

CISCO *Live!*

GO BEYOND

#CiscoLiveAPJC



Ansible Zero to Hero – Network Automation

Vishal Kakkar – Solution Architect

Sanjeewa Alahakone – Solution Architect

LTRPRG-1125

CISCO *Live!*

#CiscoLiveAPJC

Cisco Webex App

Questions?

Use Cisco Webex App to chat with the speaker after the session

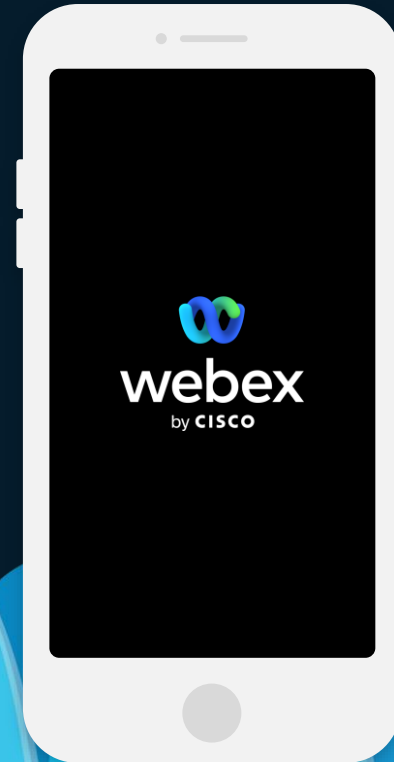
How

- 1 Find this session in the Cisco Live Mobile App
- 2 Click “Join the Discussion”
- 3 Install the Webex App or go directly to the Webex space
- 4 Enter messages/questions in the Webex space

Webex spaces will be moderated by the speaker until November 15, 2024.



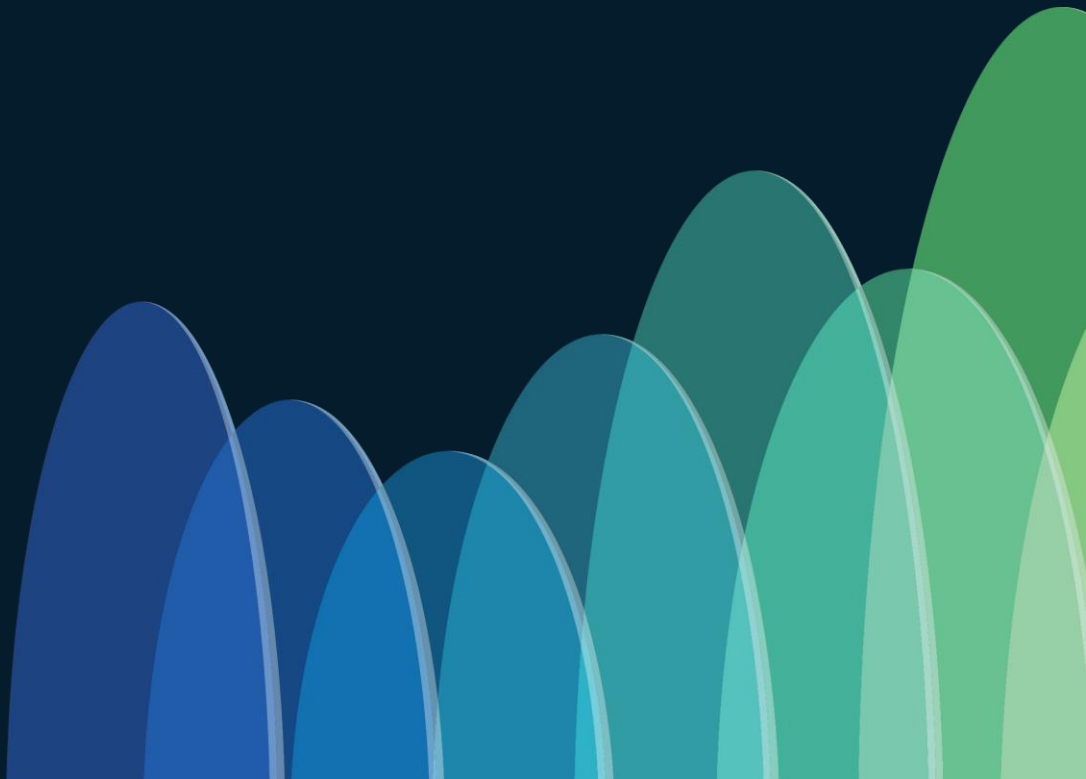
<https://ciscolive.ciscoevents.com/ciscolivebot/LTRPRG-1125>



Agenda

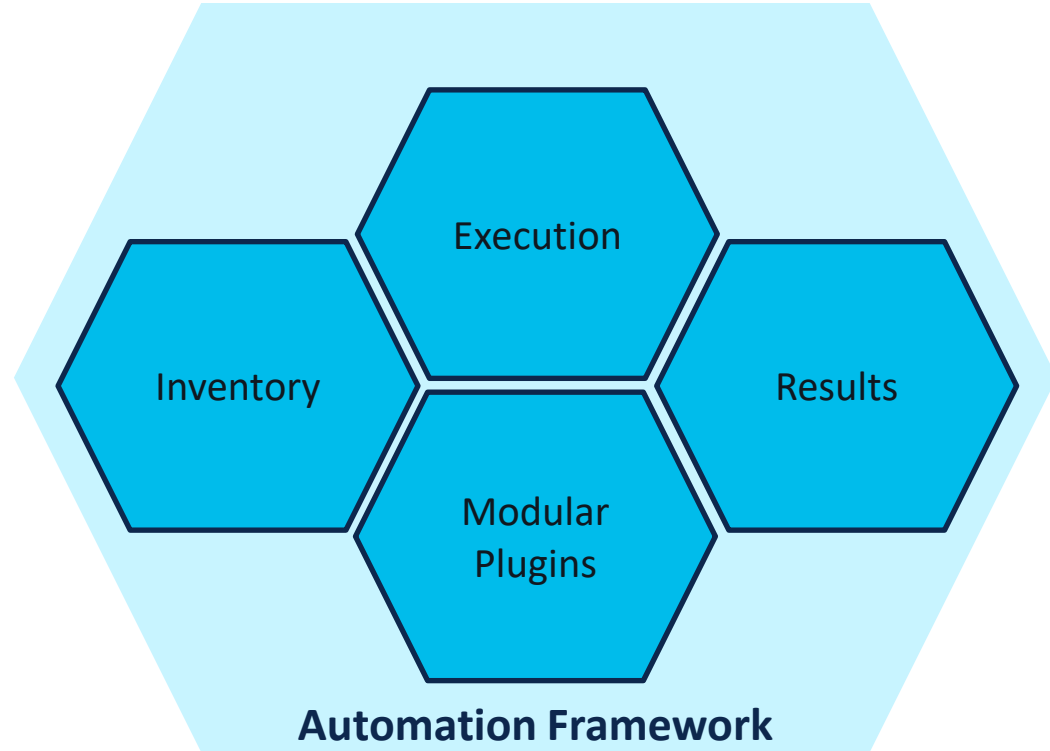
- What is Ansible
- Why Ansible
- Lab Introduction
- Lab Hands-on
 - Lab-1: Familiarize with Ansible Env
 - Lab-2: Basic Ansible Commands
 - Lab-3: Deep Dive
 - Lab-4: Advanced Topics
- Ansible vs other Tools Comparison
- Ansible vs NSO
- Conclusion

What is Ansible?

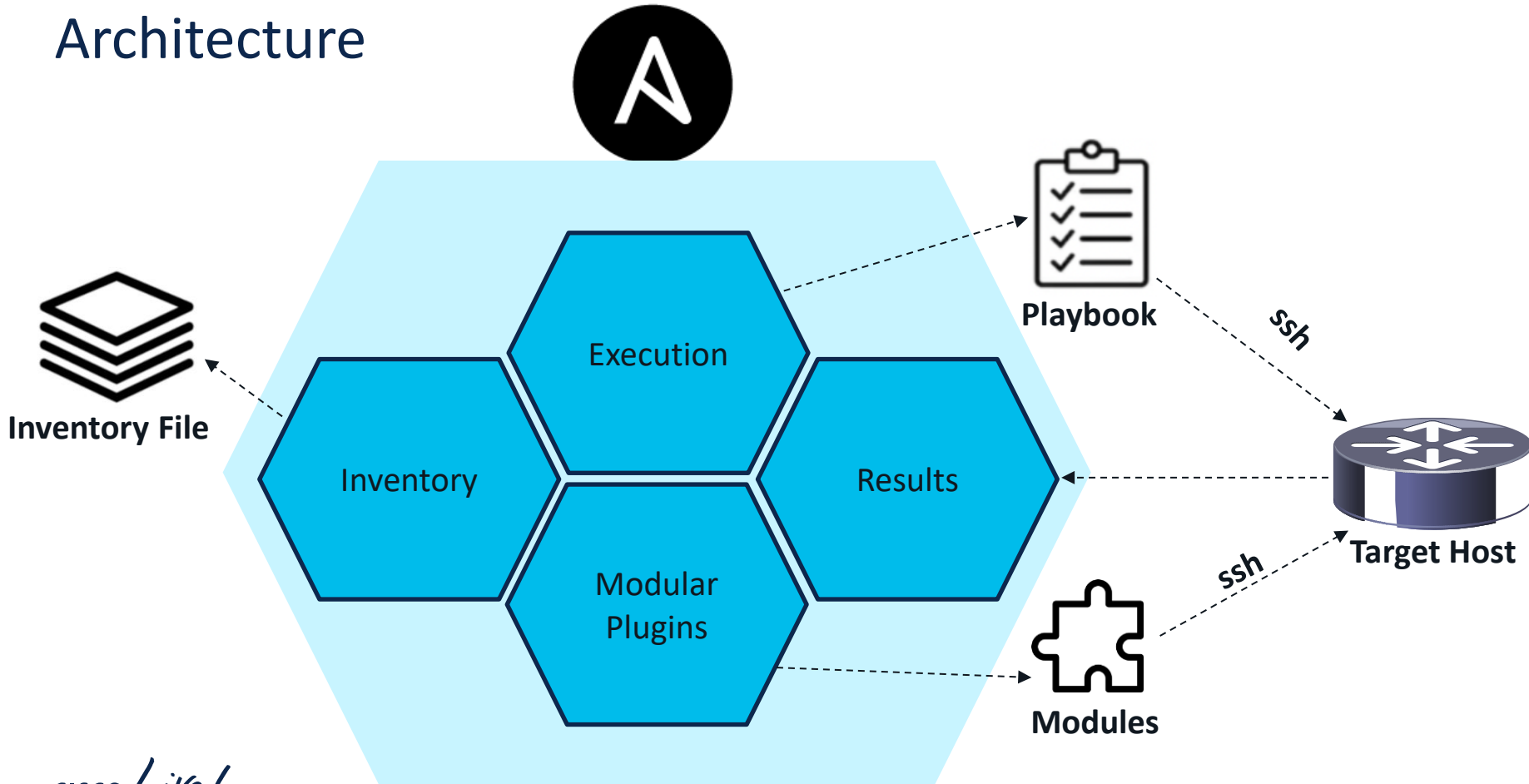


What is Ansible

- Open-source
- Automation Framework
- Founded in 2013, bought by Red Hat in 2015
- Controls all the target nodes/hosts from single machine ([ansible controller](#))
- Uses standard [ssh](#) for communication



Architecture



Ansible Packaging

- Ansible consists of 2 packages:
 - ansible-core
 - ansible
- ‘ansible-core’
 - runtime
 - fundamental modules & plugins
- ‘ansible’
 - Community-developed modules

Ansible Installation

- On Fedora:

```
$ sudo dnf install ansible
```

- On RHEL and CentOS:

```
$ sudo yum install ansible
```

- Ubuntu:

```
$ sudo apt update
```

```
$ sudo apt install software-properties-common
```

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

```
$ sudo apt install ansible
```

- MacOS:

```
$ pip3 install ansible
```

- Windows is not supported as controller

Upgrade from v2.9 and earlier not possible!

```
pip3 uninstall ansible
```

```
pip3 install ansible
```

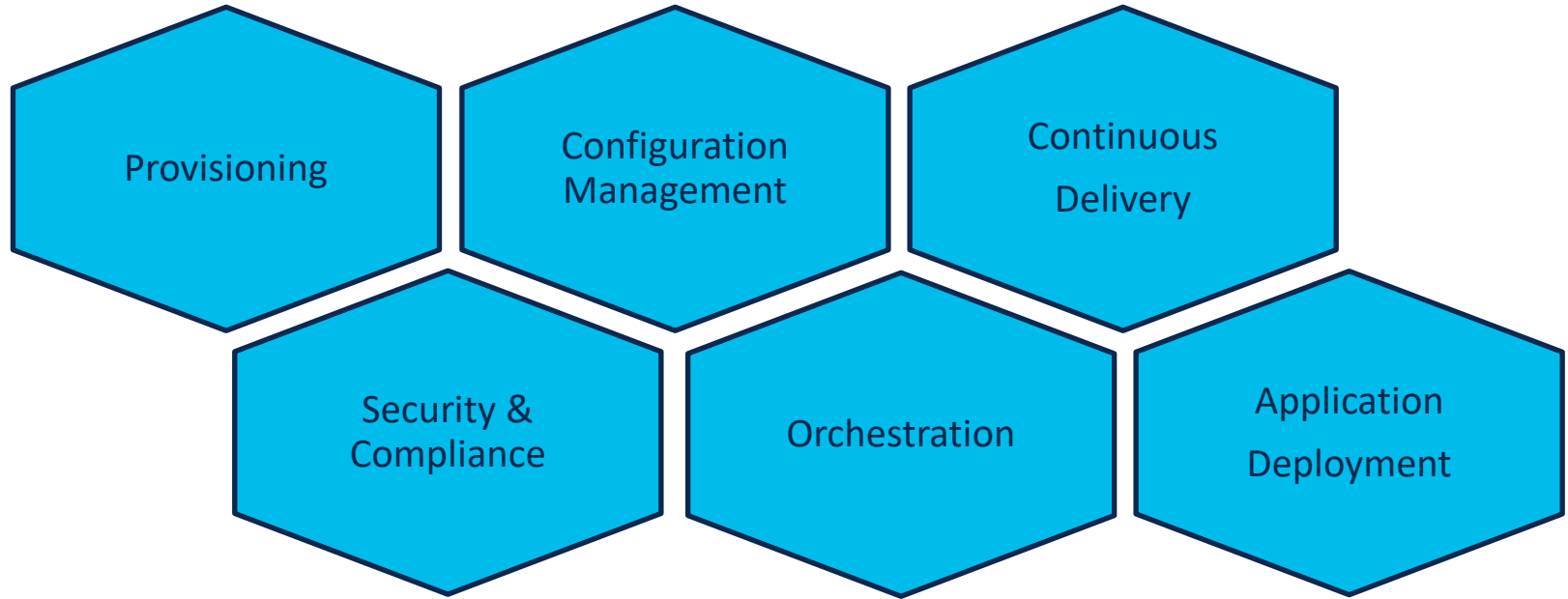
Why Ansible?



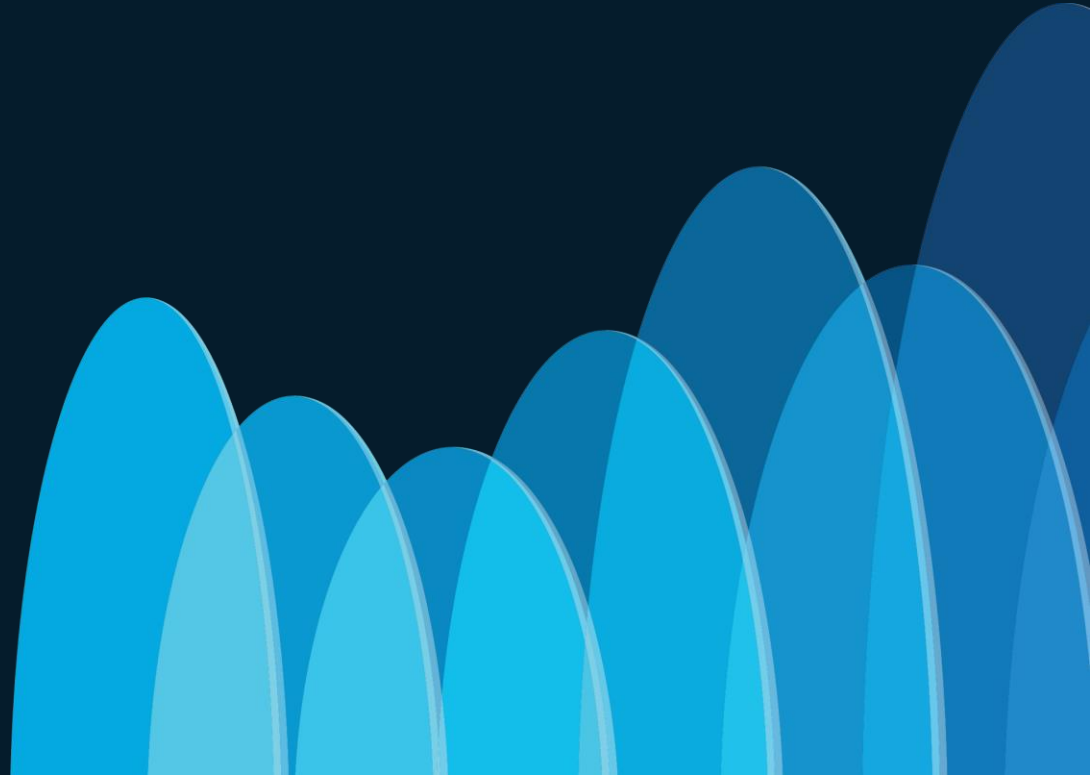
Why Ansible

- Simple
 - Uses simple syntax written in **YAML** (***Y**AML **A**in't **M**arkup **L**anguage*).
 - 'No Code Low' Code philosophy, YAML is just enough.
- Agentless
 - No Agents or software required to be installed on target hosts.
 - No special firewall ports needs to be opened as ansible uses ssh.
- Powerful
 - Features that enables to model even complex workflows.

Use Cases

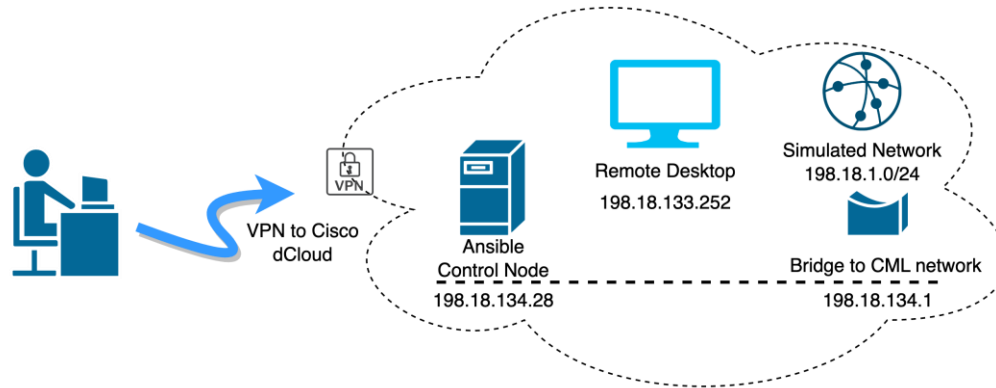


Lab Introduction



dCloud Lab Setup

- **Control node:** Ansible VM based on Ubuntu
- **Managed nodes:** 2 XRv9K core router, 2 IOS & CSR1kv branch routers each, 2 NX-OSv switches
- **Windows Jump host:** Windows with Visual Studio editor, Putty SSH client

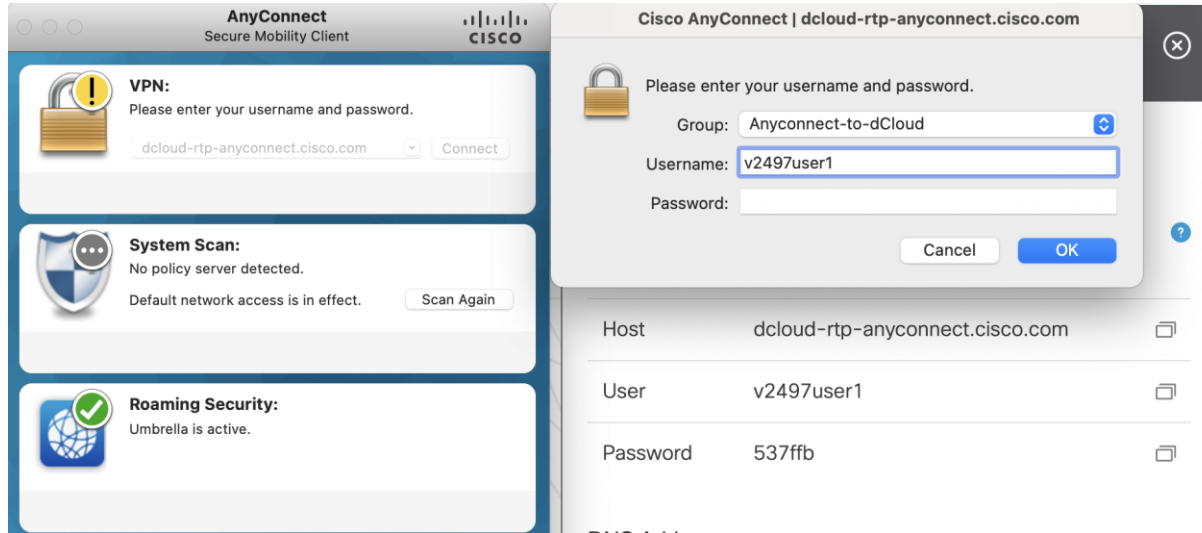


CISCO *Live!*



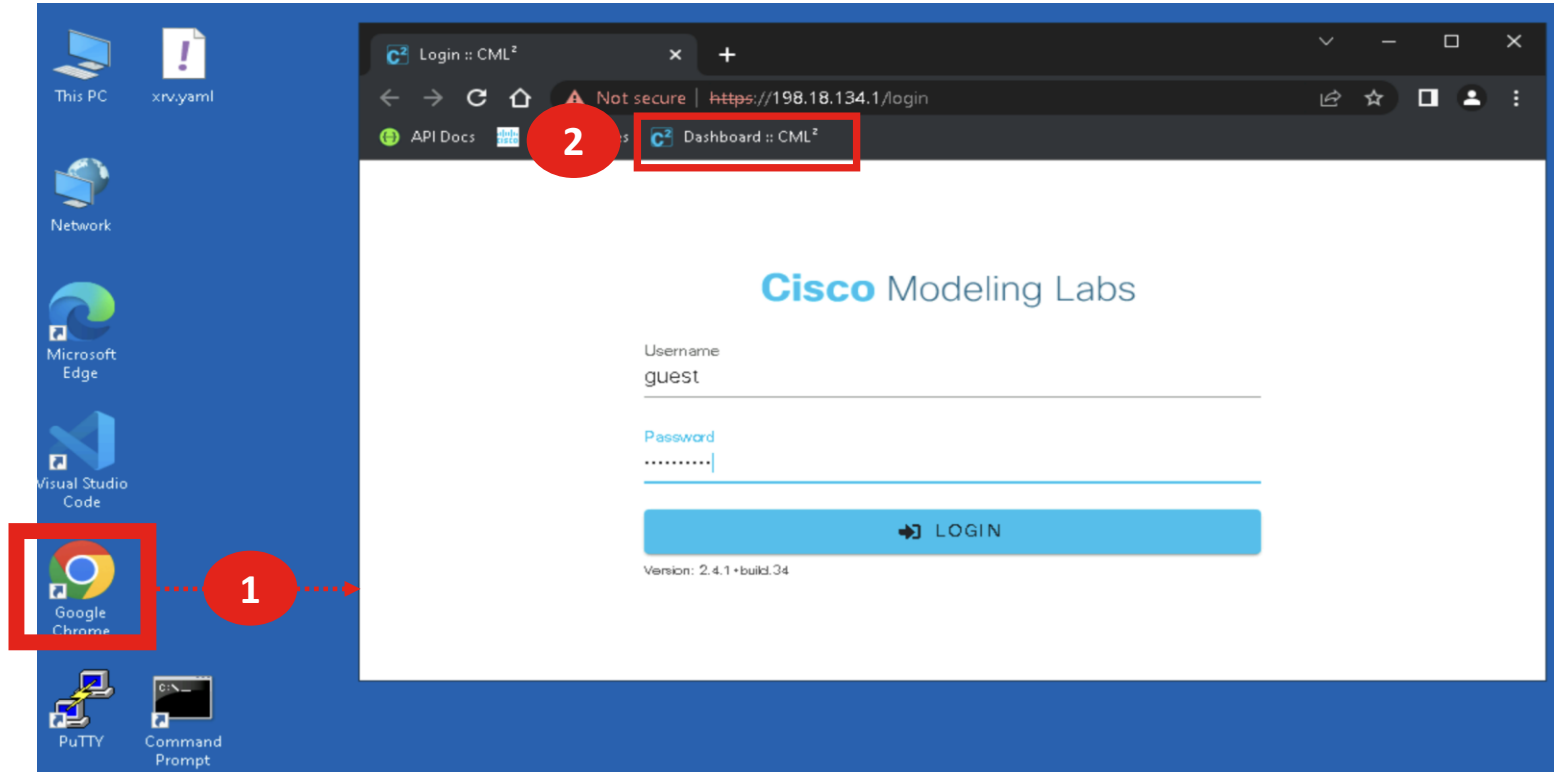
Lab Access

- Use the **Cisco AnyConnect Client** and your provided VPN username and password to connect to your lab instance

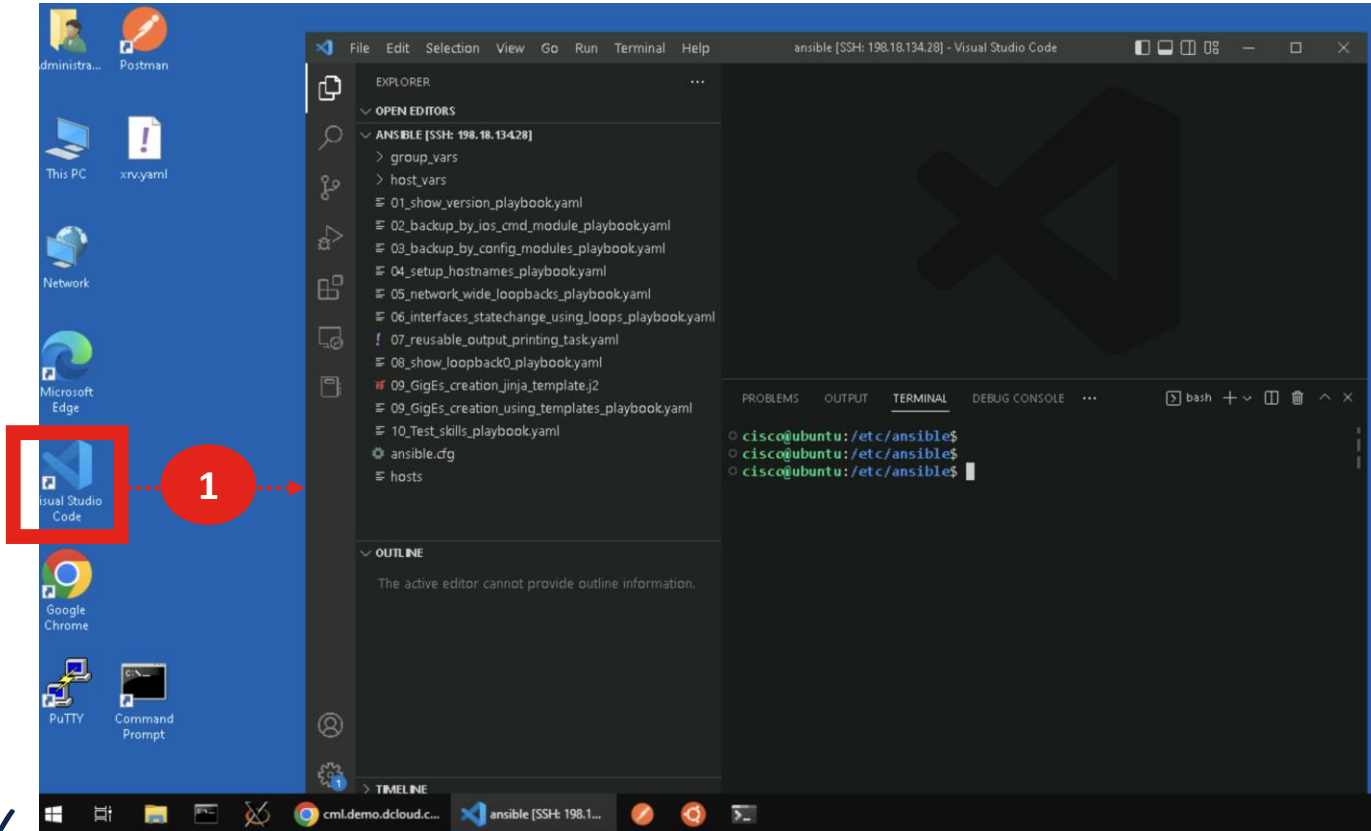


- Connect to the **Windows jump host** using **RDP client** to address **198.18.133.252**

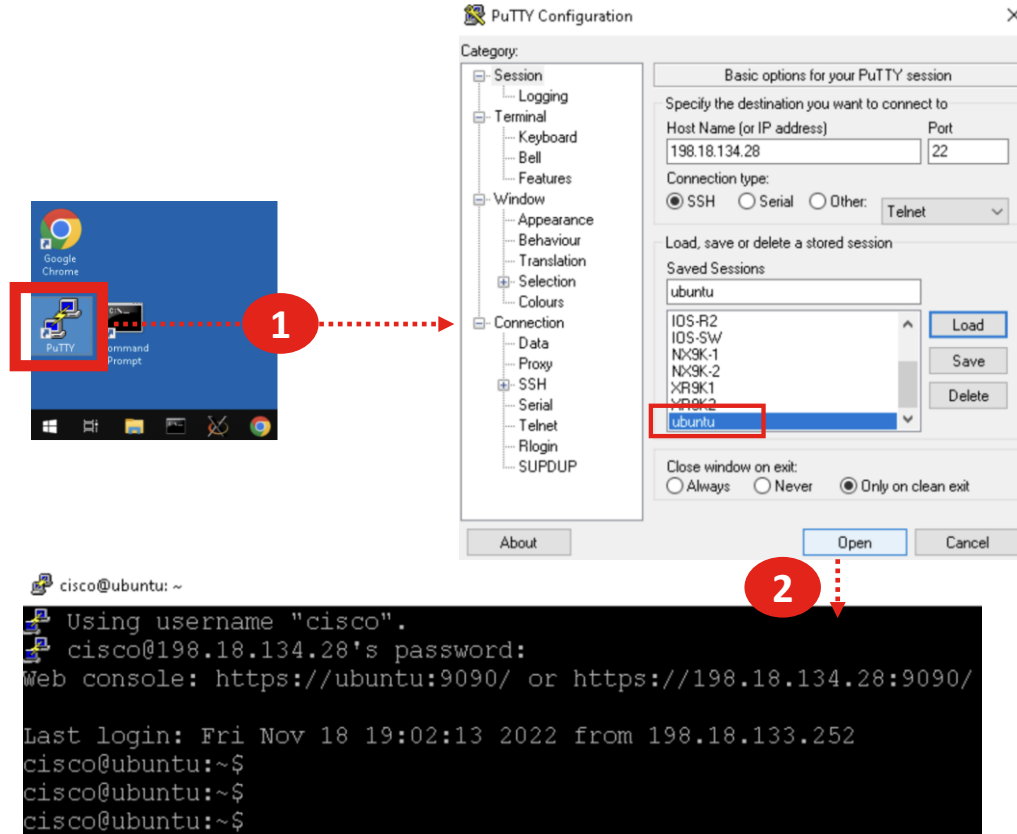
Windows Jump host – CML



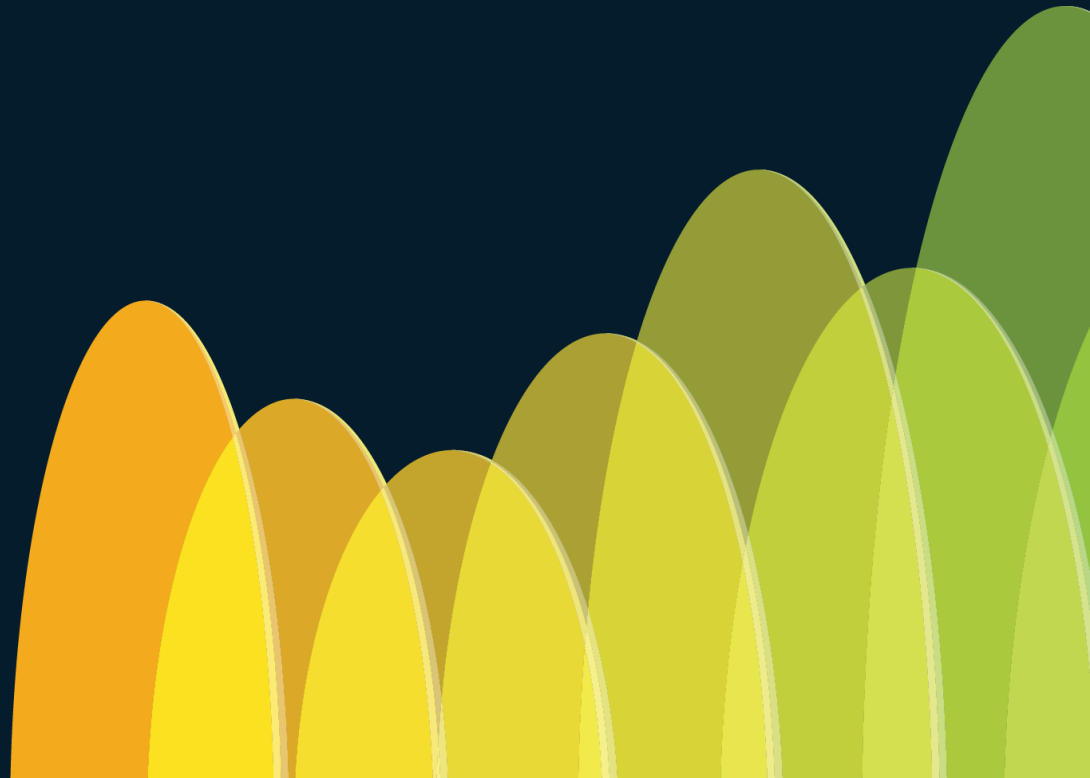
Windows Jump host – Visual Studio (Visio)



Windows Jump host – Ansible Controller (Ubuntu)



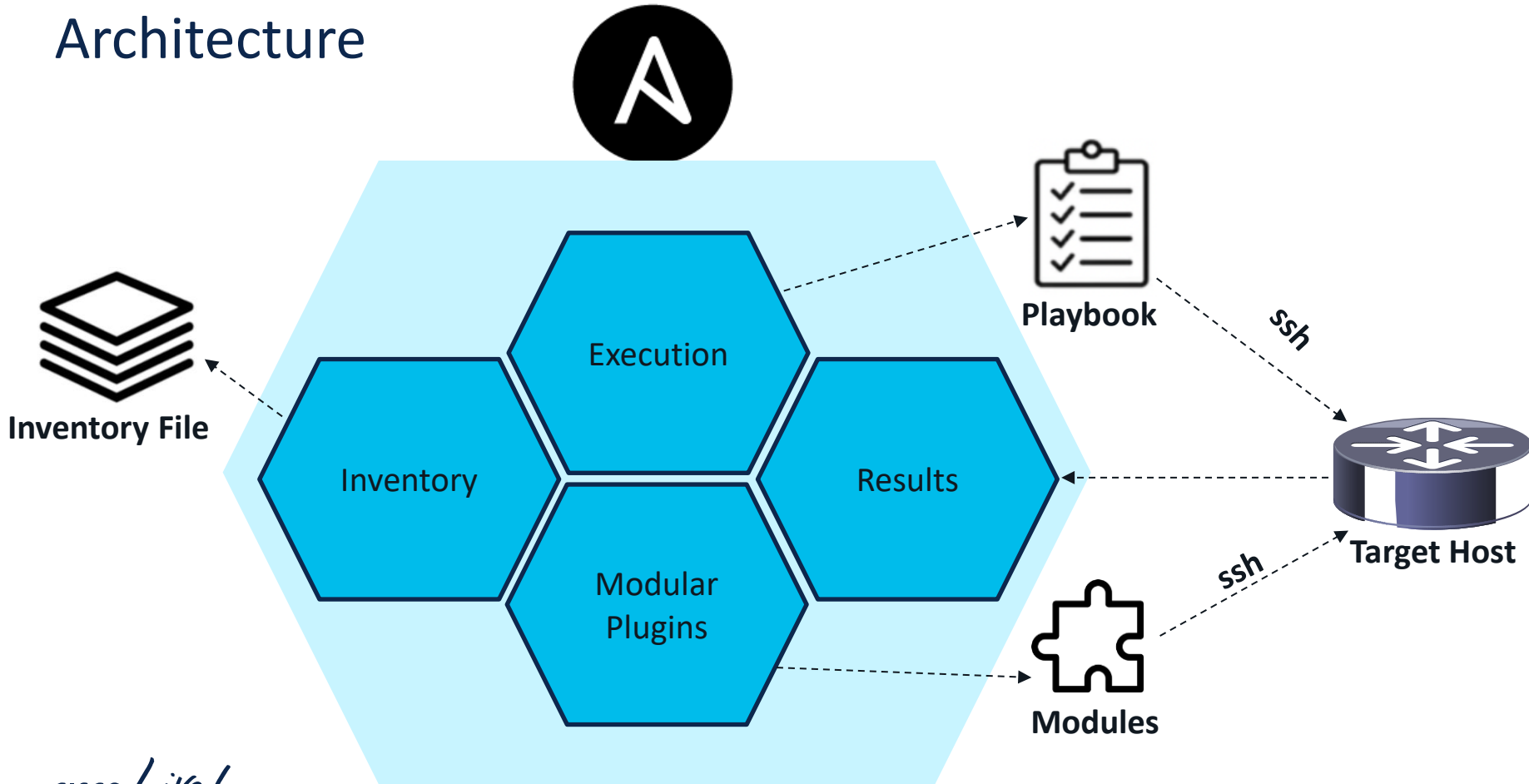
Hands-on Lab



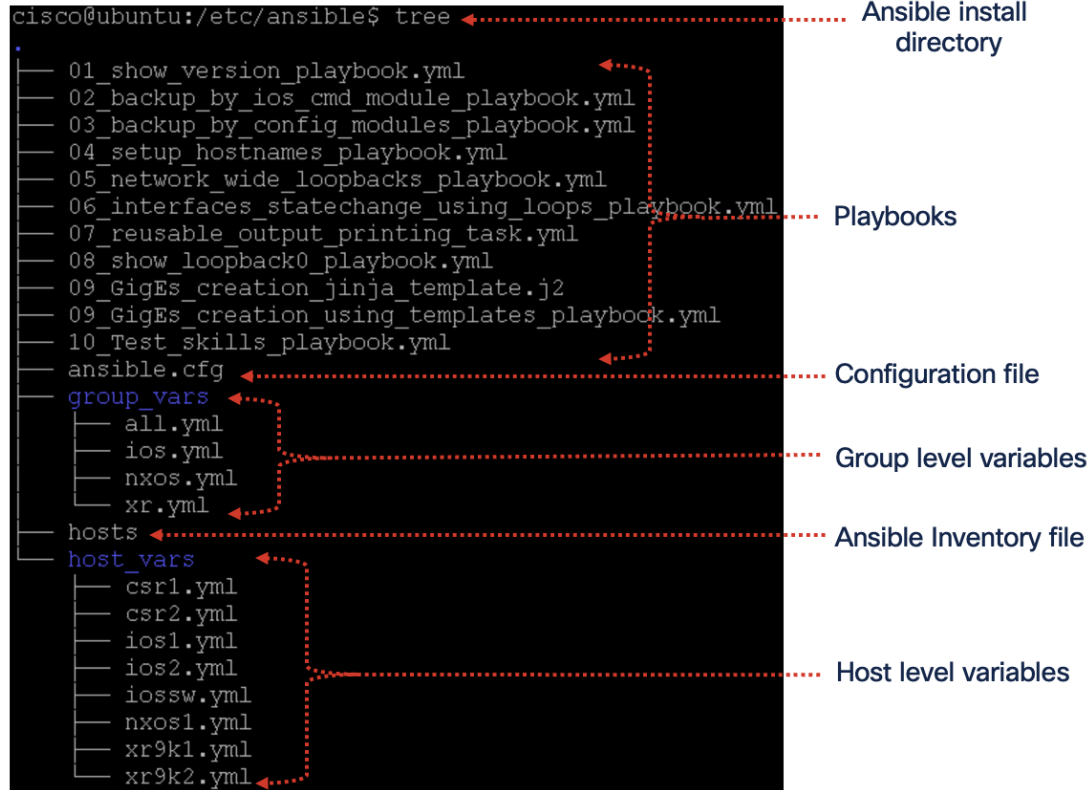
LAB 1: Familiarize with Ansible Environment



Architecture



Ansible Directory Structure (LAB)



Ansible Inventory

- Written in [INI](#) or [YAML](#).
- Default location [/etc/ansible/hosts](#)
 - Can be changed in [/etc/ansible/ansible.cfg](#)
- Contains [target hosts details](#) like hostname, IP, protocol, credentials
- Allows [Grouping](#) of target hosts for collative reference
 - Two default groups: all & ungrouped
- Can also stores [variables](#), specific per host or per group or for all.
- Separate file can also be used for variable, but this file must be located in subfolder [group_vars](#) or [host_vars](#) in same directory of inventory/host file

Ansible Playbook

- Written in **YAML**
- Contains one or multiple **plays** in a playbook
- Each play further contains one or multiple **tasks**
- Task is a **single action** to be performed by ansible



Ansible Modules

- Reusable or standalone script written in any programming language (usually python) that can return JSON in response
- Modules expose functions with acceptable inputs and perform desired execution on target host
- Modules abstract the underlying complexity and simplify user tasks


ios_command
ios_config
nxos_config


Junos_command
Junos_config


eos_command
eos_config

LAB 2:

Basic Ansible Commands

Ad-hoc Commands

- Allows you to execute tasks quickly without saving steps
- Useful to understand the basics of how Ansible works
- `ansible -m <module> [-a <arguments>] <hosts_section>`
 - Default module is “command” (“-m command” can be omitted)
 - “-m ping” is the ‘Hello World’ of Ansible

```
$ ansible -a "date" all
localhost | SUCCESS | rc=0 >>
Wed Nov 25 05:52:52 CET 2022
$ ansible -m ping csrl
198.18.134.28 | SUCCESS => {
  "changed": false,
  "failed": false,
  "ping": "pong"
}
```

Playbooks

- Ansible's method of procedures (MoP)

```

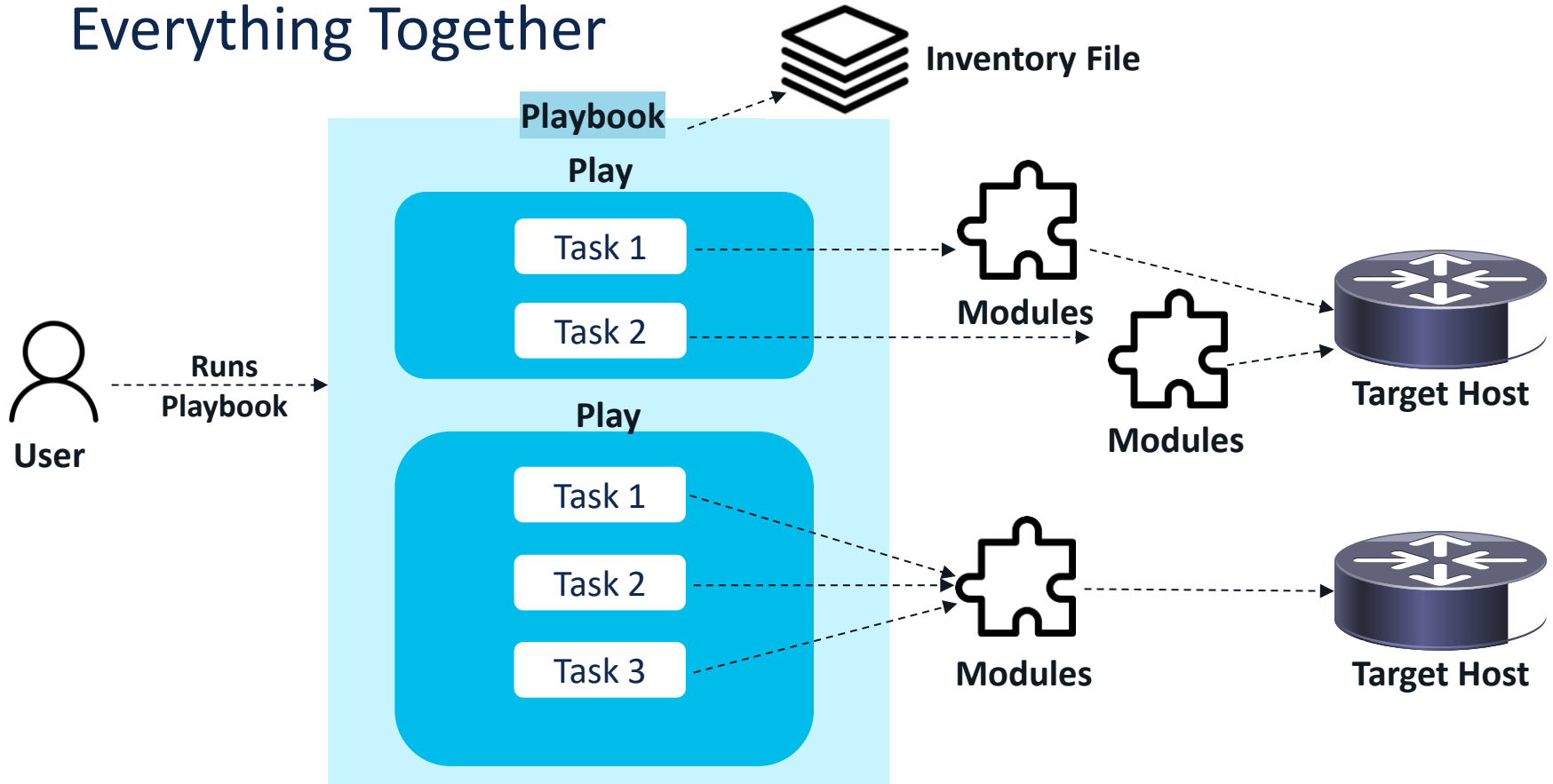
1  ---
2
3  # Play book to show version of all IOS & XRs
4  - name: Get Device versions
5    hosts: all, !nxos
6    tasks:
7      # First Task to get the versions from devices
8      - name: Running 'show version' command
9        # Using ios_command module
10       ios_command:
11         commands:
12           - show version
13       # Saving the output of above CLI to a variable named show_ver_output
14       register: show_ver_output
15
16     # Second Task to parse returned JSON and extract only relevant information for user
17     - name: Extracting only relevant information from 'show version' response
18       debug:
19         # Extracting Only the first line for show version CLI output
20         var: show_ver_output.stdout_lines[0][0]

```

- Starts with three "--"
- "#" used for comments
- Name of Play
- Target device/groups
- Task to run CLI on target
- Network Module being used
- Function of network module
- Parameters to the function
- Saving task output to variable
- Task to parse CLI output
- Module to print to screen
- Parameter to print

```
cisco@ubuntu:/etc/ansible$ ansible-playbook 01_show_version_playbook.yaml
```

Everything Together



Ansible Building Blocks

Inventory
+
Playbooks
+
Modules

1. Inventory of target hosts
2. Playbook to group the actions to be executed
3. Each action is a task
4. Each Task may or may not use modules.

LAB 3: Deep Dive

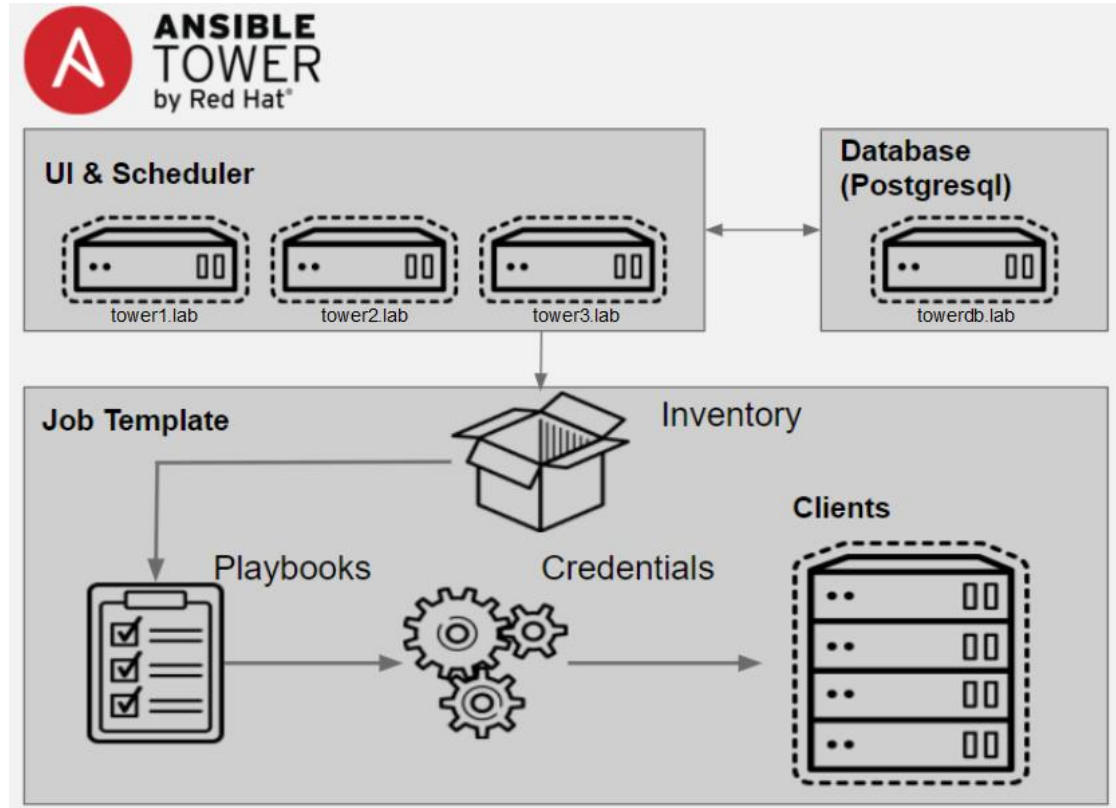


LAB 4: Advanced Topics



Ansible Tower/AWX

- **Web** Interface for ansible
 - Exposes **Rest APIs**
 - **User** control
 - **Audit** logs (**Most important)
-
- Ansible Tower is Paid & supported like RedHat Enterprise Linux
 - Ansible AWX is Community edition like Fedora upstream Linux



Ansible & Other Tools Comparison

Ansible & Other Tools Comparison

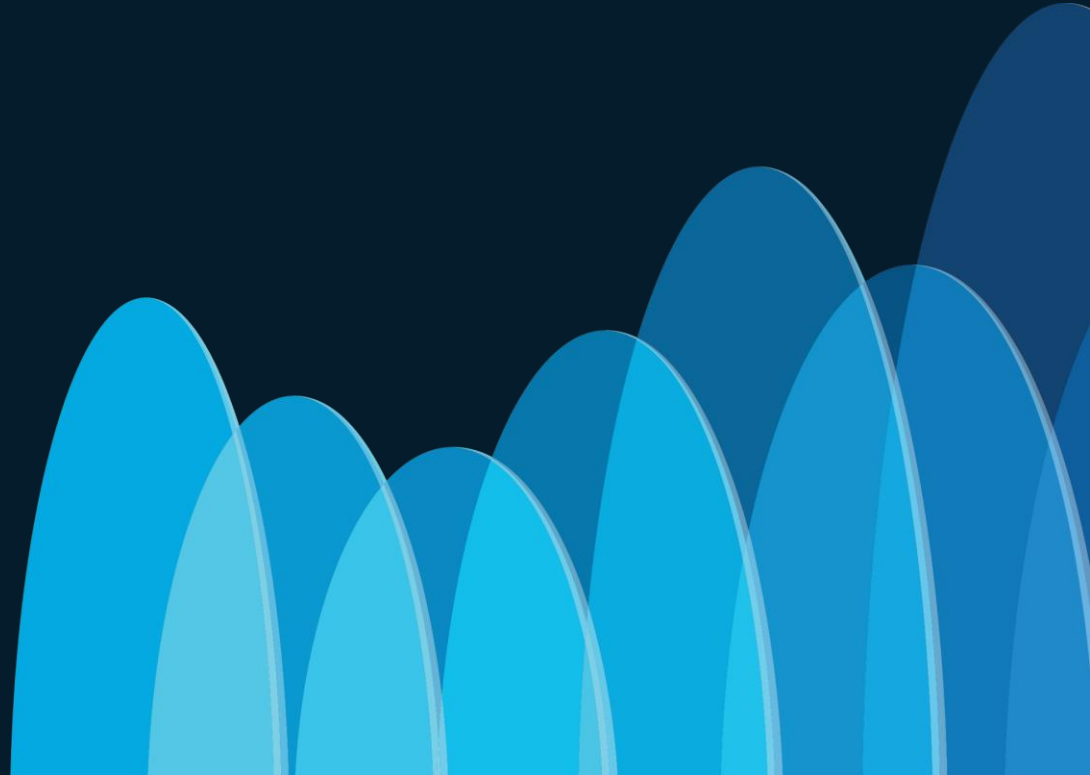
Tools	Architecture	Agentless	Communication
Puppet	Master/Agent	No	Pull Mode
Chef	Master/Agent	No	Pull Mode
Salt	Master/Minion	No	Push Mode
NorNir	Python Automation Framework, Code Centric	☒	SSH
NetPalm	REST APIs Broker for network automation	☒	REST/SSH
Ansible	YAML based	☒	SSH

Ansible vs NSO

- Similarities:
 - Both are [Agentless](#), works in [Push Mode](#)
- Not so similar:

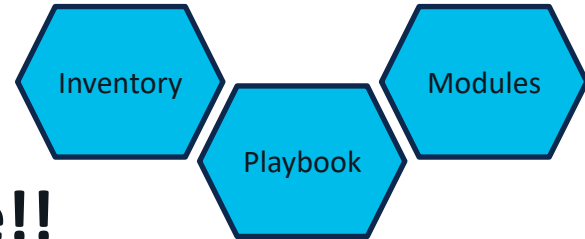
Criteria	Ansible	NSO
Protocols	Mainly SSH/ but increasing	NED based (CLI, Netconf, Rest, TAPI & ...)
Transactional	Run till Completion with Errors	All or Nothing
Rollback	Not supported	Built-in Rollback
Use cases Scope	Device lifecycle, DevOps	Service Lifecycle Management (YANG based)
North Bound Interfaces	CLI, REST (TOWER)	CLI, Rest, JSON RPC, Netconf

Conclusion



Ansible + Cisco = Unlimited Potential

- Simple, Stateless, Powerful yet open-source automation Framework
- Quick Win scenarios for automation
- Best suited for large size automation teams than other dev-ops options like python, NorNir etc. YAML is the key.
- Limited desire or skills to adopt python programming
- Multi-vendor support utilizing vendor developed & maintained network modules.
- All that you need is a workstation/VM and start with



Happy Automating with Ansible!!

Keynote Deep Dives

Wednesday
10:30am – 11:30am



Experiences Amplified:
How AI Can Fuel
Better Employee
and Customer
Experiences

Level 1
Room 106



Smart, Secure,
Seamless:
Transforming
Experiences with
Next-Generation
Networking

Level 2
Room 204



Harness a Bold
New Era:
Transform Data
Centre and
Service Provider
Connectivity

Level 2
Room 203



Securing User to
Application and
Everything in
Between

Level 2
Melbourne Room 2



Unlocking
Digital Resilience
through Unified
Observability

The HUB
Centre Stage

Complete Your Session Evaluations



Complete a minimum of 4 session surveys and the Overall Event Survey to claim a **Cisco Live T-Shirt**.



Complete your surveys in the **Cisco Live mobile app**.



Continue your education



- Visit the Cisco Stand for related demos
- Book your one-on-one Meet the Expert meeting
- Attend the interactive education with DevNet, Capture the Flag, and Walk-in Labs
- Visit the On-Demand Library for more sessions at www.CiscoLive.com/on-demand



Thank you

CISCO *Live!*

#CiscoLiveAPJC

CISCO *Live!*

GO BEYOND

#CiscoLiveAPJC