

Migrating legacy Networks into ACI/DCNM with Ansible/AWX

DevNet Partner University 2020

Alexander Papenburg, Stephan Grunske

Technical Solutions Architect, Customer Success Specialist DC

#NoTwitter

Agenda



Scope



Szenario



Intro AWX



DCNM



ACI

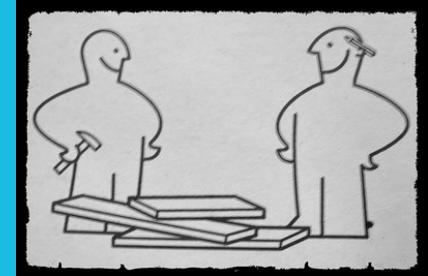
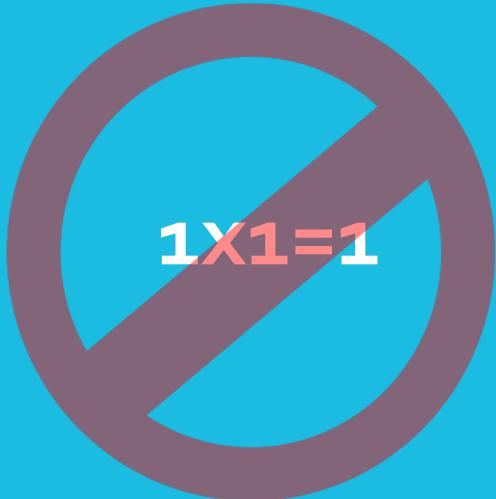


Questions

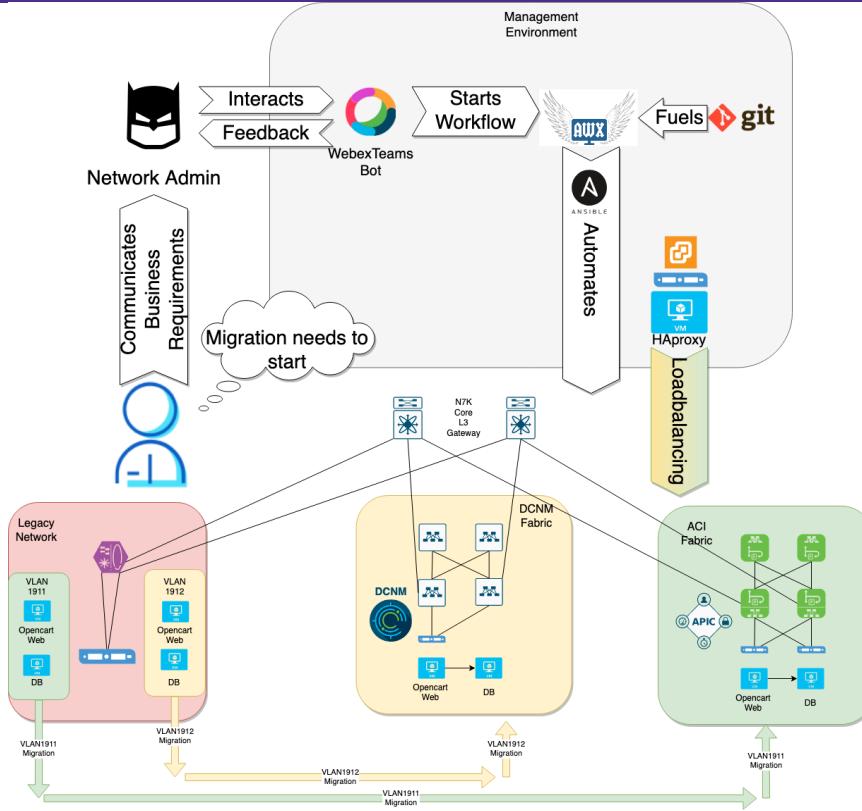
Scope

- With the shift to a fabric design most customers are going to face the challenge of migrating their existing workload, which is also the perfect opportunity to show the value of network automation. Therefore we created two demos which are easy to use, understand and replicate. After the session you will have a good starting point to create a demo or a real migration on your own. We are going to demonstrate how to move a legacy network including virtual workload into ACI/DCNM leveraging AWX, Ansible and Webex Teams to configure different technologies like NXOS, ACI, DCNM and VMWare VCenter.

What is this Session



Szenario



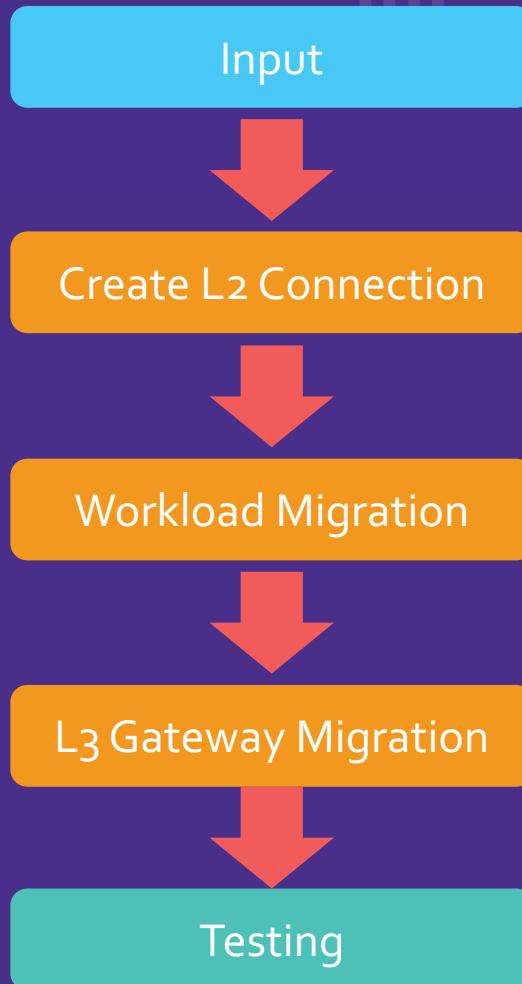
Basic Workflow

Critical Input Parameters ACI:

- Tenant Name
- VLAN ID

Critical Input Parameters DCNM:

- VLAN ID



Intro Ansible Tower / AWX



Red Hat
Commercially
Supported /
OpenSource



Run Ansible
playbooks and
workflows from GUI



Schedule playbooks
for specific time(s)



Manage playbooks
via SCM (like git)



Manage inventories
and variables via GUI



RBAC



Logging



Store credentials
securely

DEMO AWX

DCNM



DCNM API

Overview - Cisco DCNM REST X +

developer.cisco.com/docs/data-center-network-manager/

DEVNET Discover Technologies Community Support Events New Announcement

Documentation > Cisco DCNM REST API Reference Guide, Release 11.3(1)

DCNM Release: 11.3(1) Documentation

Cisco DCNM API Reference Guide Release 11.3(1)

- Overview
- Guidelines and Limitations
- Cisco DCNM Classic LAN REST API
- Cisco DCNM LAN Fabric REST API
- Switch Roles
- Cisco DCNM Media Controller REST API
- Cisco DCNM SAN Management REST API
- Enabling External Alarms

Overview

Cisco Data Center Network Manager (DCNM) is a management system for the Cisco Programmable Fabric. It provides a comprehensive feature-set that meets the routing, switching, and storage administration needs of the datacenter. DCNM streamlines the provisioning for the Programmable Fabric and monitors the SAN and LAN environments.

The Cisco Fabric Automation REST APIs for third party applications enables you to program and automate the Cisco Fabric. The REST API supports POAP (Power On Auto Provisioning), Auto Config and API operations can also be performed using the DCNM GUI as DCNM uses these REST APIs.

The Authentication REST APIs can be used by an external application to authenticate itself to the Fabric Automation cluster. After calling logon to get the token, all the subsequent REST API calls must include the Token field with the token in the HTTPS header. From Release 10.0(1), by default, the Cisco DCNM REST API uses the token-based authentication.

For additional information on Cisco DCNM, its utilization, deployment and more, refer to the Cisco DCNM documentation at <https://www.cisco.com/c/en/us/support/cloud-systems-management/data-center-network-manager/dc-dcnm>.

DCNM API Docs Not Secure | 10.49.164.189/api-docs/#/Control_fabrics/getFabric

GET /control/fabrics/{fabricName}

Parameters

Parameter	Value	Description	Parameter Type	Data Type
fabricName	MULTI_SITE		path	string

Response Messages

HTTP Status Code	Reason	Response Model	Headers
200	Fetch the given fabric details		
401	Unauthorized access to API		

Try it out! Hide Response

Curl

```
curl -X GET --header 'Accept: application/json' 'https://10.49.164.189/rest/control/fabrics/MULTI_SITE'
```

Request URL

https://10.49.164.189/rest/control/fabrics/MULTI_SITE

Response Body

```
{"fabricType": "MFD", "fabricTypeFriendly": "Multi-Fabric Domain", "fabricTechnology": "VXLANFabric", "fabricTechnologyFriendly": "VXLAN Fabric", "provisionMode": "DCNMTopdown", "deviceType": "n9K", "replicationMode": "IngressReplication", "templateName": "MSD_Fabric_11_1", "nvPairs": [{"FID": "MSD", "vrf_extension_template": "Default_VRF_Extension_Universal", "FABRIC_TYPE": "MFD", "DCI_SUBNET_RANGE": "172.30.0.0/22", "TOR_AUTO_DEPLOY": "false", "FABRIC_NAME": "MULTI_SITE", "network_extension_template": "Default_Network_Extension_Universal", "L3_PARTITION_ID_RANGE": "50000-59000", "RP_SERVER_IP": "", "MS_UNDERLAY_AUTOCONFIG": "true", "LOOPBACK100_IP_RANGE": "172.30.100.0/24", "ANYCAST_GW_MAC": "2020.0000.00aa", "L3_PARTITION_ID": "50000", "RP_SERVER_IP": "172.30.100.1", "MS_UNDERLAY_AUTOCONFIG": "true", "LOOPBACK100_IP": "172.30.100.1", "ANYCAST_GW_MAC": "2020.0000.00aa"}]
```

Response Code

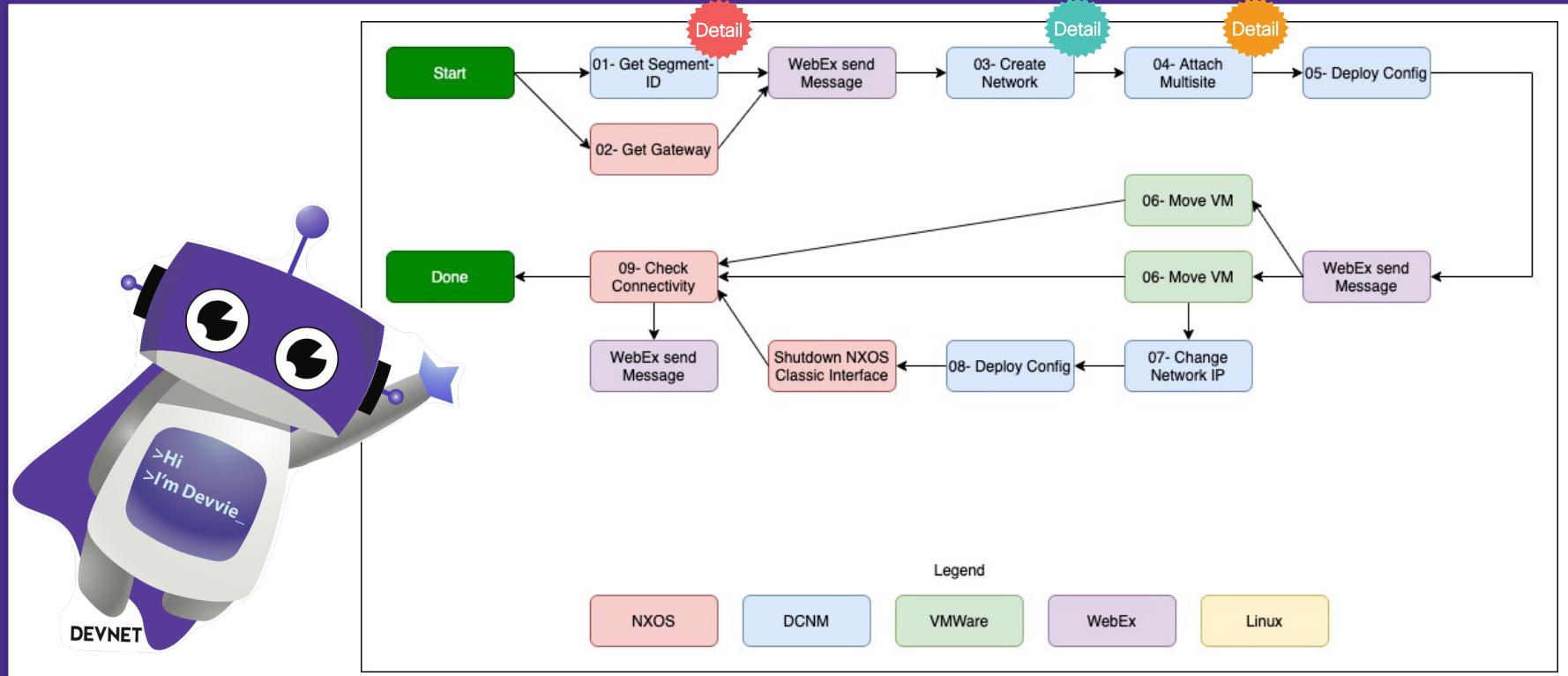
DCNM API



Overview DCNM



Workflow DCNM + How to get there



Demo DCNM



DCNM – Get segment-ID

```
24  - name: 01 - Get Segment ID
25    hosts: all
26    connection: local
27    gather_facts: no
28
29  tasks:
30
31    - name: Get
32      uri:
33        url: https://{{ dcnmhost }}/rest/managed-pool/fabrics/{{ multisite_fabric_name }}/segments/ids
34        headers:
35          | Dcnm-token: "{{ dcnmapitoken }}"
36          | Content-Type: "application/json"
37        method: POST
38        body_format: json
39        body: null
40        status_code: 200
41        return_content: yes
42        validate_certs: no
43        register: qResult
44
45    - debug:
46      var: qResult
47
48    - set_stats:
49      data:
50        | segment_id: "{{ qResult.json.segmentId }}"
51        per_host: no
```



DCNM – Create network

```
32  - name: Play 1 Create Network
33    hosts: all
34    connection: local
35    gather_facts: no
36
37  tasks:
38
39    - set_fact:
40      config: "{\"gatewayIpAddress\":\"{{ gateway_ipv4_address }}\", \"gatewayIpV6Address\":\"\", \"vlanName\":\"{{ vlan_name }}\", \"intfDescript
41
42    - name: Create New Network
43      uri:
44        url: https://{{ dcnmhost }}/rest/top-down/fabrics/{{ multisite_fabric_name }}/networks
45        headers:
46          Dcmn-token: "{{ dcnmapitoken }}"
47          Content-Type: "application/json"
48        method: POST
49        body_format: json
50        body:
51          {
52            "fabric": "{{ multisite_fabric_name }}",
53            "vrf": "{{ vrf_name }}",
54            "networkName": "{{ network_name }}",
55            "displayName": "{{ network_name }}",
56            "networkId": "{{ segment_id }}",
57            "networkTemplateConfig": "{{ config | string }}",
58            "networkTemplate": "Default_Network_Universal",
59            "networkExtensionTemplate": "Default_Network_Extension_Universal",
60            "source": null,
61            "serviceNetworkTemplate": null
62          }
63        status_code: 200
64        return_content: yes
65        validate_certs: no
66
67    - set_stats:
68      data:
69        webextext: "Network {{ network_name }} with VLAN{{ vlan_id }} created."
70        per_host: no
```

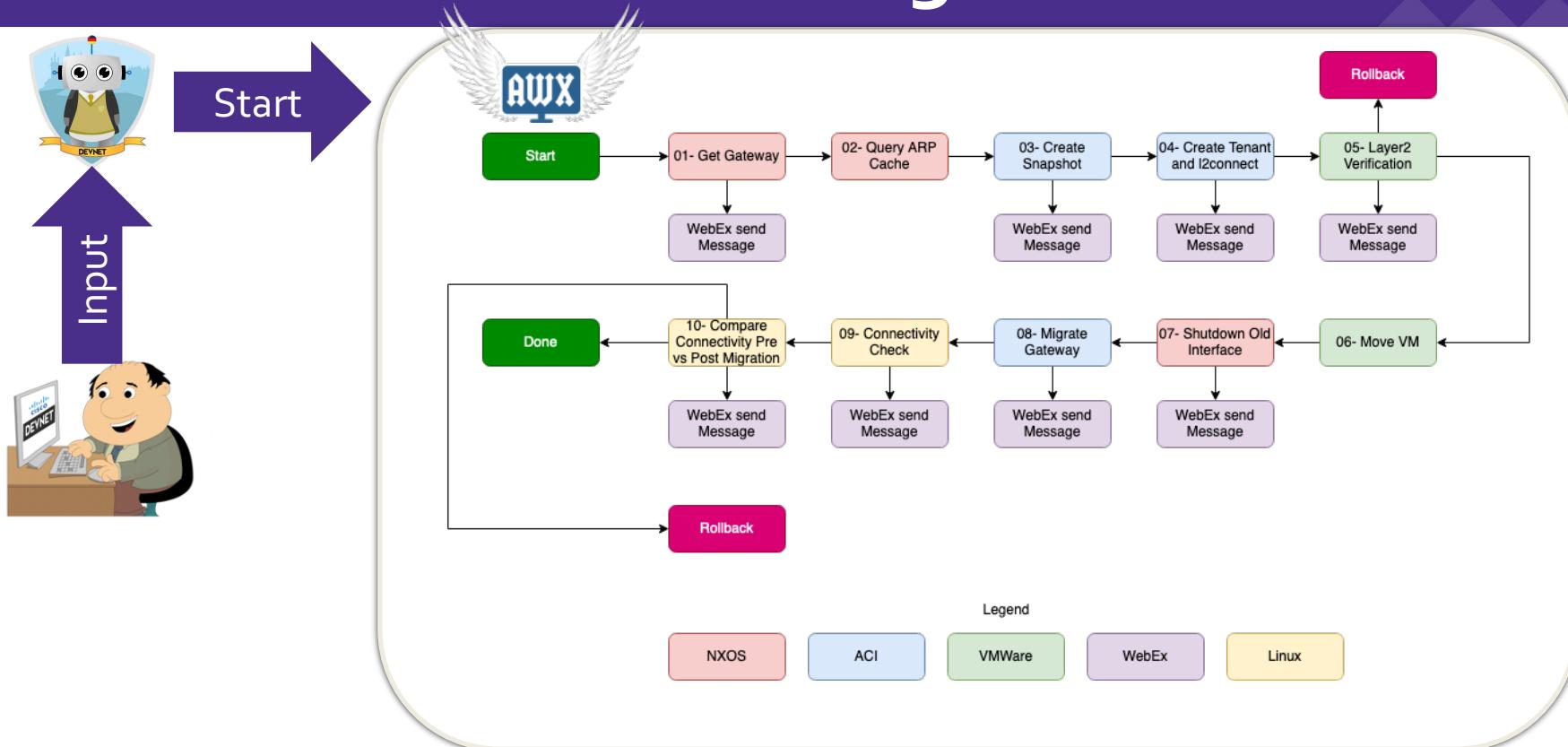


DCNM – Attach Multisite

```
29 - name: Attach or detach networks to fabric
30   hosts: all
31   connection: local
32   gather_facts: no
33
34
35 tasks:
36
37 - name: Modify network on nonVPC devices
38   uri:
39     url: https://{{ dcnmhost }}/rest/top-down/fabrics/{{ multisite_fabric_name }}/networks/attachments
40   headers:
41     Dcnm-token: "{{ dcnmapioken }}"
42     Content-Type: "application/json"
43   method: POST
44   body_format: json
45   body:
46     [
47       {
48         "networkName": "{{ network_name }}",
49         "lanAttachList": [
50           {
51             "fabric": "{{ item.fn }}",
52             "networkName": "{{ network_name }}",
53             "serialNumber": "{{ item.sn }}",
54             "vlan": "{{ vlan_id }}",
55             "dot1QVlan": 1,
56             "untagged": false,
57             "freeformConfig": "",
58             "deployment": "{{ deployment }}",
59             "extensionValues": "{{ \"\\\"MULTISITE_CONN\\\":\\\"{{{MULTISITE_CONN}}}\\\":[]\\\"}}",
60             "switchPorts": "{{ item.in }}",
61             "detachSwitchPorts": "",
62             "instanceValues": ""
63           }
64         ]
65       }
66     ]
67   status_code: 200
68   return_content: yes
69   validate_certs: no
70   with_items: "{{ nonvpc_devices }}"
```

ACI

Workflow ACI + How to get there



Demo ACI

Facts



22 Playbooks



100+ Tasks



2 Webex Bots



**4 Different Cisco
Technologies**

Webex Teams

DCNM

ACI

NXOS

Ressources



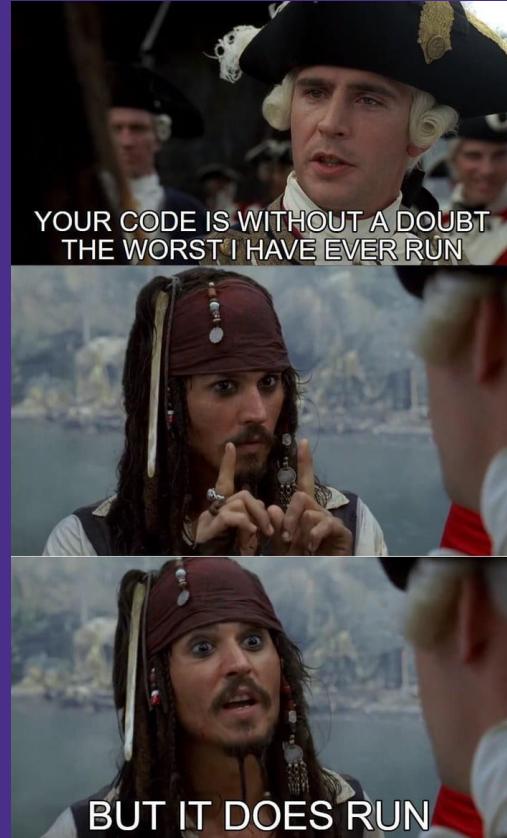
[Session Repo \(Playbooks and PDF\)](#)



[Stephan Grunske](#)



[Alexander Papenburg](#)



Fragen?

