EVPN (3-Tier Network) with ESI-LAG to Access Layer

This will define a 3-Tier network (Core/Distribution/Access) with ESI-LAG to the access layer.

Note: This is an early draft of the API for EVPN-VXLAN. Things could change prior to going GA.

Required Variables:

- `site_id` xxxxxxxx-xxxx-xxxx-xxxx-00000000000b
- `device_id` (Core-1)
- `device_id` (Core-2)
- `device_id` (Distribution-1)
- `device_id` (Distribution-2)
- `device_id` (Access-1)
- `device_id` (Access-2)
- `mac_address` (Core-1)
- `mac_address` (Core-2)
- `mac_address` (Distribution-1)
- `mac_address` (Distribution-2)
- `mac_address` (Access-1)
- `mac_address` (Access-2)

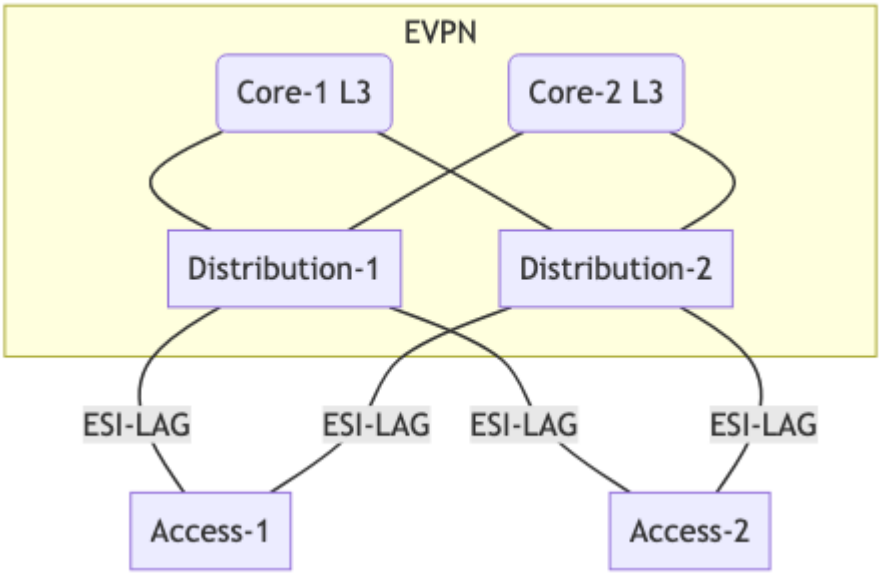
EVPN Topology:

In this topology we are doing EVPN between the Core and Distribution switches. There are two main architectual decisions in this. Do we do Centrally-Routed Bridging (CRB) or Edge-Routed Bridging (ERB).

CRB vs ERB

From the purposes of the API, the only decision is which switches receive the VRF and IRB configurations. If CRB, the core switches get the VRF and IRB configs (along with VRF Routes). For Edge Bridging, the distribution switches have the IRB and VRF configurations. From conversations with EX PLM, CRB is much more common than ERB.

Scenario 1: EVPN 3-Tier with CRB



Step 1: (Define Networks/VRFs/PortUsage)

VRF

This payload configures 2 networks (`vlan101`, `vlan102`) that go into the `internal_vrf`. The internal VRF also include a static route.

EVPN Options

We also specify the EVPN option, but these are not required.

Port Usages

In this scenario, we will define a port usage that will be the L2 trunk link over the ESI-Lag to the access layer switch.

Site Settings vs Network Template

This can also be applied to a network template and applied to the site, this example is using site settings only.

Site Settings

```
PUT:
/api/v1/sites/:site_id/setting
```


```
{
    "evpn_options": {
    "overlay": {
        "as": 65000
    },
        "underlay": {
            "as_base": 65001,
            "subnet": "10.255.240.0/20" } },
    "networks": {
        "vlan101": {
            "vlan_id": "101",
            "subnet":"192.168.101.0/24",
            "gateway": "192.168.101.1"},
        "vlan102": {
            "vlan_id": "102",
            "subnet": "192.168.102.0/24",
            "gateway": "192.168.102.1"} },
    "vrf_instances": {
        "internal_vrf": {
            "networks": ["vlan101", "vlan102"],
            "extra_routes": {"0.0.0.0/0": {"via": "192.168.192.1"} } } },
    "port_usages": {
      "distribution-access": {
            "mode": "trunk",
            "disabled": false,
            "port_network": null,
            "voip_network": null,
            "stp_edge": false,
            "all_networks": false,
            "networks": ["vlan101", "vlan102"],
            "port_auth": null,
            "speed": "auto",
            "duplex": "auto",
            "mac_limit": 0,
            "poe_disabled": true,
            "enable_qos": false,
            "storm_control": {},
            "mtu": 9200
        } }  }
```

Step 2: Apply Router ID/IRBs/VRF to Core switches

In this section, we are going to configure 3 things.

- Router ID
- IRB configurations for the L3 Gateways.
- Enable VRF for devices that need VRF

For the CBR scenario, we are going to configure the IRBs and VRFs on the Core switches.

Scenario 1: Core-1 Config

```
PUT:
/api/v1/sites/:site_id/devices/{{ Core-1_device_id }}
```

```
{
"router_id": "192.168.255.11",
"other_ip_configs": {
        "vlan101": {
            "type": "static",
            "ip": "192.168.101.2",
            "netmask": "255.255.255.0"
        },
        "vlan102": {
            "type": "static",
            "ip": "192.168.102.2",
            "netmask": "255.255.255.0"
        }
    },
"vrf_config": {
    "enabled": true
    }
}
```

Scenario 1: Core-2 Config

```
PUT:
/api/v1/sites/:site_id/devices/{{ Core-2_device_id }}
```

```
{
"router_id": "192.168.255.12",
   "other_ip_configs": {
        "vlan101": {
            "type": "static",
            "ip": "192.168.101.3",
            "netmask": "255.255.255.0"
        },
        "vlan102": {
            "type": "static",
            "ip": "192.168.102.3",
            "netmask": "255.255.255.0"
        }
    },
"vrf_config": {
    "enabled": true
    }
}
```

Step 3: Apply Router ID config to each Distribution/Spine switches.

In CRB, you will apply only a router_id to the Distribution switches.

Scenario 1: Distribution-1 Config

```
    PUT:
    /api/v1/sites/:site_id/devices/{{ Distribution-1_device_id }}
```

```
{
    "router_id": "192.168.255.13"
}
```

Scenario 1: Distribution-2 Config

```
    PUT:
    /api/v1/sites/:site_id/devices/{{ Distribution-2_device_id }}
```

```
{
    "router_id": "192.168.255.14"
}
```

Step 4: Build EVPN Topology:

This step defines which switches will participate in the EVPN and what their role is.

```
POST
/api/v1/sites/:site_id/devices/evpn_topology
```

```
{
    "overwrite": true,
    "switches": [{
            "mac": "{{ Core-1_mac_address }}",
            "role": "core"
        },
        {
            "mac": "{{ Core-2_mac_address }}",
            "role": "core"
        },
        {
            "mac": "{{ Distribution-1_mac_address }}",
            "role": "distribution"
        },
        {
            "mac": "{{ Distribution-2_mac_address }}",
            "role": "distribution"
        }
    ]
}
```

Record Output from EVPN topology

Sample OUTPUT:

```
{
    "switches": [
        {
            "mac": "{{ Core-1_mac_address }}",
            "evpn_id": 1,
            "model": "xxxxxx-24P",
            "router_id": "192.168.255.11",
            "role": "core",
            "downlinks": [
                "{{ Distribution-1_mac_address }}",
                "{{ Distribution-2_mac_address }}"],
            "downlink_ips": ["10.255.240.2", "10.255.240.4"]},
        {
            "mac": "{{ Core-2_mac_address }}",
            "evpn_id": 2,
            "model": "xxxxxxx-24P",
            "router_id": "192.168.255.12",
            "role": "access",
            "downlinks": [
                "{{ Distribution-1_mac_address }}",
                "{{ Distribution-2_mac_address }}"],
            "downlink_ips": ["10.255.240.6", "10.255.240.8"]},
        {
            "mac": "{{ Distribution-1_mac_address }}",
            "evpn_id": 3,
            "model": "xxxxxx-48P",
            "router_id": "192.168.255.14",
            "role": "distribution",
            "uplinks": [
                "{{ Core-1_mac_address }}",
                "{{ Core-2_mac_address }}"],
        },
        {
            "mac": "{{ Distribution-2_mac_address }}",
            "evpn_id": 4,
            "model": "xxxxxx-48P",
            "router_id": "192.168.255.13",
            "role": "distribution",
            "uplinks": [
                "{{ Core-1_mac_address }}",
                "{{ Core-2_mac_address }}"],
        }
    ]
}
```

Step 5: Match up the EVPN topology uplinks and downlinks.

Each switch will have uplinks,downlinks or both. Each Core switch will have evpn_downlinks Each Distribution switch will have evpn_uplinks

The EVPN Topolgy will tell you which links go where.

Make sure you match up the port to the correct port type (ge vs mge vs xe vs et)

Core-1 Port Config

```
    PUT:
    /api/v1/sites/:site_id/devices/{{ Core-1_device_id }}
```

```
{
    "port_config": {
        "ge-0/0/22-23": {
            "usage": "evpn_downlink"
        }
    }
}
```

Based on the configuration and output from the EVPN_Topology, Core-1 will have:

- `ge-0/0/22` connected to `Distribution-1`
- `ge-0/0/23` connected to `Distribution-2`

Core-2 Port Config

```
PUT:
/api/v1/sites/:site_id/devices/{{ Core-2_device_id }}
```

```
{
    "port_config": {
        "ge-0/0/22-23": {
            "usage": "evpn_downlink"
        }
    }
}
```

Based on the configuration and output from the EVPN_Topology, Core-2 will have:

- `ge-0/0/22` connected to `Distribution-1`
- `ge-0/0/23` connected to`Distribution-2`

Distribution-1 Port Config

```
    PUT:
    /api/v1/sites/:site_id/devices/{{ Distribution-1_device_id }}
```

```
{
    "port_config": {
        "ge-0/0/22-23": {
            "usage": "evpn_uplink"
            },
        "ge-0/0/1": {
            "usage": "distribution-access",
            "aggregate": true,
            "ae_idx": 1,
            "esilag": true
            },
        "ge-0/0/2": {
            "usage": "distribution-access",
            "aggregate": true,
            "ae_idx": 2,
            "esilag": true
            },
        }
    }
}
```

Based on the configuration and output from the EVPN_Topology, Distribution-1 will have:

- `ge-0/0/22` connected to `Core-1`
- `ge-0/0/23` connected to `Core-2`
- `ge-0/0/1` to connect to `Access-1` (Static ae_idx for esi-lag consistency)
- `ge-0/0/2` to connect to `Access-2` (Static ae_idx for esi-lag consistency)

Distribution-2 Port Config

```
PUT:
/api/v1/sites/:site_id/devices/{{ Distribution-2_device_id }}
```

```
{
    "port_config": {
        "ge-0/0/22-23": {
            "usage": "evpn_uplink"
            },
        "ge-0/0/1": {
            "description": "Link to Access-1",
            "usage": "distribution-access",
            "aggregate": true,
            "ae_idx": 1,
            "esilag": true
            },
        "ge-0/0/2": {
            "description": "Link to Access-2",
            "usage": "distribution-access",
            "aggregate": true,
            "ae_idx": 2,
            "esilag": true
            },
        }
    }
}
```

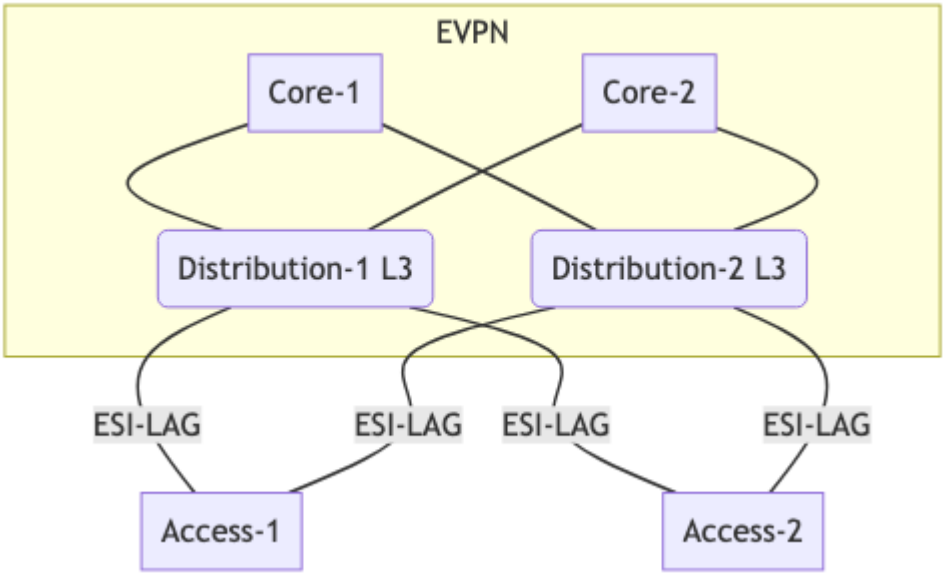
Based on the configuration and output from the EVPN_Topology, Distribution-2 will have:

- `ge-0/0/22` connected to `Core-1`
- `ge-0/0/23` connected to `Core-2`
- `ge-0/0/1` to connect to `Access-1` (Static ae_idx for esi-lag consistency)
- `ge-0/0/2` to connect to`Access-2` (Static ae_idx for esi-lag consistency)

Step 6: Access Layer Config:

With ESI-LAG, the access layer switches are configured as a normal AE. They are not aware of the EVPN or the ESI-LAG.

Scenario 2: EVPN 3-Tier with ERB



Step 1: (Define Networks/VRFs/PortUsage)

VRF

This payload configures 2 networks (`vlan101`, `vlan102`) that go into the `internal_vrf`. The internal VRF also include a static route.

EVPN Options

We also specify the EVPN option, but these are not required.

Port Usages

In this scenario, we will define a port usage that will be the L2 trunk link over the ESI-Lag to the access layer switch.

Site Settings vs Network Template

This can also be applied to a network template and applied to the site, this example is using site settings only.

Site Settings

```
PUT:
/api/v1/sites/:site_id/setting
```

```
{
    "evpn_options": {
    "overlay": {
        "as": 65000
    },
        "underlay": {
            "as_base": 65001,
            "subnet": "10.255.240.0/20" } },
    "networks": {
        "vlan101": {
            "vlan_id": "101",
            "subnet":"192.168.101.0/24",
            "gateway": "192.168.101.1"},
        "vlan102": {
            "vlan_id": "102",
            "subnet": "192.168.102.0/24",
            "gateway": "192.168.102.1"} },
    "vrf_instances": {
        "internal_vrf": {
            "networks": ["vlan101", "vlan102"],
            "extra_routes": {"0.0.0.0/0": {"via": "192.168.192.1"} } } },
    "port_usages": {
      "distribution-access": {
            "mode": "trunk",
            "disabled": false,
            "port_network": null,
            "voip_network": null,
            "stp_edge": false,
            "all_networks": false,
            "networks": ["vlan101", "vlan102"],
            "port_auth": null,
            "speed": "auto",
            "duplex": "auto",
            "mac_limit": 0,
            "poe_disabled": true,
            "enable_qos": false,
            "storm_control": {},
            "mtu": 9200
        } }  }
```

Step 2: Apply Router ID/IRBs/VRF to Distribution Switches.

In this section, we are going to configure 3 things.

- Router ID
- IRB configurations for the L3 Gateways.
- Enable VRF for devices that need VRF

In ERB we will apply these to the Distribution Switches.

Distribution-1 Config

```
    PUT:
    /api/v1/sites/:site_id/devices/{{ Distribution-1_device_id }}
```

```
{
"router_id": "192.168.255.11",
"other_ip_configs": {
        "vlan101": {
            "type": "static",
            "ip": "192.168.101.2",
            "netmask": "255.255.255.0"
        },
        "vlan102": {
            "type": "static",
            "ip": "192.168.102.2",
            "netmask": "255.255.255.0"
        }
    },
"vrf_config": {
    "enabled": true
    }
}
```

Distribution-2 Config

```
    PUT:
    /api/v1/sites/:site_id/devices/{{ Distribution-2_device_id }}
```

```
{
"router_id": "192.168.255.12",
   "other_ip_configs": {
        "vlan101": {
            "type": "static",
            "ip": "192.168.101.3",
            "netmask": "255.255.255.0"
        },
        "vlan102": {
            "type": "static",
            "ip": "192.168.102.3",
            "netmask": "255.255.255.0"
        }
    },
"vrf_config": {
    "enabled": true
    }
}
```

Step 3: Apply Router ID config to each Core switches.

In ERB, you will apply only a router_id to the Core switches.

Core-1 Config

```
    PUT:
    /api/v1/sites/:site_id/devices/{{ Core-1_device_id }}
```

```
{
    "router_id": "192.168.255.13"
}
```

Core-2 Config

```
    PUT:
    /api/v1/sites/:site_id/devices/{{ Core-2_device_id }}
```

```
{
    "router_id": "192.168.255.14"
}
```

Step 4: Build EVPN Topology:

This step defines which switches will participate in the EVPN and what their role is.

Build EVPN Topology

```
POST
/api/v1/sites/:site_id/devices/evpn_topology
```

```
{
    "overwrite": true,
    "switches": [{
            "mac": "{{ Core-1_mac_address }}",
            "role": "core"
        },
        {
            "mac": "{{ Core-2_mac_address }}",
            "role": "core"
        },
        {
            "mac": "{{ Distribution-1_mac_address }}",
            "role": "distribution"
        },
        {
            "mac": "{{ Distribution-2_mac_address }}",
            "role": "Distribution"
        }
    ]
}
```

Record Output from EVPN topology

```
{
    "switches": [
        {
            "mac": "{{ Core-1_mac_address }}",
            "evpn_id": 1,
            "model": "xxxxxx-24P",
            "router_id": "192.168.255.11",
            "role": "core",
            "downlinks": [
                "{{ Distribution-1_mac_address }}",
                "{{ Distribution-2_mac_address }}"],
            "downlink_ips": ["10.255.240.2", "10.255.240.4"]},
        {
            "mac": "{{ Core-2_mac_address }}",
            "evpn_id": 2,
            "model": "xxxxxxx-24P",
            "router_id": "192.168.255.12",
            "role": "access",
            "downlinks": [
                "{{ Distribution-1_mac_address }}",
                "{{ Distribution-2_mac_address }}"],
            "downlink_ips": ["10.255.240.6", "10.255.240.8"]},
        {
            "mac": "{{ Distribution-1_mac_address }}",
            "evpn_id": 3,
            "model": "xxxxxx-48P",
            "router_id": "192.168.255.14",
            "role": "distribution",
            "uplinks": [
                "{{ Core-1_mac_address }}",
                "{{ Core-2_mac_address }}"],
        },
        {
            "mac": "{{ Distribution-1_mac_address }}",
            "evpn_id": 4,
            "model": "xxxxxx-48P",
            "router_id": "192.168.255.13",
            "role": "distribution",
            "uplinks": [
                "{{ Core-1_mac_address }}",
                "{{ Core-2_mac_address }}"],
        }
    ]
}
```

Step 5: Match up the EVPN topology uplinks and downlinks.

Each switch will have uplinks,downlinks or both. Each Core switch will have evpn_downlinks Each Distribution switch will have evpn_uplinks

The EVPN Topolgy will tell you which links go where.

Core-1 Ports

```
    PUT:
    /api/v1/sites/:site_id/devices/{{ Core-1_device_id }}
```

```
{
    "port_config": {
        "ge-0/0/22-23": {
            "usage": "evpn_downlink"
        }
    }
}
```

Based on the configuration and output from the EVPN_Topology, Core-1 will have:

- `ge-0/0/22` connected to `Distribution-1`
- `ge-0/0/23` connected to `Distribution-2`

Core-2 Ports

```
PUT:
/api/v1/sites/:site_id/devices/{{ Core-2_device_id }}
```

```
{
    "port_config": {
        "ge-0/0/22-23": {
            "usage": "evpn_downlink"
        }
    }
}
```

Based on the configuration and output from the EVPN_Topology, Core-2 will have:

- `ge-0/0/22` connected to `Distribution-1`
- `ge-0/0/23` connected to `Distribution-2`

Distribution-1 Ports

```
PUT:
/api/v1/sites/:site_id/devices/{{ Distribution-1_device_id }}
```

```
{
    "port_config": {
        "ge-0/0/22-23": {
            "usage": "evpn_uplink"
            },
        "ge-0/0/1": {
            "description": "Link to Access-1",
            "usage": "distribution-access",
            "aggregate": true,
            "ae_idx": 1,
            "esilag": true
            },
        "ge-0/0/2": {
            "description": "Link to Access-2",
            "usage": "distribution-access",
            "aggregate": true,
            "ae_idx": 2,
            "esilag": true
            },
        }
    }
}
```

Based on the configuration and output from the EVPN_Topology, Distribution-1 will have:

- `ge-0/0/22` connected to `Core-1`
- `ge-0/0/23` connected to `Core-2`
- `ge-0/0/1` to connect to `Access-1` (Static ae_idx for esi-lag consistency)
- `ge-0/0/2` to connect to `Access-2` (Static ae_idx for esi-lag consistency)

Distribution-2 Ports

```
PUT:
/api/v1/sites/:site_id/devices/{{ Distribution-2_device_id }}
```

```
{
    "port_config": {
        "ge-0/0/22-23": {
            "usage": "evpn_uplink"
            },
        "ge-0/0/1": {
            "description": "Link to Access-1",
            "usage": "distribution-access",
            "aggregate": true,
            "ae_idx": 1,
            "esilag": true
            },
        "ge-0/0/2": {
            "description": "Link to Access-2",
            "usage": "distribution-access",
            "aggregate": true,
            "ae_idx": 2,
            "esilag": true
            },
        }
    }
}
```

Based on the configuration and output from the EVPN_Topology, Distribution-2 will have:

- `ge-0/0/22` connected to `Core-1`
- `ge-0/0/23` connected to `Core-2`
- `ge-0/0/1` to connect to `Access-1` (Static ae_idx for esi-lag consistency)
- `ge-0/0/2` to connect to `Access-2` (Static ae_idx for esi-lag consistency)

Step 6: Access Layer Config:

With ESI-LAG, the access layer switches are configured as a normal AE. They are not aware of the EVPN or the ESI-LAG.