

# Ansible과 NXOS

최혜연 프로, Meraki Technical Solutions Specialist

# 순서

- Nexus & NXOS 소개
- Ansible for NXOS 모듈
- Ansible for NXOS 실습
- 미션



# Nexus 스위치



차세대 데이터센터 네트워크를  
실현하기 위한 플랫폼

Datacenter 아키텍처 지원

Cisco의 자체 ASIC

고밀도의 다양한 속도/타입을 지원

낮은 지연 시간과 높은 전력 효율

뛰어난 개방형 API



# NX-OS 소개

**고가용성 지원**  
Enhanced ISSU



**보안**

First Hop 보안 기능 지원  
트래픽 암호화 지원(MacSec/CloudSec)



**가시성 및 분석**

HW Streaming Telemetry 지원  
Netflow 지원



**인프라**

NX-SDK 지원  
지능형 서비스 지원-NDB  
FX 플랫폼 기반의 FCoE, FC지원

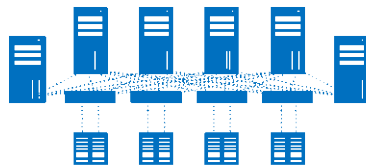


**프로그래머블 패브릭 지원**

VXLAN EVPN multi-site 지원  
VXLAN OAM, Tenant Multicast지원  
Segment Routing L3 EVPN 지원  
DCNM 기반 관리 지원



**Cisco NX-OS**



**자동화**

DCNM 지원  
Nexus 구성 관리 모델링 지원  
(Puppet/Chef/Ansible)  
업계 표준 구성 모델링 지원  
(OpenConfig /IETF YANG)



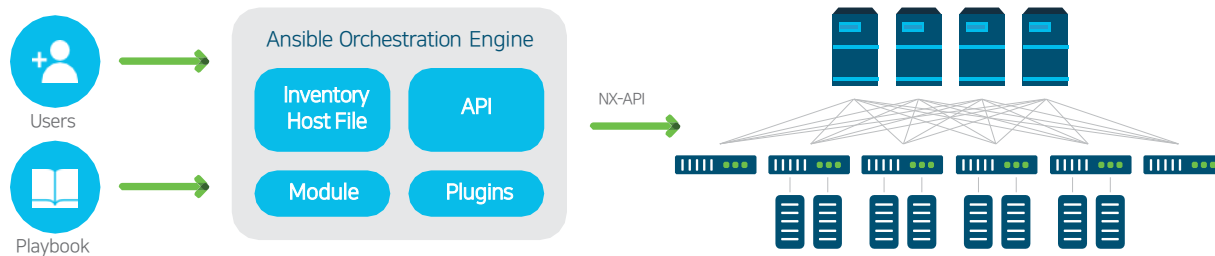
# Cisco NX-OS Programmability

## 요구 사항

- ✓ 서버 환경에서 사용하는 구성 관리 구조인 Ansible을 네트워크 구성 관리로 확장 적용

## 솔루션

- ✓ 서버 자원 증감에 따른 네트워크 구성 관리 자동화 실현
- ✓ 구성순서의 코드화로 운용 관리 간소화
- ✓ 휴먼 에러 방지



# Ansible for NXOS 소개

# Ansible Cisco.Nxos Collection 모듈

Documentation

Namespace

Collections in the Cisco Namespace

Cisco.Aci

Cisco.Asa

Cisco.Dnac

Cisco.Intersight

Cisco.Ios

Cisco.Iosxr

Cisco.Ise

Cisco.Meraki

Cisco.Mso

Cisco.Nxos

Description

Plugin Index

Cisco.Ucs

Collections in the Cloud Namespace

Collections in the Cloudscale\_ch Namespace

Collections in the Community Namespace

Collections in the Containers Namespace

Collections in the Cyberark Namespace

Collections in the Dellemc Namespace

Collections in the F5networks Namespace

Collections in the Fortinet Namespace

Collections in the Frr Namespace

Collections in the Gluster Namespace

Collections in the Google Namespace

Modules

- `nxos_aaa_server` module - Manages AAA server global configuration.
- `nxos_aaa_server_host` module - Manages AAA server host-specific configuration.
- `nxos_acl_interfaces` module - ACL interfaces resource module
- `nxos_acls` module - ACLs resource module
- `nxos_banner` module - Manage multiline banners on Cisco NXOS devices
- `nxos_bfd_global` module - Bidirectional Forwarding Detection (BFD) global-level configuration
- `nxos_bfd_interfaces` module - BFD interfaces resource module
- `nxos_bgp_address_family` module - BGP Address Family resource module.
- `nxos_bgp_global` module - BGP Global resource module.
- `nxos_bgp_neighbor_address_family` module - BGP Neighbor Address Family resource module.
- `nxos_bgp_templates` module - BGP Templates resource module.
- `nxos_command` module - Run arbitrary command on Cisco NXOS devices
- `nxos_config` module - Manage Cisco NXOS configuration sections
- `nxos_devicealias` module - Configuration of device alias for Cisco NXOS MDS Switches.
- `nxos_evpn_global` module - Handles the EVPN control plane for VXLAN.
- `nxos_evpn_vni` module - Manages Cisco EVPN VXLAN Network Identifier (VNI).
- `nxos_facts` module - Gets facts about NX-OS switches
- `nxos_fc_interfaces` module - Fc Interfaces resource module
- `nxos_feature` module - Manage features in NX-OS switches.
- `nxos_file_copy` module - Copy a file to a remote NXOS device.
- `nxos_gir` module - Trigger a graceful removal or insertion (GIR) of the switch.
- `nxos_gir_profile_management` module - Create a maintenance-mode or normal-mode profile for GIR.
- `nxos_hostname` module - Hostname resource module.
- `nxos_hsrp` module - Manages HSRP configuration on NX-OS switches.
- `nxos_hsrp_interfaces` module - HSRP interfaces resource module
- `nxos_igmp` module - Manages IGMP global configuration.
- `nxos_igmp_interface` module - Manages IGMP interface configuration.
- `nxos_igmp_snooping` module - Manages IGMP snooping global configuration.
- `nxos_install_os` module - Set boot options like boot, kickstart image and issu.
- `nxos_interfaces` module - Interfaces resource module
- `nxos_l2_interfaces` module - L2 interfaces resource module
- `nxos_l3_interfaces` module - L3 interfaces resource module

- Cisco NXOS 장비용 Ansible 네트워크 Collection
- 사용법 세부 정보, 필수 변수, 기본 변수 등 확인 가능
- RedHat이 유지 관리 (Cisco 지원)
- **`nxos_command` module** : Cisco NX-OS 스위치에서 명령어 실행할 수 있는 모듈
- **`nxos_facts` module**: NX-OS 스위치에 대한 정보를 가져오는 모듈

<https://docs.ansible.com/ansible/latest/collections/cisco/nxos/index.html>

# Ansible nxos\_facts parameters

Parameter	Comments
<code>available_network_resources</code> boolean	true로 설정하면 리소스 모듈을 사용할 수 있는 네트워크 리소스 목록이 제공됩니다.  Choices: •false ← (default) •true
<code>gather_network_resources</code> list / <code>elements=string</code>	이 인수는 주어진 subset에 대한 설정 사항만 수집합니다. 더 많은 subset을 포함하려면 값들을 목록으로 지정할 수 있습니다. 또한 특정 subset을 수집하지 않도록 지정하려면 해당 subset 앞에 !를 사용할 수 있습니다.  subsets all, bfd_interfaces, lag_interfaces, telemetry, vlans, lacp, lacp_interfaces, interfaces, l3_interfaces, l2_interfaces, lldp_global, acls, acl_interfaces, ospfv2, ospfv3, ospf_interfaces, bgp_global, bgp_address_family, route_maps, prefix_lists, logging_global, ntp_global, snmp_server, hostname
<code>gather_subset</code> list / <code>elements=string</code>	이 인수는 주어진 subset에 대해서만 operation 정보를 수집합니다. 이 인수에 사용할 수 있는 값은 all, hardware, config, legacy, interfaces, min입니다. 더 많은 subset을 포함하도록 값 목록을 지정할 수 있습니다. 또한 특정 subset을 수집하지 않도록 지정하려면 해당 subset 앞에 !를 사용할 수 있습니다.  Default: ["min"]



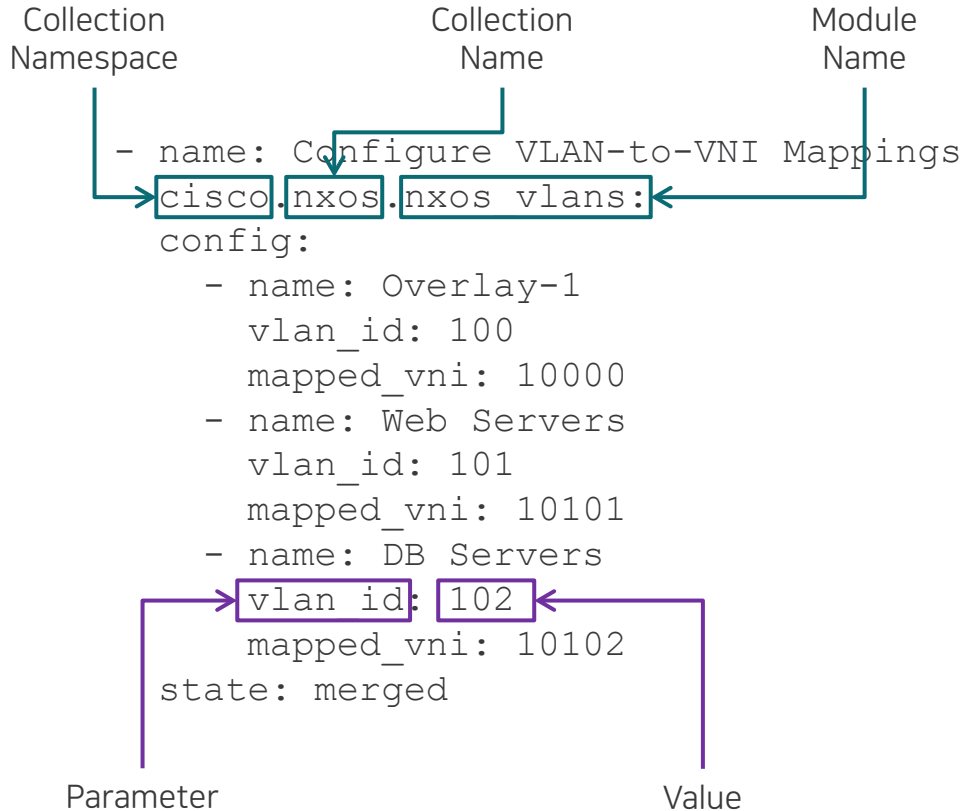
# Ansible nxos\_facts 모듈 실행 결과값

<b>ansible_net_all_ipv4_addresses</b> list / elements=string	All IPv4 addresses configured on the device <b>Returned:</b> when interfaces is configured
<b>ansible_net_all_ipv6_addresses</b> list / elements=string	All IPv6 addresses configured on the device <b>Returned:</b> when interfaces is configured
<b>ansible_net_api</b> string	The name of the transport <b>Returned:</b> always
<b>ansible_net_config</b> string	The current active config from the device <b>Returned:</b> when config is configured
<b>ansible_net_filesystems</b> list / elements=string	All file system names available on the device <b>Returned:</b> when hardware is configured
<b>ansible_net_gather_network_resources</b> list / elements=string	The list of fact for network resource subsets collected from the device <b>Returned:</b> when the resource is configured
<b>ansible_net_hostname</b> string	The configured hostname of the device <b>Returned:</b> always
<b>ansible_net_image</b> string	The image file the device is running <b>Returned:</b> always
<b>ansible_net_interfaces</b> dictionary	A hash of all interfaces running on the system <b>Returned:</b> when interfaces is configured
<b>ansible_net_license_hostid</b> string	The License host id of the device <b>Returned:</b> always
<b>ansible_net_memfree_mb</b> integer	The available free memory on the remote device in Mb <b>Returned:</b> when hardware is configured
<b>ansible_net_memtotal_mb</b> integer	The total memory on the remote device in Mb <b>Returned:</b> when hardware is configured

<b>ansible_net_model</b> string	The model name returned from the device <b>Returned:</b> always
<b>ansible_net_neighbors</b> dictionary	The list of LLDP and CDP neighbors from the device. If both, CDP and LLDP neighbor data is present on one port, CDP is preferred. <b>Returned:</b> when interfaces is configured
<b>ansible_net_python_version</b> string	The Python version Ansible controller is using <b>Returned:</b> always
<b>ansible_net_serialnum</b> string	The serial number of the remote device <b>Returned:</b> always
<b>ansible_net_version</b> string	The operating system version running on the remote device <b>Returned:</b> always
<b>fan_info</b> dictionary	A hash of facts about fans in the remote device <b>Returned:</b> when legacy is configured
<b>hostname</b> dictionary	The configured hostname of the remote device <b>Returned:</b> when legacy is configured
<b>interfaces_list</b> dictionary	The list of interface names on the remote device <b>Returned:</b> when legacy is configured
<b>kickstart</b> string	The software version used to boot the system <b>Returned:</b> when legacy is configured
<b>module</b> string	A hash of facts about the modules in a remote device <b>Returned:</b> when legacy is configured
<b>platform</b> string	The hardware platform reported by the remote device <b>Returned:</b> when legacy is configured
<b>power_supply_info</b> string	A hash of facts about the power supplies in the remote device <b>Returned:</b> when legacy is configured
<b>vlan_list</b> list / elements=string	The list of VLAN IDs configured on the remote device <b>Returned:</b> when legacy is configured

# Ansible Modules

- 항상 FQCN(정규화된 컬렉션 이름) 모듈명 사용
- 모듈에는 구성 의도를 정의하는 값이 할당된 매개변수 필요



# Ansible nxos\_facts를 사용한 playbook 실행

- ubuntu 서버 접속
- DevNet\_Korea/01\_NXOS\_facts 폴더로 이동
- -i 옵션으로 inventory 파일 (파일명:hosts) 지정

```
devnet@devnet-virtual-machine:~/DevNet_Korea/01_NXOS_facts$ cat hosts
[nexus:vars]
ansible_connection= ansible.netcommon.network_cli
ansible_user = admin
ansible_password = Cisc0l23!@#
ansible_network_os = cisco.nxos.nxos
ansible_network_cli_ssh_type=paramiko

[nexus]
N9K ansible_host=192.168.14.12
```

## vi 파일 저장 및 종료하기

모드	명령어	설명
마지막 행 모드	q	vi에서 작업한것이 없을때 vi 종료합니다.
	q!	작업한 내용을 저장하지 않고 종료합니다.
	:w[파일명]	작업한 내용을 저장만 한다. 파일명을 지정하면 새 파일로 저장합니다.
	:wq, :wq!	작업한 내용을 저장하고 vi를 종료합니다.
명령 모드	ZZ (대문자)	작업한 내용을 저장하고 vi를 종료합니다.

내용을 편집한 후 vi를 종료하려면 명령모드나 마지막 행 모드에서 명령을 입력해야 합니다. 마지막 행 모드로 가려면 : 을 누르면 됩니다. 명령모드에서 파일의 저장과 종료를 동시에 수행하려면 대문자ZZ를 누르시면 파일을 저장한 후 vi를 바로 종료합니다.

## vi 입력모드로 전환하기

명령어	설명
I	현재 커서 앞에 입력합니다.
a	현재 커서 뒤에 입력합니다.
O	커서가 위치한 행의 다음 행에 입력합니다.
[I(문자 I)]	커서가 위치한 행의 첫 줄점으로 이동하여 입력합니다.
A	커서가 위치한 행의 마지막 줄점으로 이동하여 입력합니다.
O	커서가 위치한 행의 이전 행에 입력합니다.

# Ansible NXOS 모듈 Connection 설정

## Connections available

	CLI	NX-API
Protocol	SSH	HTTP(S)
Credentials	uses SSH keys / SSH-agent if present accepts <code>-u myuser -k</code> if using password	uses HTTPS certificates if present
Indirect Access	by a bastion (jump host)	by a web proxy
Connection Settings	<code>ansible_connection:</code> <code>ansible.netcommon.network_cli</code>	<code>ansible_connection:</code> <code>ansible.netcommon.httpapi</code>
Enable Mode (Privilege Escalation) supported as of 2.5.3	supported: use <code>ansible_become: true</code> with <code>ansible_become_method: enable</code> and <code>ansible_become_password:</code>	not supported by NX-API
Returned Data Format	<code>stdout[0].</code>	<code>stdout[0].messages[0].</code>

The `ansible_connection: local` has been deprecated. Please use `ansible_connection: ansible.netcommon.network_cli` or `ansible_connection: ansible.netcommon.httpapi` instead.



# Ansible nxos\_facts를 사용한 playbook 실행

```
    five_seconds : 11,
    "five_seconds_interrupt": 0,
    "one_minute": 5
  },
},
"net_fan_info": [],
"net_features_enabled": [
  "telnet",
  "bgp"
],
"net_filesystems": [
  "bootflash:"
],
},
"net_gather_network_resources": [],
"net_gather_subset": [
  "config",
  "legacy",
  "features",
  "interfaces",
  "default",
  "hardware"
],
"net_hostname": "Nexus9k",
"net_image": "bootflash:///nxos64-cs.10.3.1.F.bin",
"net_interfaces": {
  "Ethernet1/1": {
    "bandwidth": "10000000",
    "duplex": "auto",
    "macaddress": "0097.e76b.0101",
    "mode": "access",
    "mtu": "1500",
    "speed": "auto-speed",
    "state": "down",
    "type": "100/1000/10000 Ethernet"
  },
  "Ethernet1/10": {
    "bandwidth": "10000000",
    "duplex": "auto",
    "macaddress": "0097.e76b.010a",
    "mode": "access",
    "mtu": "1500",
    "speed": "auto-speed",
    "state": "down",
    "type": "100/1000/10000 Ethernet"
  },
  "Ethernet1/11": {
    "bandwidth": "10000000",
```

```
    "Ethernet1/54",
    "Ethernet1/55",
    "Ethernet1/56",
    "Ethernet1/57",
    "Ethernet1/58",
    "Ethernet1/59",
    "Ethernet1/60",
    "Ethernet1/61",
    "Ethernet1/62",
    "Ethernet1/63",
    "Ethernet1/64"
  ],
"net_license_hostid": "9RFKQDDT8YI",
"net_memfree_mb": 4950.57421875,
"net_memtotal_mb": 9952.61328125,
"net_model": "Nexus9000 C9500v Chassis (\\"Supervisor Module\\")",
"net_module": [
  {
    "model": "N9K-X9564v",
    "ports": "64",
    "status": "ok",
    "type": "Nexus 9000v 64 port Ethernet Module"
  },
  {
    "model": "N9K-vSUP",
    "ports": "0",
    "status": "active **",
    "type": "Virtual Supervisor Module"
  }
],
"net_neighbors": {},
"net_platform": "N9K-C9500v",
"net_python_version": "3.10.12",
"net_serialnum": "9G9PDMTIYY0",
"net_system": "nxos",
"net_version": "10.3(1)",
"net_vlan_list": [
  "1"
],
"network_resources": {}
}
```

```
PLAY RECAP *****
N9K : ok=2 changed=0 unreachable=0 failed=0 skipped=0 rescued=0 ignored=0
root@localhost:~/DevNet_Korea/01_NXOS_facts#
```

# Playbook 실행 결과 txt 파일로 저장

```
root@localhost:~/DevNet_Korea/01_NXOS_facts# sudo ansible-playbook -i hosts  
1_gatheralllegacyfacts.yml | tee 1_gatheralllegacyfacts.txt
```

tee: 명령어의 출력 결과를 파일과 화면에 동시에 출력할 수 있도록 해주는 명령어

# nxos\_facts 실행 결과에서 특정 정보만 출력

```
- hosts: nexus
gather_facts: false

tasks:
  - name: Gather only hostname
    cisco.nxos.nxos_facts:
      gather_subset: legacy

  - name: Print hostname
    debug:
      var: ansible_facts.net_hostname
```

```
root@localhost:~/DevNet_Korea/01_NXOS_facts# ansible-playbook -i hosts 2_hostnameonly.yml
[WARNING]: Collection cisco.nxos does not support Ansible version 2.10.8

PLAY [nexus] *****

TASK [Gather only hostname] *****
[WARNING]: Collection ansible.netcommon does not support Ansible version 2.10.8
[WARNING]: Collection ansible.utils does not support Ansible version 2.10.8
ok: [N9K]

TASK [Print hostname] *****
[WARNING]: Collection ansible.netcommon does not support Ansible version 2.10.8
[WARNING]: Collection ansible.utils does not support Ansible version 2.10.8
ok: [N9K] => {
  "ansible_facts.net_hostname": "Nexus9k"
}

PLAY RECAP *****
N9K          : ok=2    changed=0    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
```



# nxos\_facts 실행 결과 예상해보기!

```
- name:  
  cisco.nxos.nxos_facts:  
    gather_subset:  
      - '!hardware'
```

```
- name:  
  cisco.nxos.nxos_facts:  
    gather_subset: hardware  
    gather_network_resources: hostname
```

# 아래 task를 실행하고 결과를 프린트하는 3\_interfaces.yml 파일 수정 및 실행해보기

```
- name: Gather only the interfaces resource facts and no
legacy facts
cisco.nxos.nxos_facts:
gather_subset:
- '!all'
- '!min'
gather_network_resources:
- interfaces
```

# Ansible nxos\_config 모듈

```
- name: Configure PIM Anycast RP
  cisco.nxos.nxos_config:
    lines:
      - "ip pim anycast-rp {{ s1_loopback1 }} {{ s1_loopback0 }}"
      - "ip pim anycast-rp {{ s2_loopback1 }} {{ s2_loopback0 }}"
    save_when: modified
```

## Options:

- always – copy always
- modified – copy only if changed since last save
- changed – copy only if the **task** made a change
- never – never copy

save\_when  
string

When changes are made to the device running-configuration, the changes are not copied to non-volatile storage by default. Using this argument will change that before. If the argument is set to *always*, then the running-config will always be copied to the startup-config and the *modified* flag will always be set to True. If the argument is set to *modified*, then the running-config will only be copied to the startup-config if it has changed since the last save to startup-config. If the argument is set to *never*, the running-config will never be copied to the startup-config. If the argument is set to *changed*, then the running-config will only be copied to the startup-config if the task has made a change. *changed* was added in Ansible 2.6.

### Choices:

- "always"
- "never" ← (default)
- "modified"
- "changed"

# Ansible nxos\_command 모듈

Parameter	Comments
<b>commands</b> list / elements=any / required	<p>The commands to send to the remote NXOS device. The resulting output from the command is returned. If the <i>wait_for</i> argument is provided, the module is not returned until the condition is satisfied or the number of retries as expired.</p> <p>The <i>commands</i> argument also accepts an alternative form that allows for complex values that specify the command to run and the output format to return. This can be done on a command by command basis. The complex argument supports the keywords <i>command</i> and <i>output</i> where <i>command</i> is the command to run and <i>output</i> is one of 'text' or 'json'.</p> <p>If a command sent to the device requires answering a prompt, it is possible to pass a dict containing <i>command</i>, <i>answer</i> and <i>prompt</i>. Common answers are 'y' or "Yr" (carriage return, must be double quotes). See examples.</p>
<b>interval</b> integer	<p>Configures the interval in seconds to wait between retries of the command. If the command does not pass the specified conditional, the interval indicates how to long to wait before trying the command again.</p> <p><b>Default:</b> 1</p>
<b>match</b> string	<p>The <i>match</i> argument is used in conjunction with the <i>wait_for</i> argument to specify the match policy. Valid values are all or any. If the value is set to all then all conditionals in the <i>wait_for</i> must be satisfied. If the value is set to any then only one of the values must be satisfied.</p> <p><b>Choices:</b></p> <ul style="list-style-type: none"><li>• "any"</li><li>• "all" ← (default)</li></ul>
<b>retries</b> integer	<p>Specifies the number of retries a command should be tried before it is considered failed. The command is run on the target device every retry and evaluated against the <i>wait_for</i> conditionals.</p> <p>The commands are run once when <i>retries</i> is set to 0.</p> <p><b>Default:</b> 9</p>
<b>wait_for</b> aliases: waitfor list / elements=string	<p>Specifies what to evaluate from the output of the command and what conditionals to apply. This argument will cause the task to wait for a particular conditional to be true before moving forward. If the conditional is not true by the configured retries, the task fails. See examples.</p>

# Ansible nxos\_command에서 prompts 사용

```
- name: run commands that require answering a prompt
cisco.nxos.nxos_command:
  commands:
    - configure terminal
    - command: no feature npv
  prompt: Do you want to continue
  answer: y
```

```
- name: Misc Commands
cisco.nxos.nxos_command:
  commands: copy ftp://nxos.bin bootflash:
  prompt:
    - "Username:"
    - "Password:"
  answer:
    - devnet
    - 1234Qwer
```

# Ansible nxos\_command를 사용한 playbook 실행

```
---
- hosts: nexus
  gather_facts: false
  tasks:
    - name: Run show version on Remote Device
      cisco.nxos.nxos_command:
        commands: show version
      register: results
    - name: Show results
      debug:
        msg: "{{ results.stdout_lines }}"
```

```
ok: [N9K] => {
  "msg": [
    "Cisco Nexus Operating System (NX-OS) Software",
    "TAC support: http://www.cisco.com/tac",
    "Documents: http://www.cisco.com/en/US/products/ps9372/tsd_products_support_series_home.html",
    "Copyright (c) 2002-2022, Cisco Systems, Inc. All rights reserved.",
    "The copyrights to certain works contained herein are owned by",
    "other third parties and are used and distributed under license.",
    "Some parts of this software are covered under the GNU Public",
    "License. A copy of the license is available at",
    "http://www.gnu.org/licenses/gpl.html.",
    "",
    "Nexus 9000v is a demo version of the Nexus Operating System",
    "",
    "Software",
    "  BIOS: version ",
    "  NXOS: version 10.3(1) [Feature Release]",
    "  BIOS compile time: ",
    "  NXOS image file is: bootflash:///nxos64-cs.10.3.1.F.bin",
    "  NXOS compile time:  8/18/2022 15:00:00 [08/19/2022 02:44:02]",
    "",
    "Hardware",
    "  cisco Nexus9000 C9500v Chassis (\\"Supervisor Module\\")",
    "  Intel(R) Xeon(R) CPU E7- 2830  @ 2.13GHz with 10191476 kB of memory.",
    "  Processor Board ID 9G9PDMTIYY0",
    "  Device name: Nexus9k",
    "  bootflash:  4287040 kB",
    "",
    "Kernel uptime is 0 day(s), 18 hour(s), 58 minute(s), 7 second(s)",
    "",
    "Last reset ",
    "  Reason: Unknown",
    "  System version: ",
    "  Service: ",
    "",
    "plugin",
    "  Core Plugin, Ethernet Plugin",
    "",
    "Active Package(s):"
  ]
}
```

```
PLAY RECAP *****
N9K : ok=2    changed=0    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
```

# Ansible nxos\_command를 사용한 playbook 실행

```
ok: [N9K] => {
  "msg": [
    "Cisco Nexus Operating System (NX-OS) Software",
    "TAC support: http://www.cisco.com/tac",
    "Documents: http://www.cisco.com/en/US/products/ps9372/tsd_products_support_series_home.html",
    "Copyright (c) 2002-2022, Cisco Systems, Inc. All rights reserved.",
    "The copyrights to certain works contained herein are owned by",
    "other third parties and are used and distributed under license.",
    "Some parts of this software are covered under the GNU Public",
    "License. A copy of the license is available at",
    "http://www.gnu.org/licenses/gpl.html.",
    "",
    "Nexus 9000v is a demo version of the Nexus Operating System",
    "",
    "Software",
    "  BIOS: version ",
    "  NXOS: version 10.3(1) [Feature Release]",
    "  BIOS compile time: ",
    "  NXOS image file is: bootflash://nxos64-cs.10.3.1.F.bin",
    "  NXOS compile time:  8/18/2022 15:00:00 [08/19/2022 02:44:02]",
    "",
    "Hardware",
    "  cisco Nexus9000 C9500v Chassis (\\\"Supervisor Module\\\")",
    "  Intel(R) Xeon(R) CPU E7- 2830  @ 2.13GHz with 10191476 kB of memory.",
    "  Processor Board ID 9G9PDMTIYYO",
    "  Device name: Nexus9k",
    "  bootflash:    4287040 kB",
    "",
    "Kernel uptime is 0 day(s), 18 hour(s), 58 minute(s), 7 second(s)",
    "",
    "Last reset ",
    "  Reason: Unknown",
    "  System version: ",
    "  Service: ",
    "",
    "plugin",
    "  Core Plugin, Ethernet Plugin",
    "",
    "Active Package(s):"
  ]
}
```

PLAY RECAP \*\*\*\*\*

Host	ok	changed	unreachable	failed	skipped	rescued	ignored
N9K	2	0	0	0	0	0	0

```
Nexus9k# show ver
Cisco Nexus Operating System (NX-OS) Software
TAC support: http://www.cisco.com/tac
Documents: http://www.cisco.com/en/US/products/ps9372/tsd_products_support_series_home.html
Copyright (c) 2002-2022, Cisco Systems, Inc. All rights reserved.
The copyrights to certain works contained herein are owned by
other third parties and are used and distributed under license.
Some parts of this software are covered under the GNU Public
License. A copy of the license is available at
http://www.gnu.org/licenses/gpl.html.

Nexus 9000v is a demo version of the Nexus Operating System

Software
  BIOS: version
  NXOS: version 10.3(1) [Feature Release]
  BIOS compile time:
  NXOS image file is: bootflash://nxos64-cs.10.3.1.F.bin
  NXOS compile time:  8/18/2022 15:00:00 [08/19/2022 02:44:02]

Hardware
  cisco Nexus9000 C9500v Chassis ("Supervisor Module")
  Intel(R) Xeon(R) CPU E7- 2830  @ 2.13GHz with 10191476 kB of memory.
  Processor Board ID 9G9PDMTIYYO
  Device name: Nexus9k
  bootflash:    4287040 kB

Kernel uptime is 0 day(s), 18 hour(s), 58 minute(s), 52 second(s)

Last reset
  Reason: Unknown
  System version:
  Service:

plugin
  Core Plugin, Ethernet Plugin

Active Package(s):

Nexus9k#
```

# Ansible nxos\_command를 사용한 playbook 실행

```
---
- hosts: nexus
  gather_facts: false
  tasks:
    - name: Run show version on Remote Device
      cisco.nxos.nxos_command:
        commands: show version
        wait_for: result[0] contains hyeyeon
        register: results
    - name: Show results
      debug:
        msg: "{{ results.stdout_lines }}"
```

```
root@localhost:~/DevNet_Korea/01_NXOS_facts# ansible-playbook -i hosts 4_commandshowver.yml
[WARNING]: Collection cisco.nxos does not support Ansible version 2.10.8

PLAY [nexus] *************************************************************************************************************************************

TASK [Run show version on Remote Device] *************************************************************
[WARNING]: Collection ansible.netcommon does not support Ansible version 2.10.8
[WARNING]: Collection ansible.utils does not support Ansible version 2.10.8
fatal: [N9K]: FAILED! => {"changed": false, "failed_conditions": ["result[0] contains hyeyeon"], "msg": "One or more conditional statements have not been satisfied"}

PLAY RECAP *********************************************************************
N9K                : ok=0    changed=0    unreachable=0    failed=1    skipped=0    rescued=0    ignored=0
```



# Ansible nxos\_command를 사용한 playbook 실행

```
---
- hosts: nexus
  gather_facts: false
  tasks:
    - name: Run show version on Remote
      Device
      cisco.nxos.nxos_command:
        commands:
          - show version
          - show interface brief
        register: results
    - name: Show results
      debug:
        msg: "{{ results.stdout_lines }}"
```

```
root@localhost:~/DevNet_Korea/01_NXOS_facts# ansible-playbook -i hosts_4_commandshowver.yml
[WARNING]: Collection cisco.nxos does not support Ansible version 2.10.8

PLAY [nexus] *********************************************************************

TASK [Run show version on Remote Device] *****
[WARNING]: Collection ansible.netcommon does not support Ansible version 2.10.8
[WARNING]: Collection ansible.utils does not support Ansible version 2.10.8
ok: [N9K]

TASK [Show results] *****
[WARNING]: Collection ansible.netcommon does not support Ansible version 2.10.8
[WARNING]: Collection ansible.utils does not support Ansible version 2.10.8
ok: [N9K] => {
  "msg": [
    [
      "Cisco Nexus Operating System (NX-OS) Software",
      "TAC support: http://www.cisco.com/tac",
      "Documents: http://www.cisco.com/en/US/products/ps9372/tsd_products_support_series_home.html",
      "Copyright (c) 2002-2022, Cisco Systems, Inc. All rights reserved.",
      "The copyrights to certain works contained herein are owned by",
      "other third parties and are used and distributed under license.",
      "Some parts of this software are covered under the GNU Public",
      "License. A copy of the license is available at",
      "http://www.gnu.org/licenses/gpl.html.",
      "",
      "Nexus 9000v is a demo version of the Nexus Operating System",
      "",
      "Software",
      "  BIOS: version ",
      "  NXOS: version 10.3(1) [Feature Release]",
      "  BIOS compile time: ",
      "  NXOS image file is: bootflash://nxos64-cs.10.3.1.F.bin",
      "  NXOS compile time:  8/18/2022 15:00:00 [08/19/2022 02:44:02]",
      "",
      "Hardware",
      "  cisco Nexus9000 C9500v Chassis ("Supervisor Module")",
      "  Intel(R) Xeon(R) CPU E7- 2830  @ 2.13GHz with 10191476 kB of memory.",
      "  Processor Board ID 9G9PDM1YV0",
      "  Device name: Nexus9K",
      "  bootflash:  4287040 kB",
      "",
      "Kernel uptime is 0 day(s), 19 hour(s), 18 minute(s), 34 second(s)",
      "",
      "Last reset ",
    ]
  ]
}
```

Port	VRF	Status	IP Address	Speed	MTU		
mgmt0	--	up	198.18.135.100	1000	1500		
Ethernet Interface	VLAN	Type	Mode	Status	Reason	Speed	Port Ch #
Eth1/1	1	eth	access	down	Link not connected	auto(D)	--
Eth1/2	1	eth	access	down	Link not connected	auto(D)	--
Eth1/3	1	eth	access	down	Link not connected	auto(D)	--
Eth1/4	1	eth	access	down	Link not connected	auto(D)	--
Eth1/5	1	eth	access	down	Link not connected	auto(D)	--
Eth1/6	1	eth	access	down	Link not connected	auto(D)	--
Eth1/7	1	eth	access	down	Link not connected	auto(D)	--
Eth1/8	1	eth	access	down	Link not connected	auto(D)	--
Eth1/9	1	eth	access	down	Link not connected	auto(D)	--
Eth1/10	1	eth	access	down	Link not connected	auto(D)	--
Eth1/11	1	eth	access	down	Link not connected	auto(D)	--
Eth1/12	1	eth	access	down	Link not connected	auto(D)	--
Eth1/13	1	eth	access	down	Link not connected	auto(D)	--
Eth1/14	1	eth	access	down	Link not connected	auto(D)	--
Eth1/15	1	eth	access	down	Link not connected	auto(D)	--
Eth1/16	1	eth	access	down	Link not connected	auto(D)	--
Eth1/17	1	eth	access	down	Link not connected	auto(D)	--
Eth1/18	1	eth	access	down	Link not connected	auto(D)	--
Eth1/19	1	eth	access	down	Link not connected	auto(D)	--
Eth1/20	1	eth	access	down	Link not connected	auto(D)	--
Eth1/21	1	eth	access	down	Link not connected	auto(D)	--
Eth1/22	1	eth	access	down	Link not connected	auto(D)	--
Eth1/23	1	eth	access	down	Link not connected	auto(D)	--
Eth1/24	1	eth	access	down	Link not connected	auto(D)	--
Eth1/25	1	eth	access	down	Link not connected	auto(D)	--
Eth1/26	1	eth	access	down	Link not connected	auto(D)	--
Eth1/27	1	eth	access	down	Link not connected	auto(D)	--
Eth1/28	1	eth	access	down	Link not connected	auto(D)	--
Eth1/29	1	eth	access	down	Link not connected	auto(D)	--
Eth1/30	1	eth	access	down	Link not connected	auto(D)	--
Eth1/31	1	eth	access	down	Link not connected	auto(D)	--
Eth1/32	1	eth	access	down	Link not connected	auto(D)	--
Eth1/33	1	eth	access	down	Link not connected	auto(D)	--
Eth1/34	1	eth	access	down	Link not connected	auto(D)	--
Eth1/35	1	eth	access	down	Link not connected	auto(D)	--
Eth1/36	1	eth	access	down	Link not connected	auto(D)	--
Eth1/37	1	eth	access	down	Link not connected	auto(D)	--

# 미션!



# 미션



## 목표

- nexus 스위치의 os 버전, License hostid, redundancy status를 print하도록 5\_mission\_nxos.yml playbook을 수정 후 성공적으로 실행
- 보너스 미션! (선착순 1명 스타벅스 카드)  
→ 실행 결과를 mission\_nxos\_result.txt 파일로 생성하는 playbook 작성 및 실행

## 힌트

- nxos\_facts 모듈 및 nxos\_command 모듈 사용
- OS 버전, license hostid는 nxos\_facts 모듈로 획득 가능
- Gather\_subset -> min 활용
- Redundancy status 확인하는 명령어 : show redundancy status



Thank you

