and other products, please visit the Dell EMC website at <http://www.dell.com>.