# Automating Arista Network Fabric

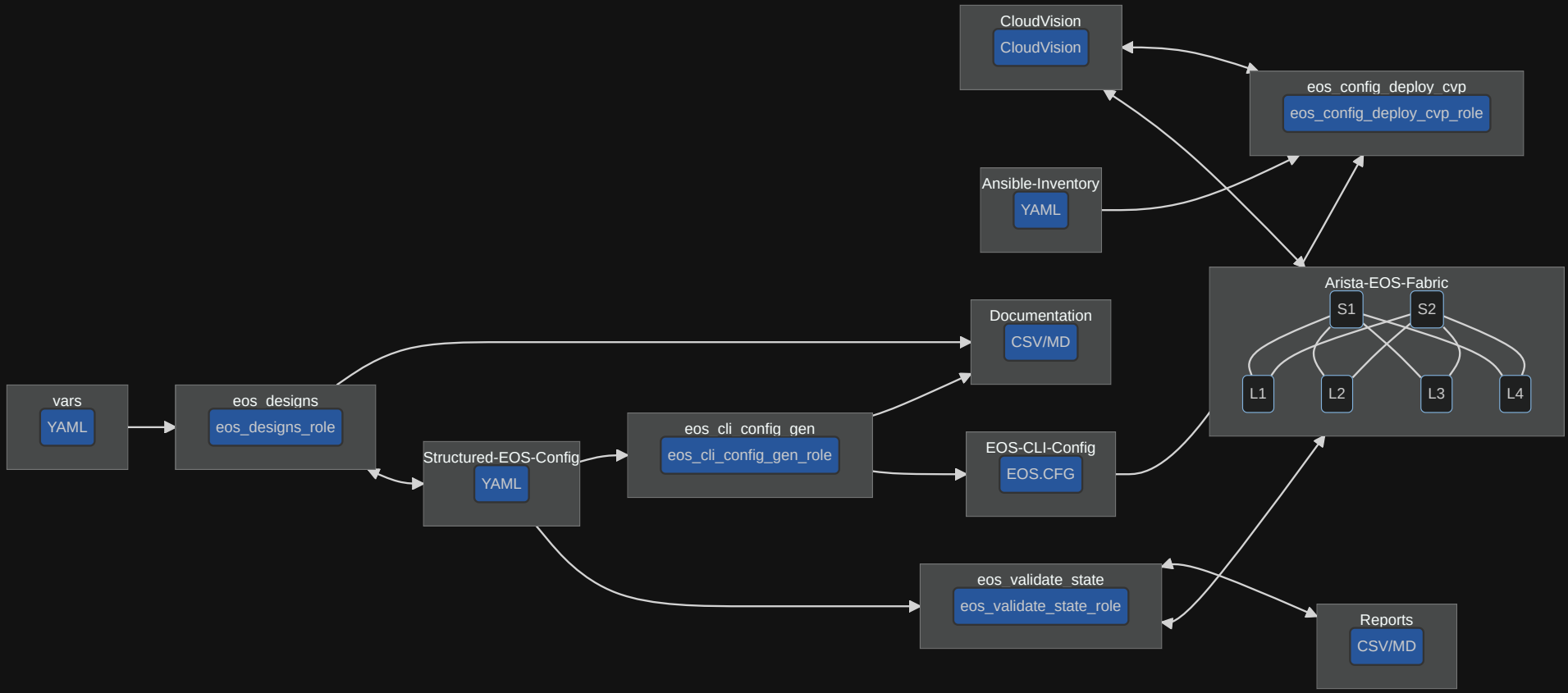
# What is Arista Validated Design (AVD)?

An extensible data model that defines Arista's Unified Cloud Network (UCN) architecture as "code"

## Benefits

- Automatic generation of documentation and validation tests
- Foundation for Infrastructure-as-Code
- Faster time to production
- Reduced risk of configuration error
- Consistent global configuration changes across the network

# Ansible AVD Collection



# The oversimplification

```yaml
# Fabric/Host variables
underlay_routing_protocol: EBGP
bgp_as: 65001
```

```yaml
# Structured configuration
router_bgp:
  as: 65001
  address_family_ipv4:
    peer_groups:
      UNDERLAY-PEERS:
        active: true
```

```
{# eos - Router BGP #}
{% if router_bgp.as is arista.avd.defi
!
router bgp {{ router_bgp.as }}
```

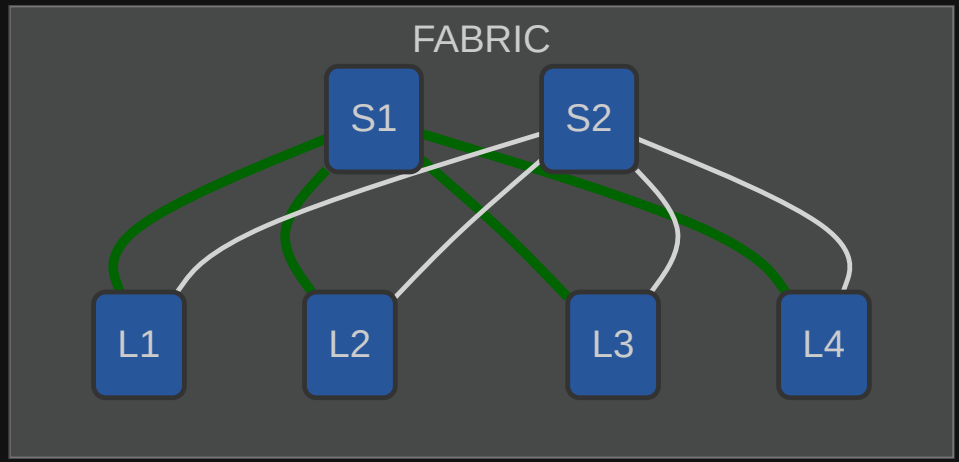
```
# EOS CLI
router bgp 65001
   address-family ipv4
      neighbor UNDERLAY-PEERS activate
```

# Group variables

# Fabric wide definitions

```yaml
# FABRIC.yml
underlay_routing_protocol: EBGP
overlay_routing_protocol: EBGP

local_users:
  ansible:
    privilege: 15
    role: network-admin
  admin:
    privilege: 15
    role: network-admin
```



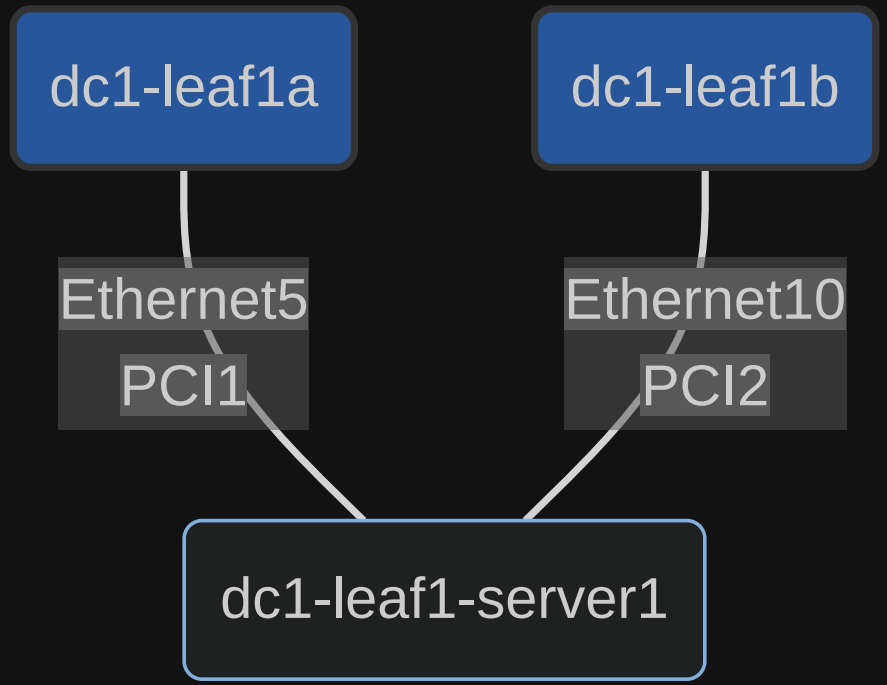
# Network services

- Tenants
- L2 & L3 services

```yaml
---
# NETWORK_SERVICES.yml
tenants:
  TENANT1:
    mac_vrf_vni_base: 10000
    vrfs:
      VRF10:
        vrf_vni: 10
        svis:
          "11":
            name: VRF10_VLAN11
            enabled: true
            ip_address_virtual: 10.10.11.1/24
    l2vlans:
      "3401":
        name: L2_VLAN3401
      "3402":
        name: L2_VLAN3402
```

# Connected endpoints

```yaml
---
# CONNECTED_ENDPOINTS.yml
servers:
  dc1-leaf1-server1:
    adapters:
    - type: server
      server_ports: [ PCI1, PCI2 ]
      switch_ports: [ Ethernet5, Ethernet10 ]
      switches: [ dc1-leaf1a, dc1-leaf1b ]
      vlans: 11-12,21-22
      native_vlan: 4092
      mode: trunk
      spanning_tree_portfast: edge
      port_channel:
        description: PortChannel dc1-leaf1-server1
        mode: active
```



# Lab

# Thank you

Documentation · GitHub · Community examples