Red Hat | ANSIBLE

#ANSIBLEAUTOMATES MOSCOW 2019

ANSIBLE NETWORK AUTOMATION

Victor da Costa
Principal Technical Business Development, CCIE #39373
Twitter/@victorobotz, Github/@victorock

“Automation is perhaps one of the highest yielding investments an entity running its network can make.”

Gartner, Oct 2019

What is the primary method of making network changes in you environment ?

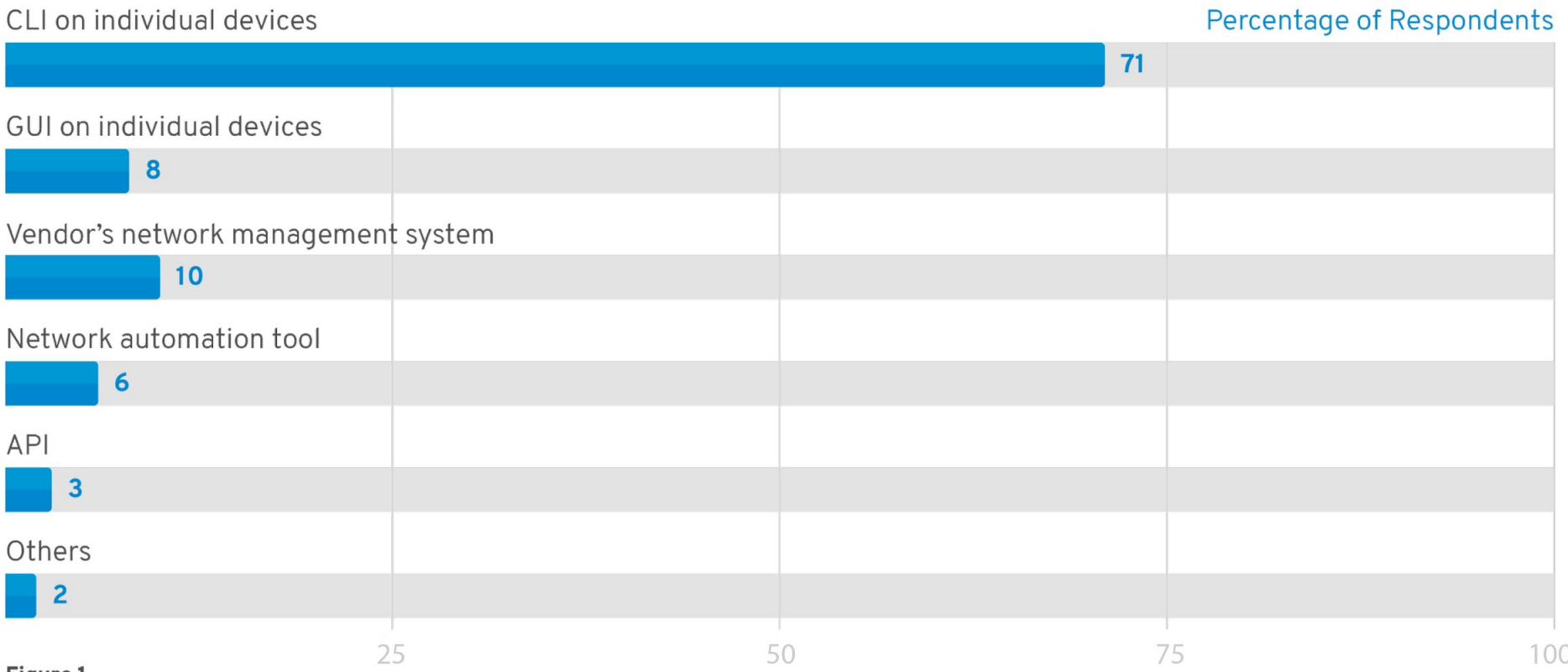


Figure 1
Primary Method for Making Network Changes

Source: Gartner, *Look Beyond Network Vendors for Network Innovation.* January 2018. Gartner ID: G00349636. (n=64)

Why hasn’t management of networking changed ?

PEOPLE

Domain specific skill sets

Vendor oriented experience

Siloed organizations

Legacy operational practices

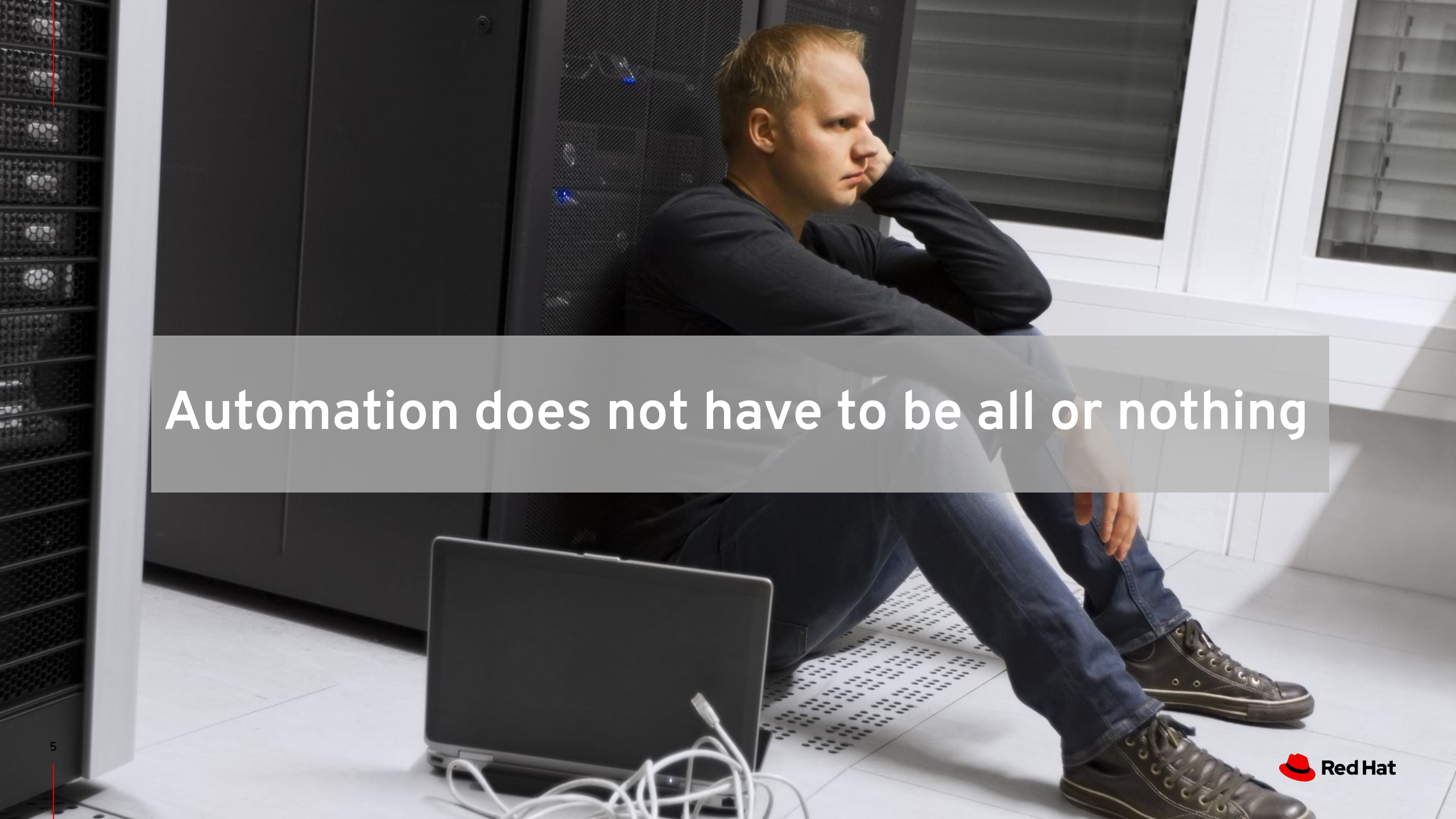
PRODUCTS

Infrastructure-focused features

Baroque, CLI-only methodologies

Siloed technologies

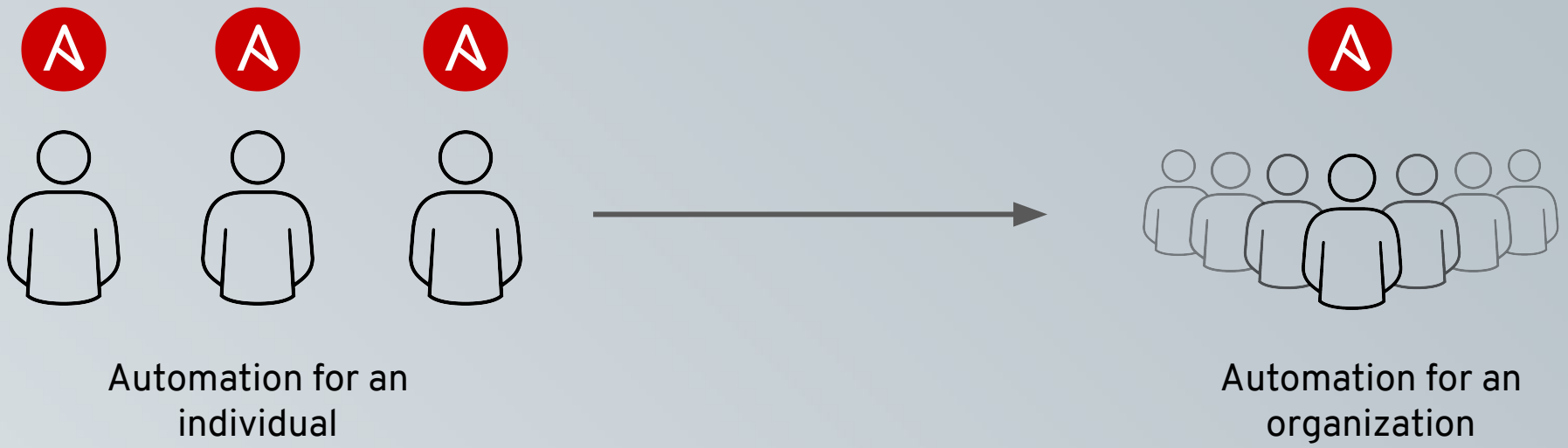
Monolithic, proprietary platforms



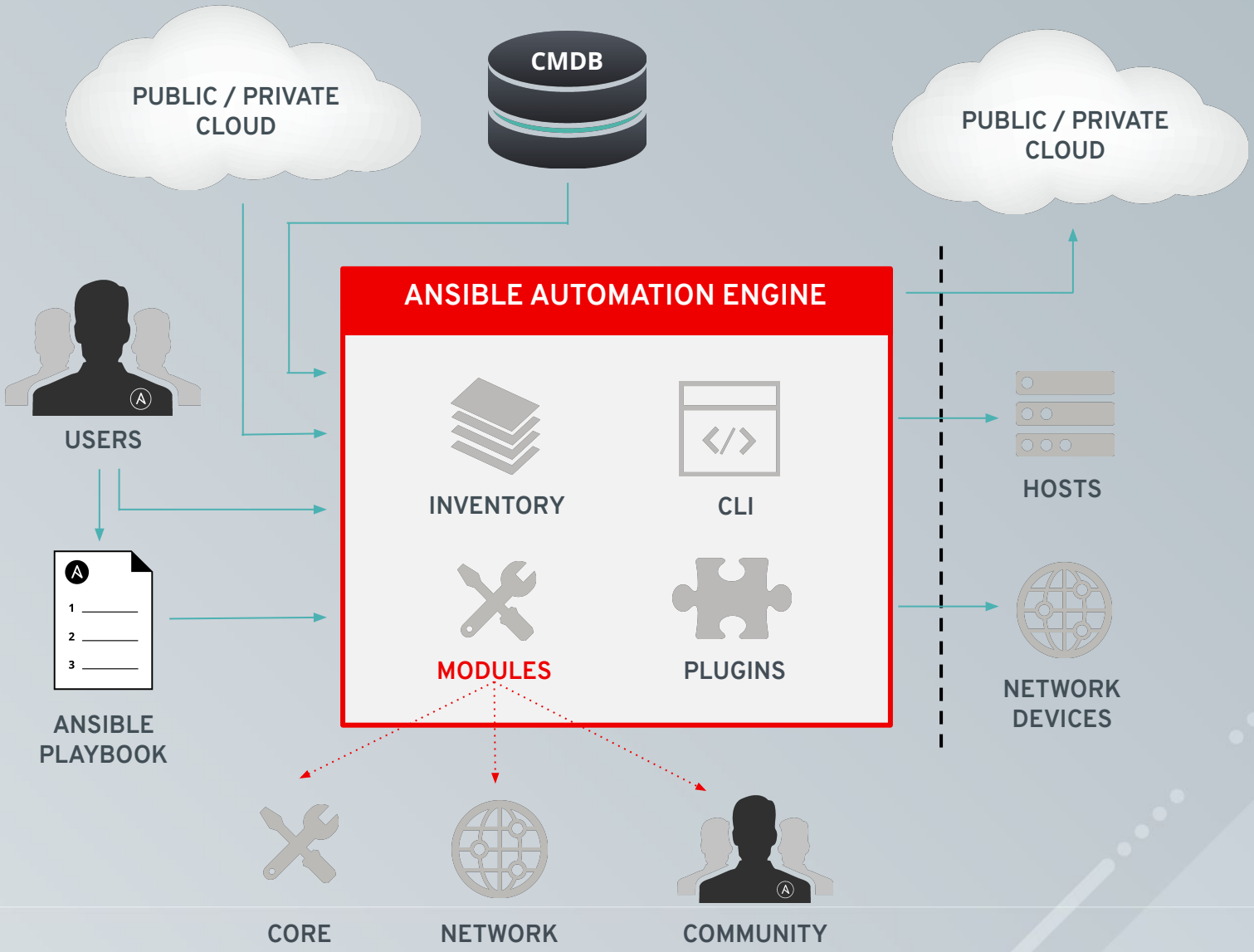
Automation does not have to be all or nothing

Red Hat

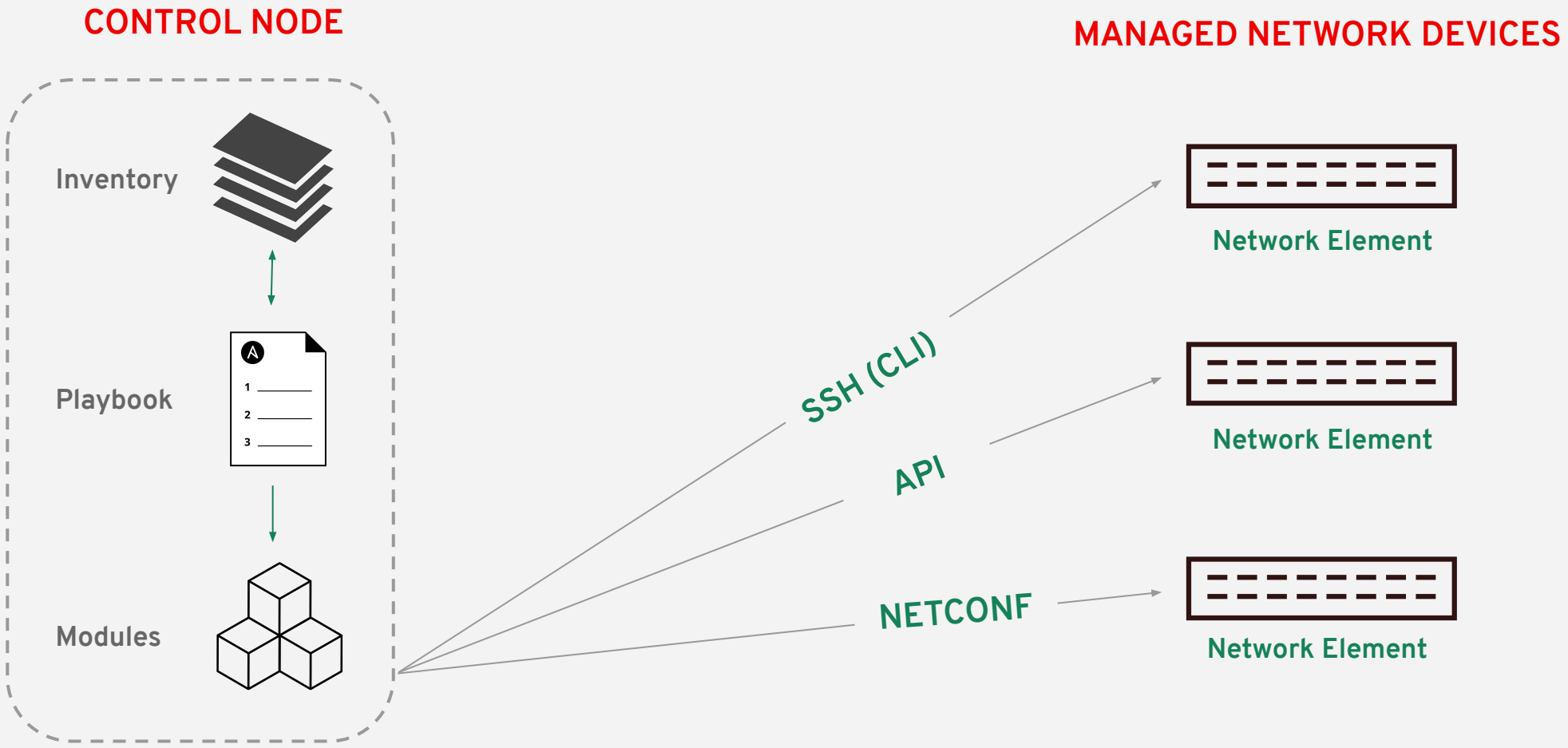
Automation is a cultural journey



How does Ansible Works?



Just like YOU, but at SCALE



ANSIBLE NETWORK AUTOMATION

65+
Network Platforms

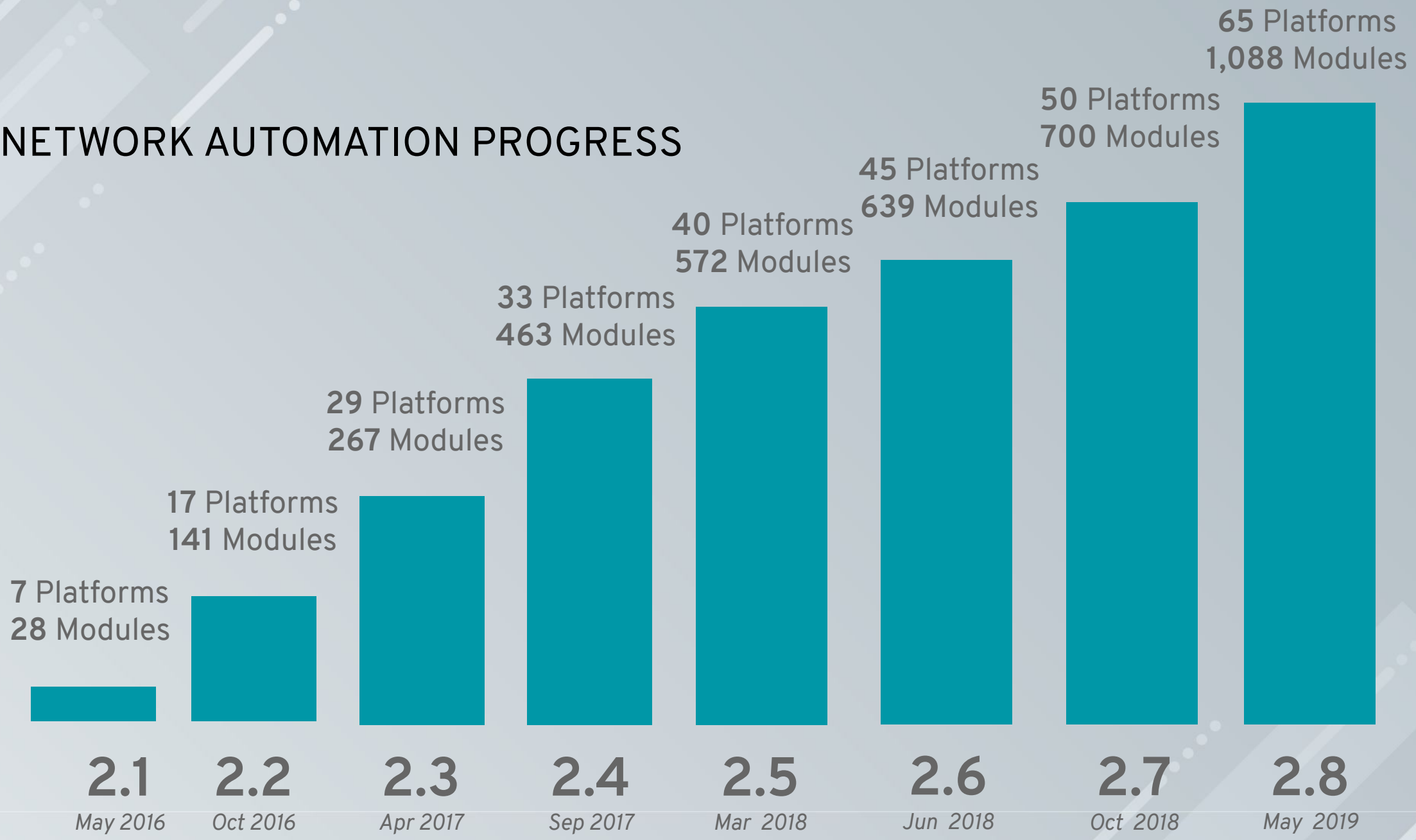
1,000+
Network Modules

15+
Galaxy Network Roles

ansible.com/for/networks
galaxy.ansible.com/ansible-network

Ansible Network modules comprise 1/3 of all modules that ship with Ansible Engine

NETWORK AUTOMATION PROGRESS



Network Platforms

A10

Apstra AOS

Arista EOS, CVP

Aruba ArubaOS

AVI Networks

Big Switch Networks

Brocade Ironware

Cisco ACI, AireOS, ASA, FTD, IOS, IOS-XR, Meraki, NSO, NX-OS

Check Point

Citrix Netscaler

Cumulus Linux

Dell OS6, OS9, OS10

Exoscale

Extreme EX-OS, NOS, SLX-OS, VOSS

F5 BIG-IP, BIG-IQ

Fortinet FortIOS, FMGR

FRR (Free Range Routing)

Huawei CloudEngine

Illumos

Infoblox NIOS

Ingate SBC

Itential

Juniper JunOS

Lenovo CNOS, ENOS

Mellanox ONYX

MikroTik RouterOS

Openswitch (OPX)

Ordnance

NETCONF

Netvisor

Nokia NetAct, SR OS

Nuage VSP

OpenSwitch

Open vSwitch (OVS)

Palo Alto PAN-OS

RESTCONF

Skydive

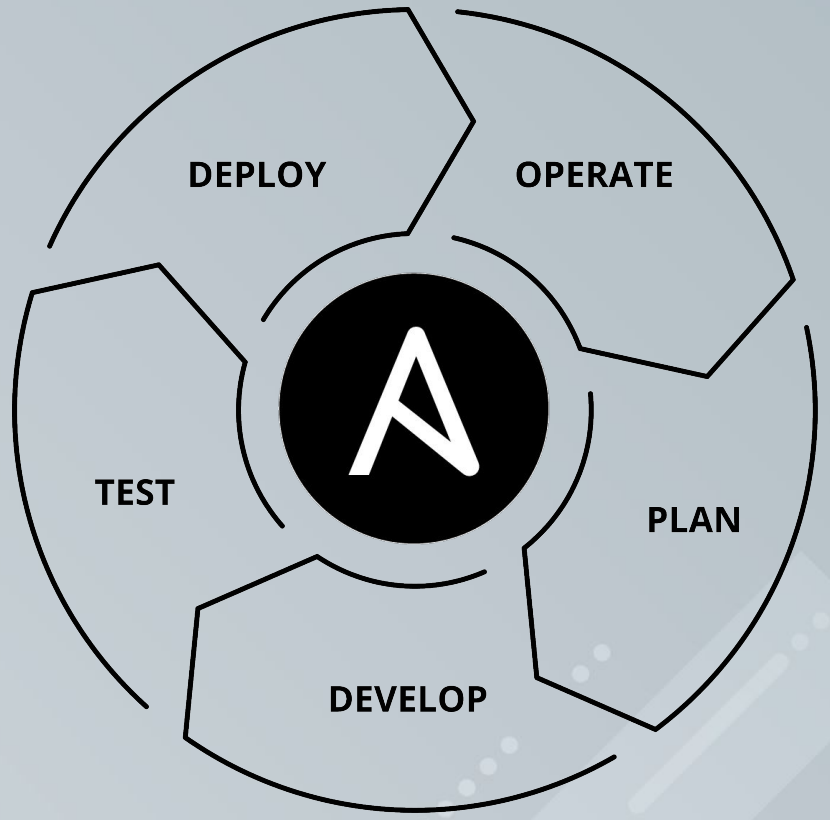
Ubiquiti EdgeOS, Edgeswitch

VyOS

https://docs.ansible.com/ansible/latest/modules/list_of_network_modules.html

How to get there ?
Think Big, Start Small, Iterate Over!

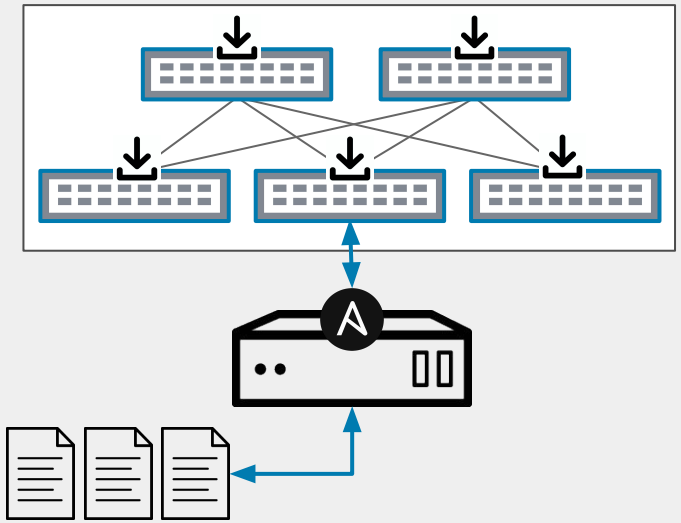
1. **Backup and Restore**
2. **Fact Collection**
3. **Configuration Management**
4. **Creating a Source of Truth**
5. **Third-Party Integrations**



Backup and Restore

Why is it important?

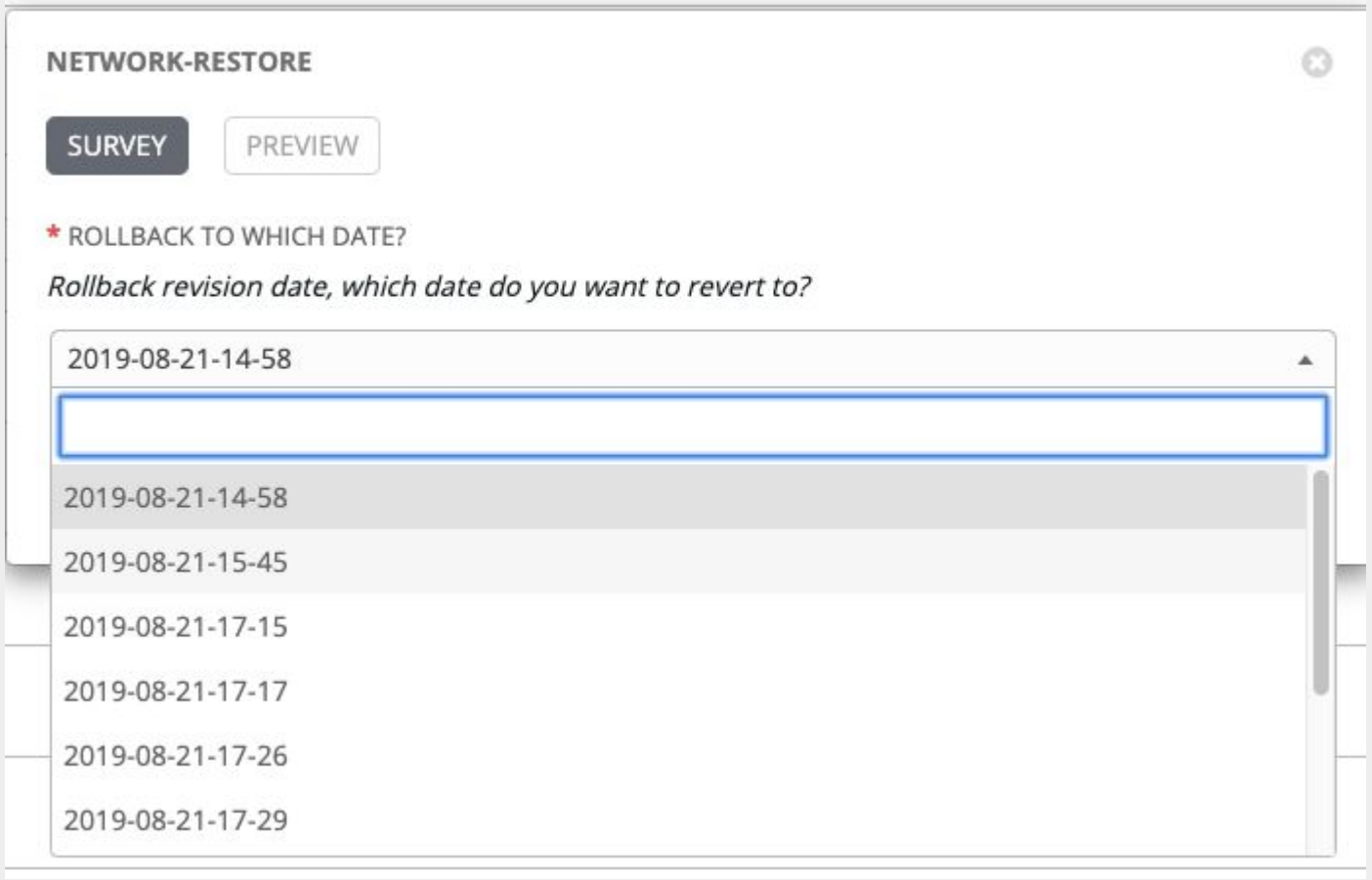
- Read-only, not changing production configs
- Ubiquitous use-case
- Easy scheduling in Ansible Tower



Red Hat

Combine with Ansible Tower Surveys

- No platform specific knowledge required!
- Automate the routine and boring tasks away
- Empower junior engineers to take on more activities with guard rails

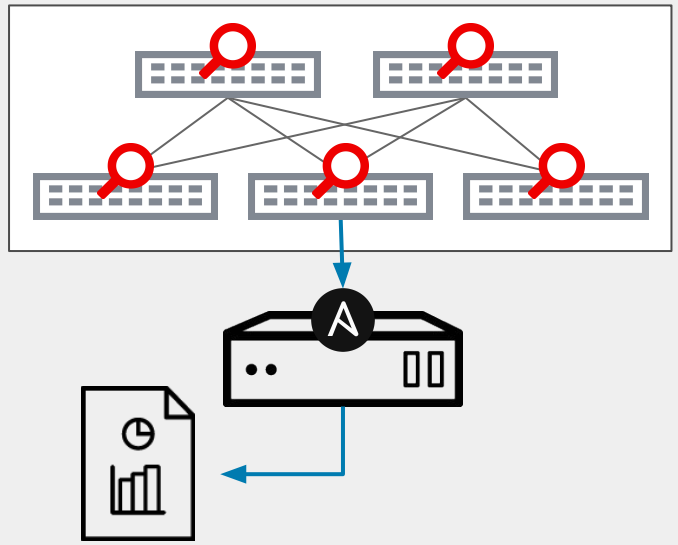


Red Hat

Fact Collection

Why is it important?

- Read-only, not changing production configs
- Structured Data
- Build reports for easy consumption



Configuration Facts

Network Native Configuration (On box)

Convert to structured data

Example facts parameter
gather_network_resources: interfaces

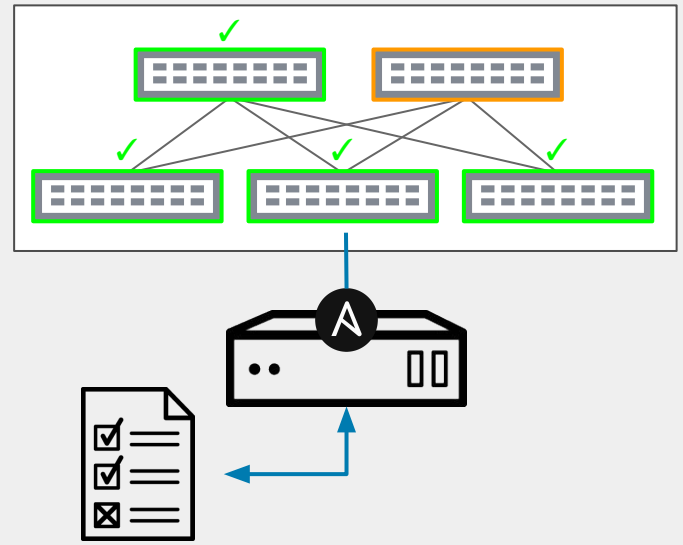
```
ansible_facts:
  ansible_net_api: cliconf
  ansible_net_fqdn: rtr2
  ansible_net_gather_network_resources:
  - interfaces
  ansible_net_gather_subset:
  - default
  ansible_net_hostname: rtr2
  ansible_net_image: flash:EOS.swi
  ansible_net_model: vEOS
  ansible_net_python_version: 2.7.5
  ansible_net_serialnum:
D00E130991A37B49F970714D8CCF7FCB
  ansible_net_system: eos
  ansible_net_version: 4.22.0F
  ansible_network_resources:
    interfaces:
    - enabled: true
      name: Ethernet1
    - enabled: true
      name: Loopback0
<<rest of output removed for slide brevity>>
```

Red Hat

Config Management

Why is it important?

- Enforce configuration policy
- Correct configuration drift
- Force multiplier for config changes
- Lockdown configs to best practices



Configuration Management Methods

- **Config modules with raw configs**

```
- name: ensure that the desired snmp strings are present
  ios_config:
    commands:
      - snmp-server community ansible-public RO
      - snmp-server community ansible-private RW
```

- **Config modules with jinja2 templating**

```
- name: ensure that the desired snmp strings are present
  ios_config:
    config: "{{ lookup('template', 'template.j2') }}"
```

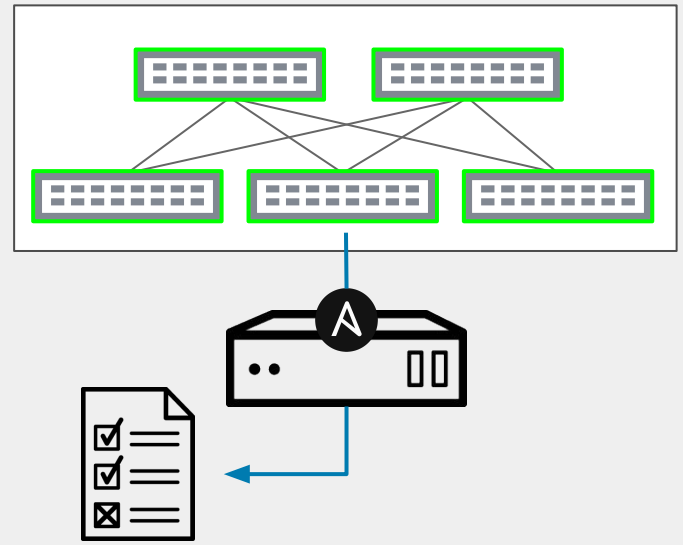
- **Resource modules**

```
- name: ensure that the IP address information is accurate
  ios_l3_interfaces:
    config: "{{ ip_address_info }}"
```

Creating a Source of Truth

Why is it important?

- Incremental steps to infrastructure as code
- Structured variables for network config
- Simple and agnostic data model

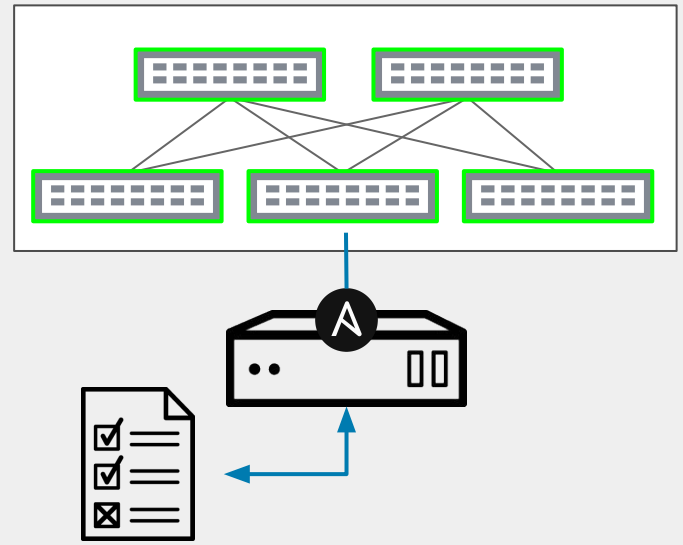


Create a structured SOT (source of truth)

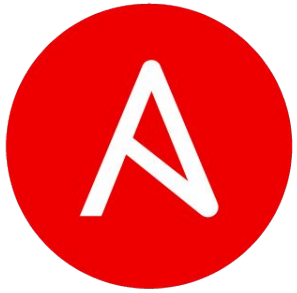
Third-party Integrations

Why is it important?

- Brownfield networks have brownfield tools
- Tower's API is a killer feature
- Ansible Tower has "in the box" integrations



Red Hat



+ servicenow

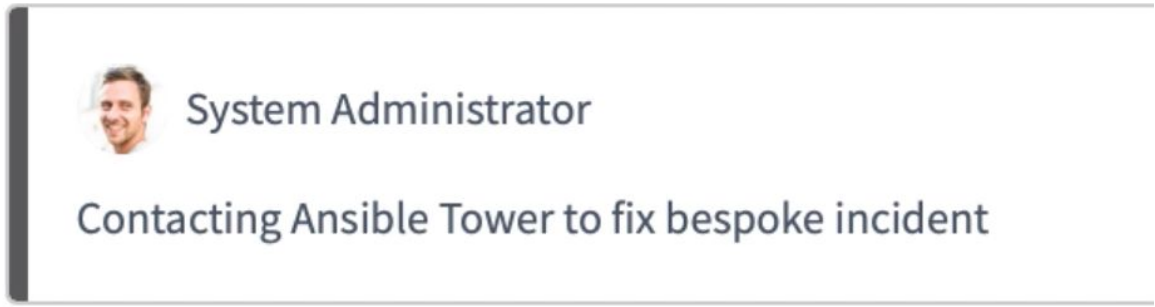
https://ansible.github.io/workshops/demos/

Ansible Tower uses `snow_record` module

```
TASK [MARK THE TICKET AS RESOLVED] **********
******************
changed: [rtr1]

TASK [MARK THE TICKET AS CLOSED] ************
******************
changed: [rtr1]
```

ServiceNow uses the Ansible Tower API



Red Hat

#ANSIBLEAUTOMATES

redhat. | ANSIBLE

USERS

ITSM

BUSINESS RULES

SERVICE CATALOG

CMDB

Business Process Automation

PROVISIONING

CONFIGURATION MANAGEMENT

APP DEPLOYMENT

CONTINUOUS DELIVERY

SECURITY & COMPLIANCE

DEVELOPERS + OPERATORS

ANSIBLE TOWER

ROLE-BASED ACCESS CONTROL

KNOWLEDGE & VISIBILITY

SCHEDULED & CENTRALIZED JOBS

SIMPLE USER INTERFACE

TOWER API

ANSIBLE ENGINE

OPEN SOURCE MODULE LIBRARY

PLUGINS

PYTHON CODEBASE

IT Automation

TRANSPORT

SSH, WINRM, ETC.

AUTOMATE YOUR ENTERPRISE

INFRASTRUCTURE

LINUX, WINDOWS, UNIX ...

NETWORKS

ARISTA, CISCO, JUNIPER ...

CONTAINERS

DOCKER, LXC ...

CLOUD

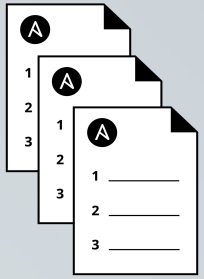
AWS, GOOGLE CLOUD, AZURE ...

SERVICES

DATABASES, LOGGING, SOURCE CONTROL MANAGEMENT...

ENGINEERING

DEFINITION



+

OPERATIONS

IMPLEMENTATION

```
system:
 hostname: "{{inventory_hostname}}"
 domain_name: eng.ansible.com

 source_interface:
   name: Management1
   vrf: default

 domain_lookup: no

 name_servers:
   - 1.1.1.1
   - 2.2.2.2

vlan_data:
  - { id: 600, name: management }
  - { id: 601, name: users }
```

=

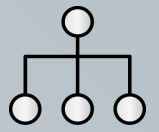
PRODUCTION

INFRASTRUCTURE

Applications

Servers

Storage



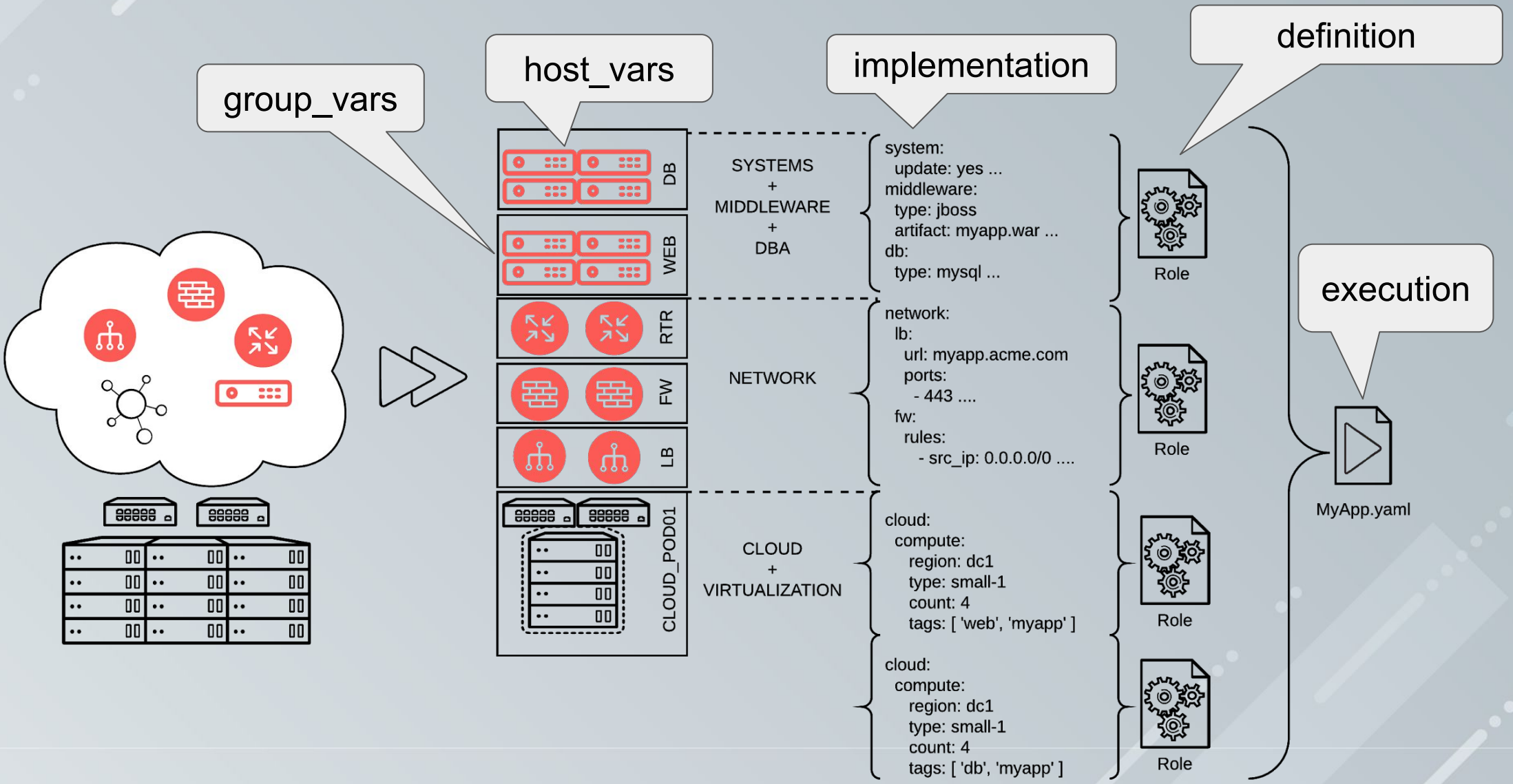
Network

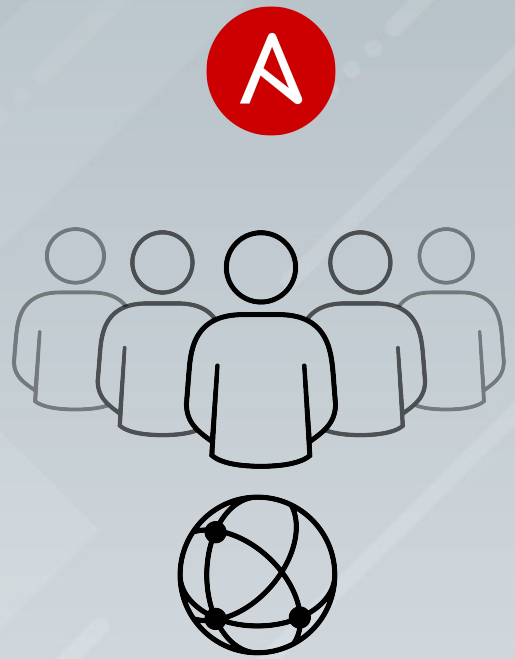
Function

Input

Desired State

COMPOSABLE INFRASTRUCTURE





DEMO?

Workshop: https://github.com/ansible/workshops
Community: https://github.com/network-automation/

AUTOMATION FOR EVERYONE: **NETWORK ENGINEERS**

```
---
- hosts: cisco
  gather_facts: false
  connection: network_cli

  tasks:
   - name: show command for cisco
     cli_command:
        command: show ip int br
     register: result

   - name: display result to terminal window
     debug:
        var: result.stdout_lines
```

AUTOMATION FOR EVERYONE: **PLAYBOOK RESULTS**

```
[student3@ansible network_setup]$ ansible-playbook example.yml

PLAY [cisco] ***********************************************************************************************************************************************************************

TASK [show command for cisco] ******************************************************************************************************************************************************
ok: [rtr2]
ok: [rtr1]

TASK [display result to terminal window] *******************************************************************************************************************************************
ok: [rtr1] => {
    "result.stdout_lines": [
        "Interface              IP-Address      OK? Method Status                Protocol",
        "GigabitEthernet1       172.16.22.120   YES DHCP   up                    up      ",
        "VirtualPortGroup0      192.168.35.101  YES TFTP   up                    up"
    ]
}
ok: [rtr2] => {
    "result.stdout_lines": [
        "Interface              IP-Address      OK? Method Status                Protocol",
        "GigabitEthernet1       172.17.1.107    YES DHCP   up                    up      ",
        "VirtualPortGroup0      192.168.35.101  YES TFTP   up                    up"
    ]
}

PLAY RECAP *************************************************************************************************************************************************************************
rtr1                       : ok=2    changed=0    unreachable=0    failed=0    skipped=0
rtr2                       : ok=2    changed=0    unreachable=0    failed=0    skipped=0

[student3@ansible network_setup]$
```

AUTOMATION FOR EVERYONE: **NETWORK ENGINEERS**

```
---
- hosts: juniper
  gather_facts: false
  connection: network_cli

  tasks:
   - name: show command for juniper
     cli_command:
        command: show interfaces terse em1
     register: result

   - name: display result to terminal window
     debug:
        var: result.stdout_lines
```

AUTOMATION FOR EVERYONE: **PLAYBOOK RESULTS**

```
[student3@ansible network_setup]$ ansible-playbook junos-example.yml

PLAY [juniper] *****************************************************************************************************************************************

TASK [show command for juniper] ************************************************************************************************************************
ok: [rtr3]
ok: [rtr4]

TASK [display result to terminal window] ***************************************************************************************************************
ok: [rtr3] => {
    "result.stdout_lines": [
        "Interface               Admin Link Proto    Local                 Remote",
        "em1                     up    up",
        "em1.0                   up    up   inet     10.0.0.4/8      ",
        "                                   128.0.0.1/2     ",
        "                                   128.0.0.4/2     ",
        "                                inet6    fe80::5254:ff:fe12:bdfe/64",
        "                                   fec0::a:0:0:4/64",
        "                                tnp      0x4"
    ]
}
ok: [rtr4] => {
    "result.stdout_lines": [
        "Interface               Admin Link Proto    Local                 Remote",
        "em1                     up    up",
        "em1.0                   up    up   inet     10.0.0.4/8      ",
        "                                   128.0.0.1/2     ",
        "                                   128.0.0.4/2     ",
        "                                inet6    fe80::5254:ff:fe12:bdfe/64",
        "                                   fec0::a:0:0:4/64",
        "                                tnp      0x4"
    ]
}

PLAY RECAP *********************************************************************************************************************************************
rtr3                       : ok=2    changed=0    unreachable=0    failed=0    skipped=0
rtr4                       : ok=2    changed=0    unreachable=0    failed=0    skipped=0

[student3@ansible network_setup]$
```