ANSIBLE

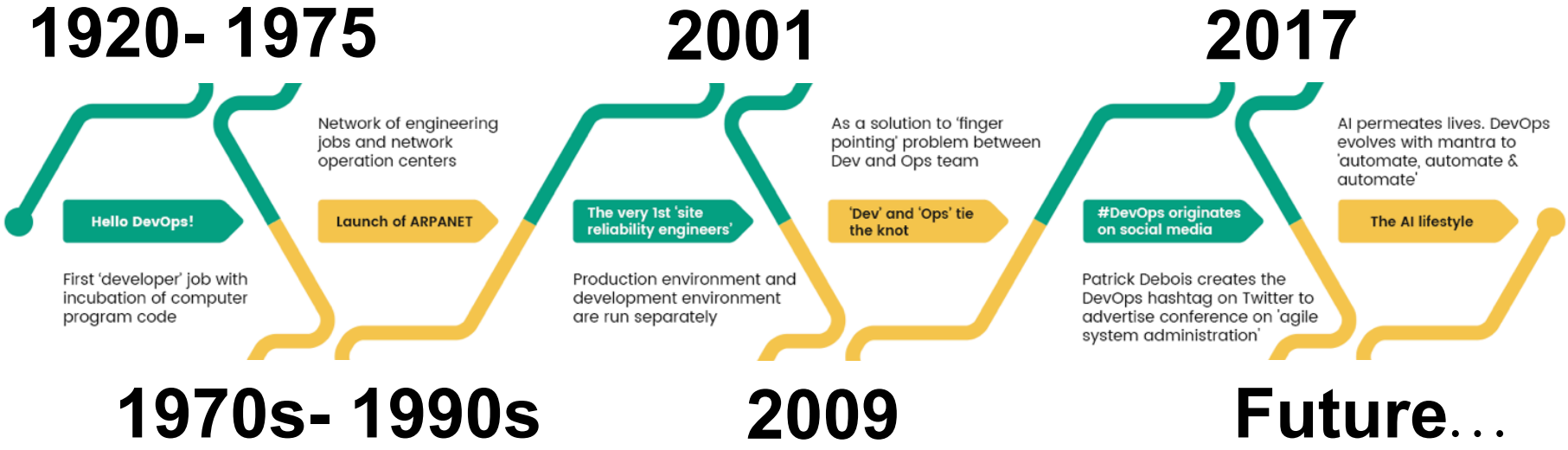
Ansible Network Automation

Faz Sadeghi
Specialist Solution Architect
Red Hat Ansible Automation
faz@redhat.com

Outline

- Why network Automation
- Why Ansible
- Sample examples
- Ansible Tower

Increase of consistency
, treat net like servers, pet vs cattle
Called at 1:00 which edge switch was connected to what because someone connected something somewhere by mistke and npw there is a spanning tree loop

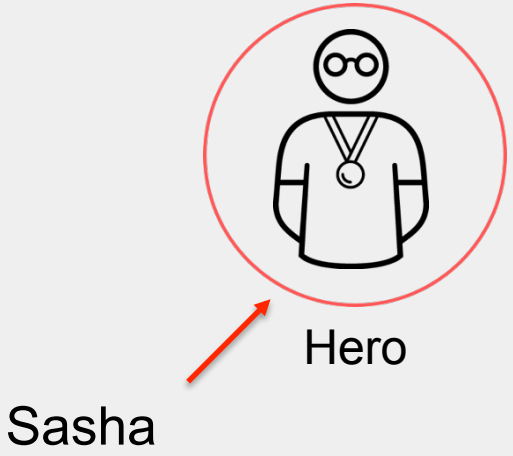

1920- 1975
Hello DevOps!
First 'developer' job with incubation of computer program code
1970s- 1990s
Network of engineering jobs and network operation centers
Launch of ARPANET
2001
The very 1st 'site reliability engineers'
Production environment and development environment are run separately
2009
As a solution to 'finger pointing' problem between Dev and Ops team
'Dev' and 'Ops' tie the knot
2017
#DevOps originates on social media
Patrick Debois creates the DevOps hashtag on Twitter to advertise conference on 'agile system administration'
Future...
AI permeates lives. DevOps evolves with mantra to 'automate, automate & automate'
The AI lifestyle


MANAGING NETWORKS
HASN'T CHANGED
IN 30 YEARS.

Sasha is a HERO!

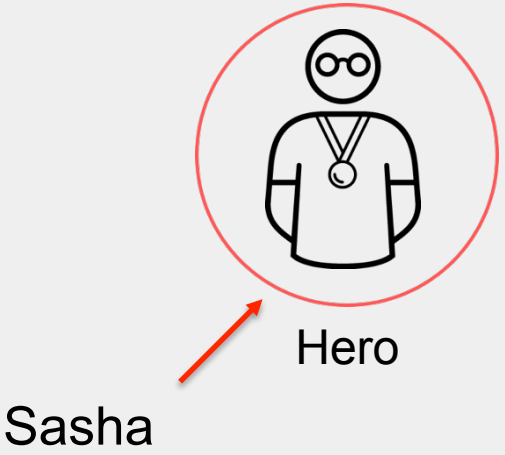
1) Sasha designs the network.

Sasha is your HERO!



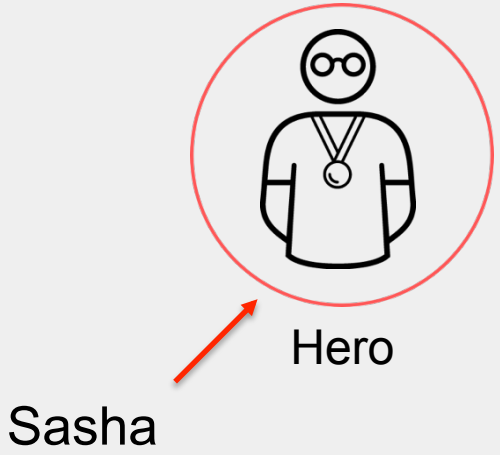
1) Sasha designs the network.
2) Sasha builds the network.

Sasha is your HERO!



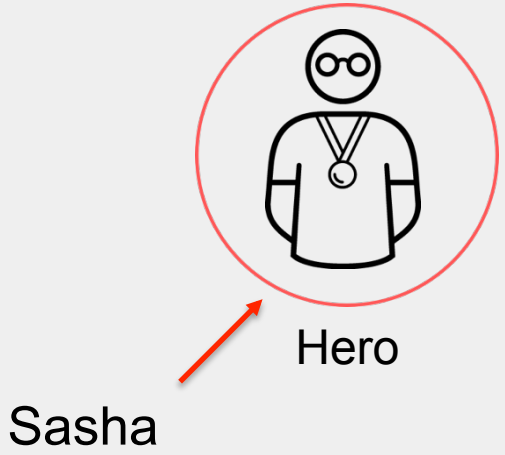
1) Sasha designs the network.
2) Sasha builds the network.
3) Sasha fixes the network.

Sasha is your HERO!



1) Sasha designs the network.
2) Sasha builds the network.
3) Sasha fixes the network.
4) Sasha deploys WIFI at the summer house by the beach.

Sasha is your HERO!

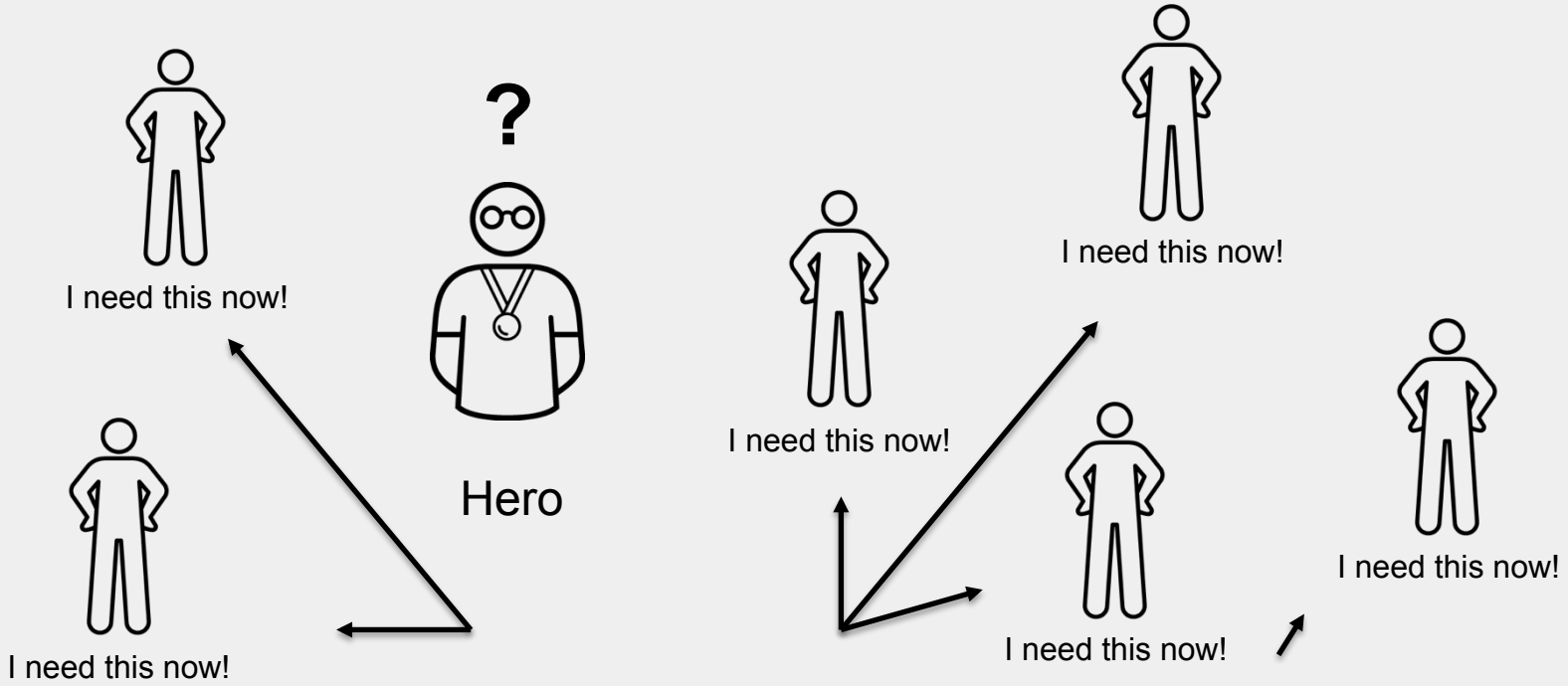


1) Sasha designs the network
2) Sasha builds the network
3) Sasha fixes the network
4) Sasha deploys WIFI at the VP's lake house.

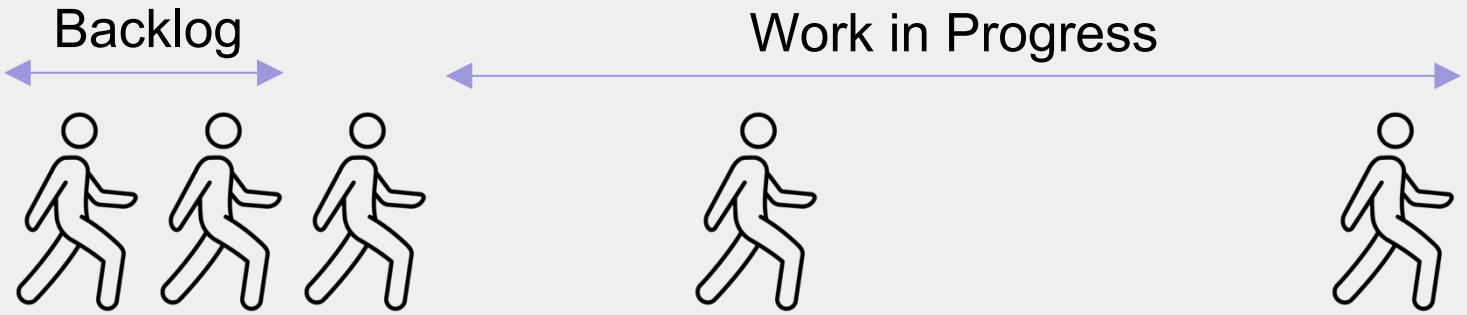
Sasha does EVERYTHING!

So… what's the problem?

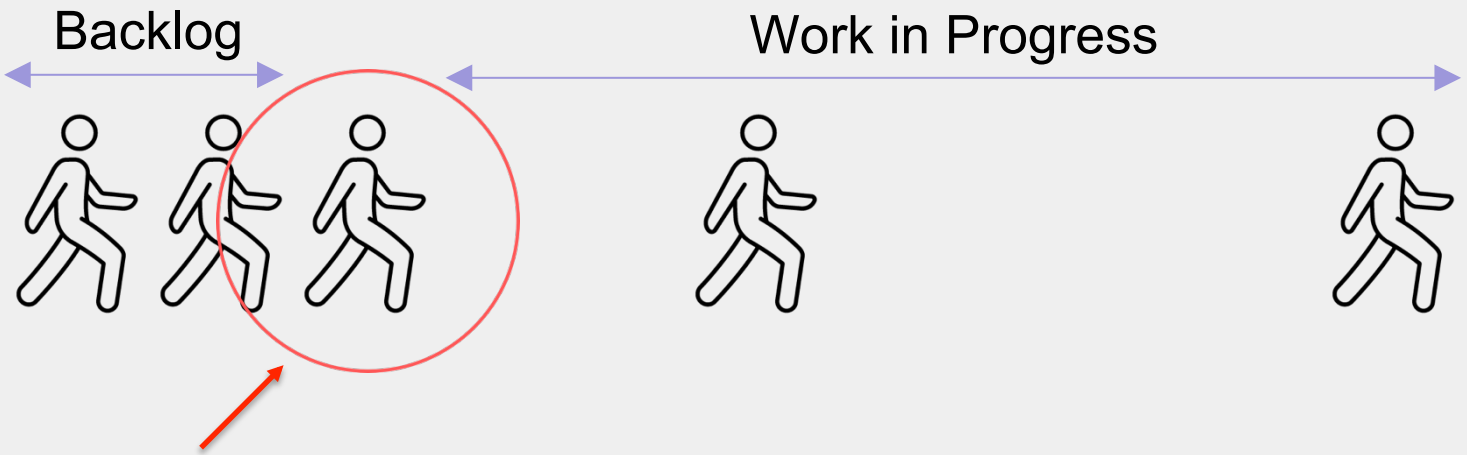
Heroes are in high demand



The most important person in the World



$$Cycle\ Time = \frac{WIP}{Throughput}$$



Sasha is slowing you down!

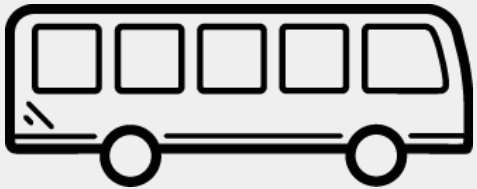
$$Cycle\ Time = \frac{WIP}{Throughput}$$

Sasha is your Bottleneck.

Sasha is slowing you down!

Or…

Hero




Sasha joins
a competitor
company

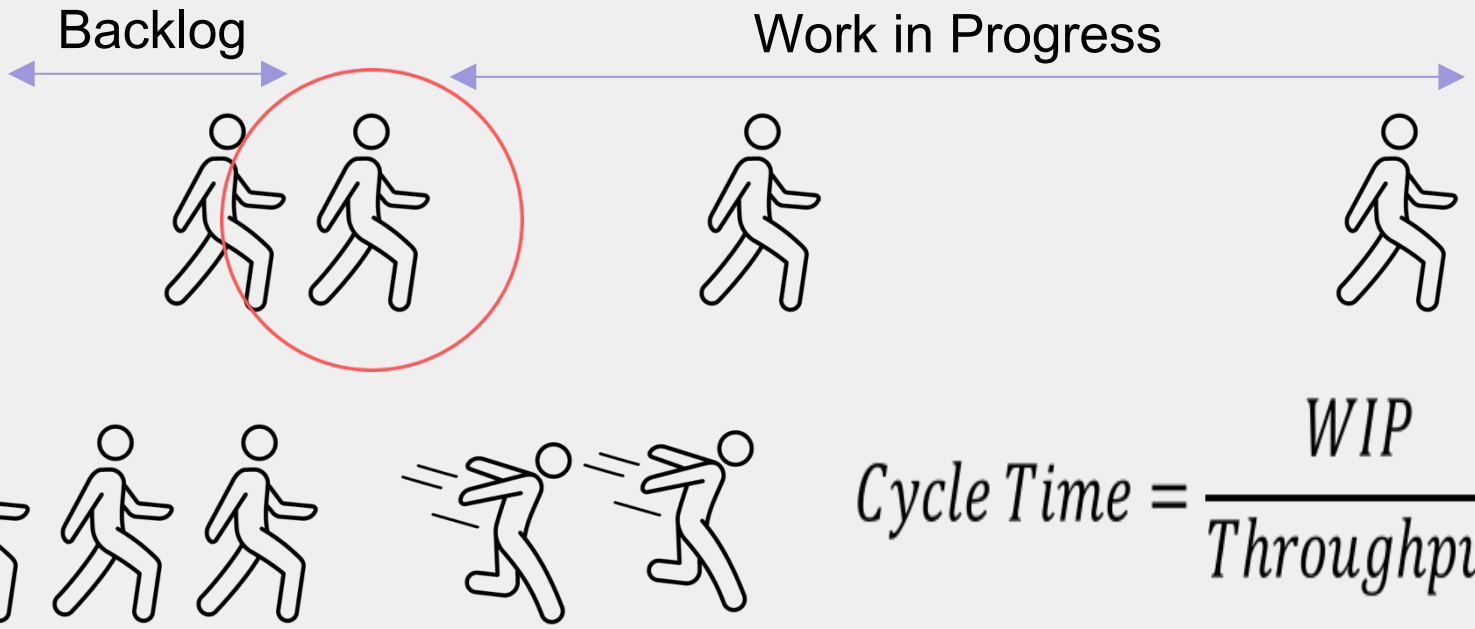
What do you do?

Option #1: Reduce WIP

$$Cycle\ Time = \frac{WIP}{Throughput}$$

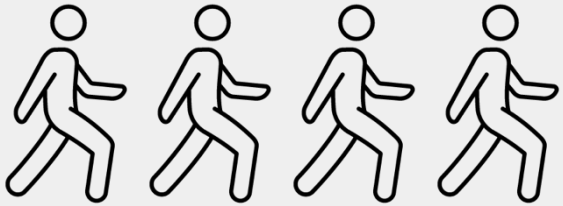
Increase throughput by
decreasing demand on Sasha

Increase Throughput



$$Cycle\ Time = \frac{WIP}{Throughput}$$

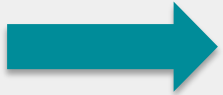
Increase Throughput

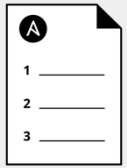


$$Cycle\ Time = \frac{WIP}{Throughput}$$

Automation: Hero as Code

Hero





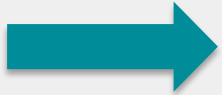
Code

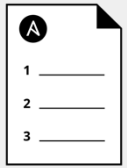
- Leverages Human Experience
- Reduce Repetition

- Reduce Variability
- Reduce Isolation

Automation: Hero as Code

Playbook





Hero

Code

Playbook

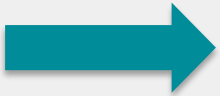
- Leverages Human Experience
- Reduce Repetition

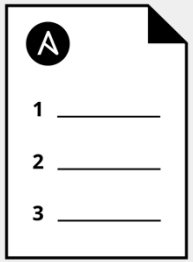
- Reduce Variability
- Reduce Isolation

Convert Procedures to Playbooks

1. Create VLAN
2. Add port to VLAN
3. Address Interface

Method of Procedure

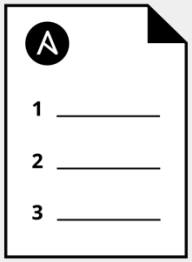


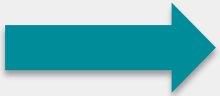


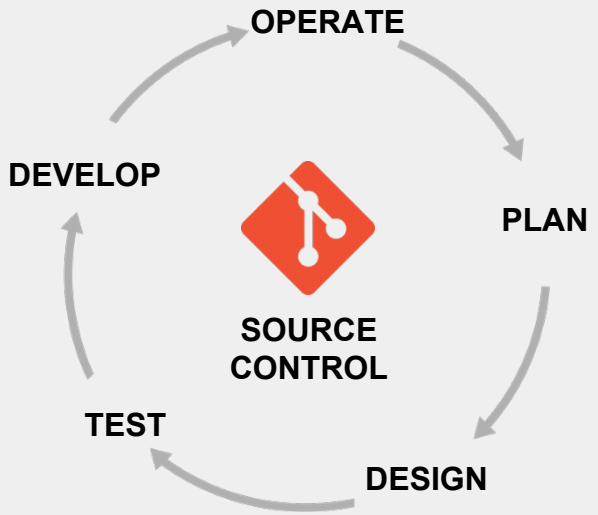
Playbook

- Define Intent, Policy, Architecture
- Apply across device type, vendor

Manage Lifecycle with Process & Playbooks

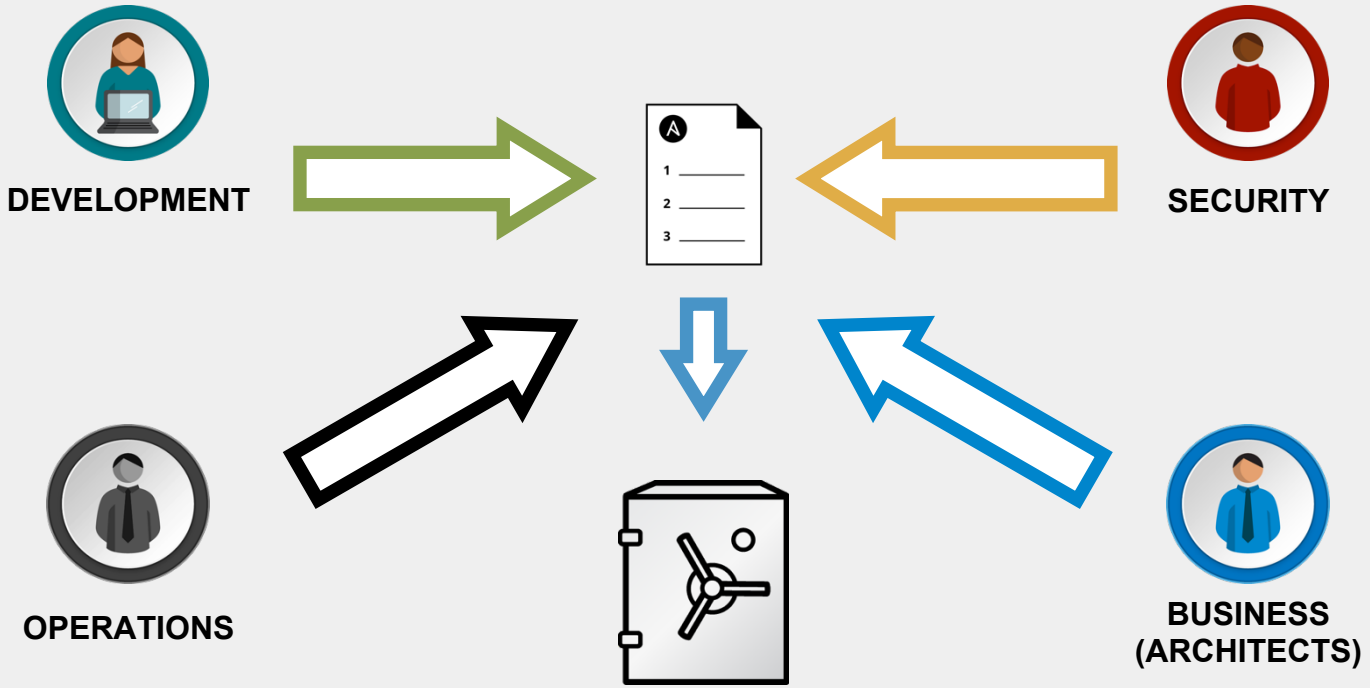






- Revision control, configuration management
- Ensure an ongoing steady-state
- Automated testing, reduce human error

Communicate with Playbooks



YAML Is Easy to Understand

```
- name: Start NGiNX
    service:
      name: nginx
      state: started
```

Why Ansible?

SIMPLE

Human readable automation

No special coding skills needed

Tasks executed in order

Get productive quickly

POWERFUL

Image updates

Configuration management

Compliance

Orchestrate the network lifecycle

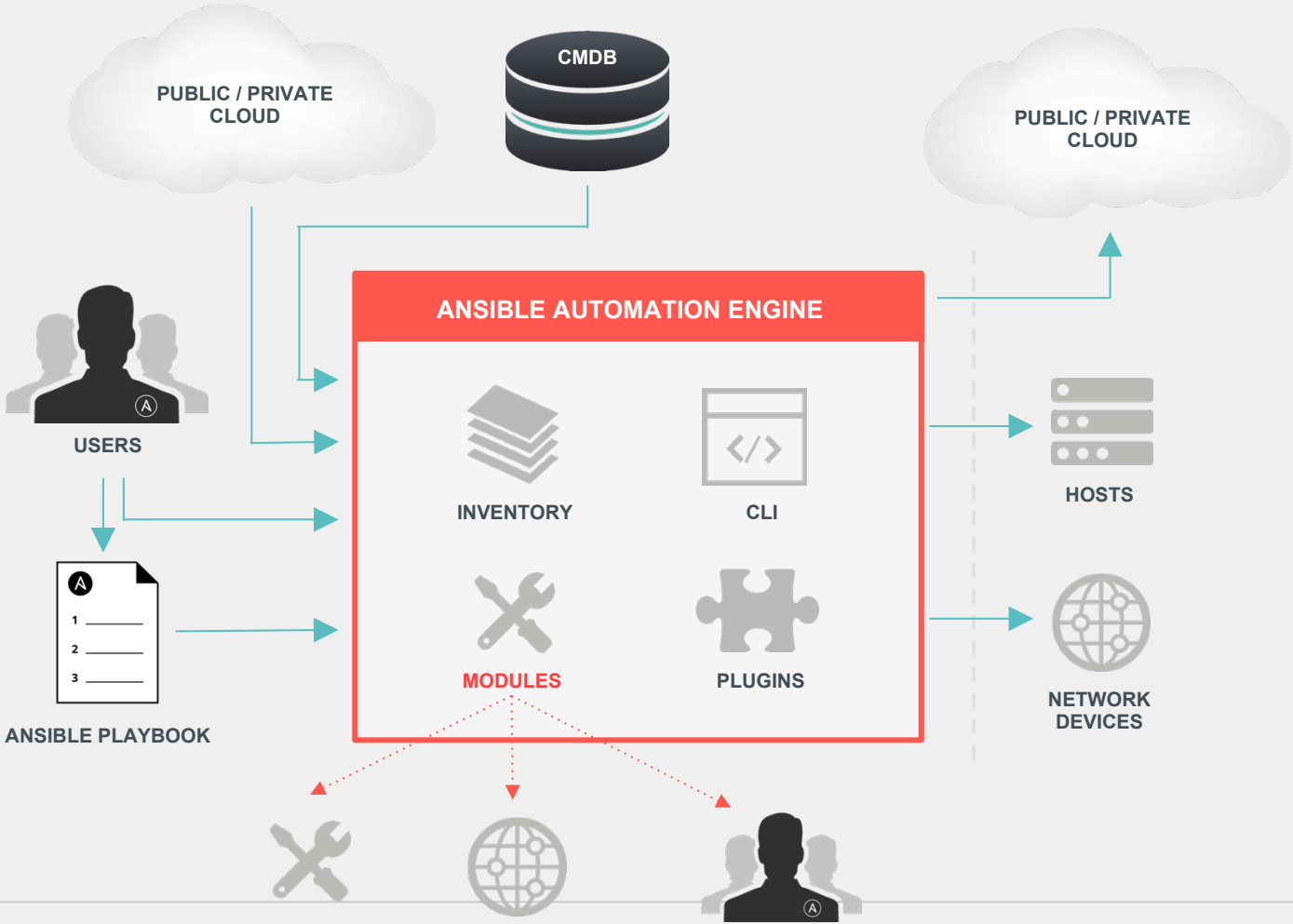
AGENTLESS

Agentless architecture

Uses OpenSSH & WinRM

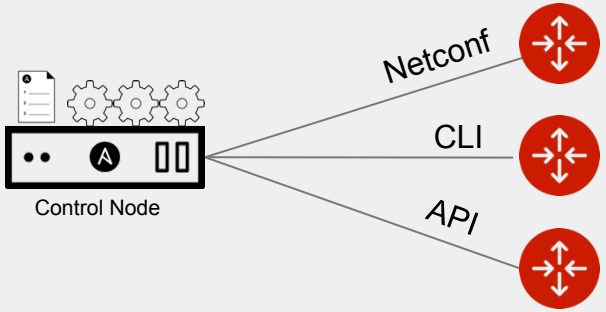
No agents to exploit or update

More efficient & more secure


PUBLIC / PRIVATE CLOUD
CMDB
PUBLIC / PRIVATE CLOUD
ANSIBLE AUTOMATION ENGINE
USERS
INVENTORY
CLI
HOSTS
1
2
3
MODULES
PLUGINS
NETWORK DEVICES
ANSIBLE PLAYBOOK

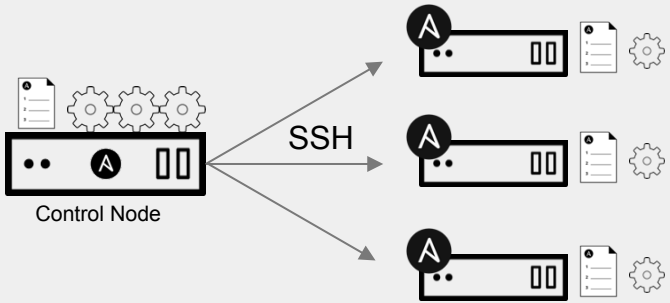
Connection Plugins

Python code is executed locally on the control node



NETWORKING DEVICES

Python code is copied to the managed node, executed, then removed



LINUX HOSTS

NETWORK MODULES: BUILT-IN DEVICE ENABLEMENT

A10

Apstra AOS

Arista EOS, CVP

Aruba Networks

AVI Networks

Big Switch Networks

Brocade Ironware

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

Citrix Netscaler

Cumulus Linux

Dell OS6, OS9, OS10

Exoscale

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

F5 BIG-IP, BIG-IQ

Fortinet FortIOS, FMGR

Huawei CloudEngine

Illumos

Infoblox NIOS

Juniper JunOS

Lenovo CNOS, ENOS

Mellanox ONYX

MikroTik RouterOS

Openswitch (OPX)

Ordnance

NETCONF

Netvisor

OpenSwitch

Open vSwitch (OVS)

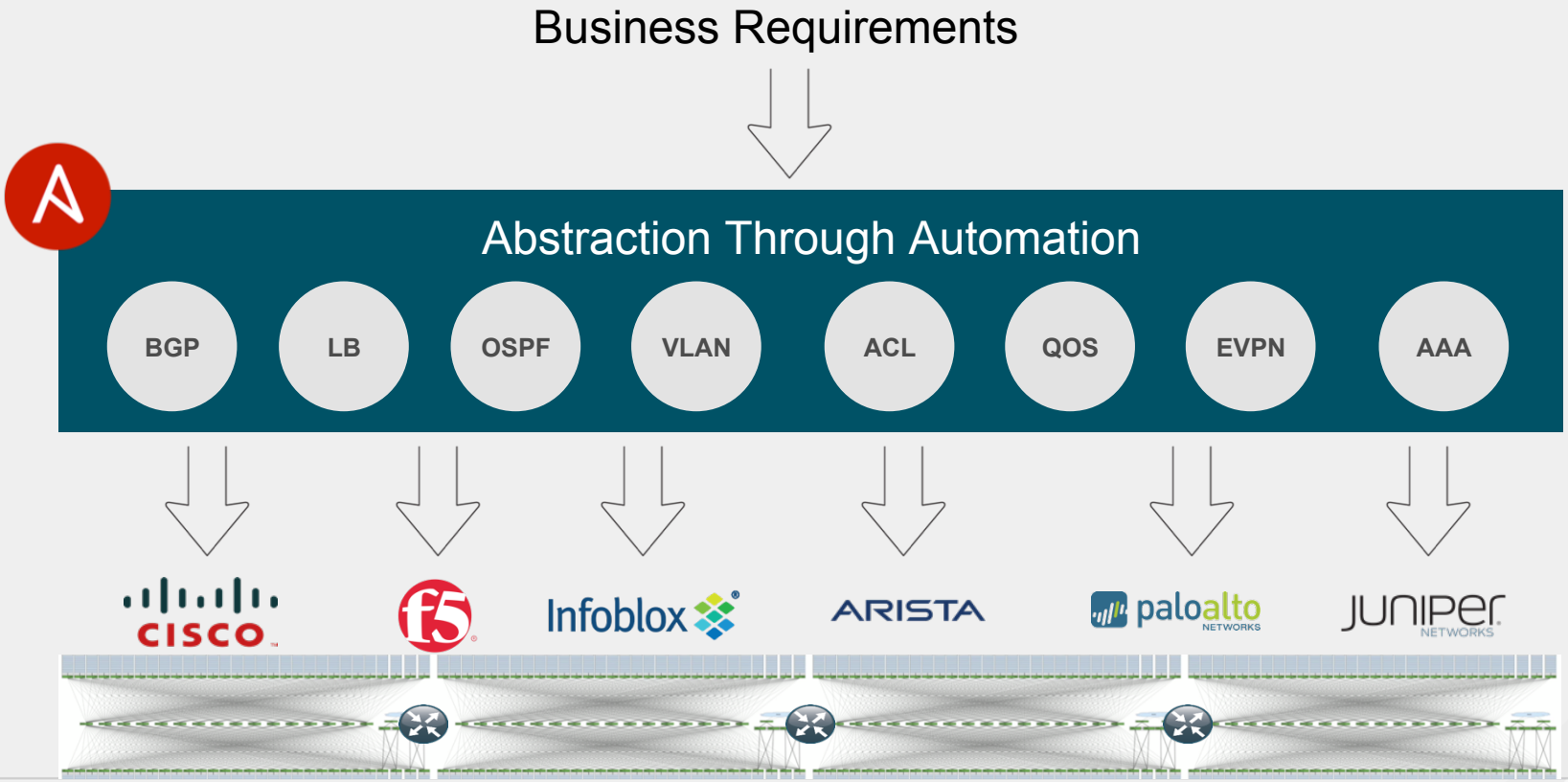
Palo Alto PAN-OS

Nokia NetAct, SR OS

Ubiquiti EdgeOS

VyOS

The Flexibility of Choice



Network Functional Modules

Building Blocks

command
(e.g. ios_command)

- Executes command on device
- Provides output for further processing

show version
show run

config
(e.g. ios_config)

- Manipulates the config of the device
- Idempotent

ip address ….
hotname …

facts
(e.g. ios_facts)

- Collects facts from the device

Network Functional Module: Config

```
- hosts: network
  gather_facts: no
  connection: local
  tasks:
    - name: configure hostname
      ios_config:
        lines:
          - "hostname {{ inventory_hostname }}"
```

Network Functional Module: Config

First Run:

```
PLAY [network]
*********************************************************************************************

TASK [configure hostname]
*********************************************************************************************
changed: [rtr1]

PLAY RECAP
*********************************************************************************************
rtr1                       : ok=1    changed=1    unreachable=0    failed=0
```

Second Run:

```
PLAY [network]
*********************************************************************************************

TASK [configure hostname]
*********************************************************************************************
ok: [rtr1]

PLAY RECAP
*********************************************************************************************
rtr1                       : ok=1    changed=0    unreachable=0    failed=0
```

Network Functional Module: Facts

```
- hosts: network
  connection: local
  gather_facts: False
  tasks:

    - name: Get facts
      ios_facts:
        gather_subset: all

    - debug: msg="Serial Number is {{ ansible_net_serialnum }}"
```

Network Functional Module: Facts

```
PLAY [network]
*********************************************************************************

TASK [Get facts]
*********************************************************************************
ok: [rtr1]

TASK [debug]
*********************************************************************************
ok: [rtr1] => {
    "msg": "Serial Number is 9G2OX4MKLVP"
}

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

Network Functional Module: Command

```
- hosts: network
  gather_facts: no
  connection: local
  tasks:
    - name: show version
      ios_command:
        commands:
          - show version
        wait_for:
          - result[0] contains Version
        register: results

    - set_fact:
        ver: "{{ results.stdout[0]|regex_search('Version ([0-9.]+)','\\1') }}"

    - debug: var=ver
```

Network Functional Module: Command

```
PLAY [network]
*********************************************************************************************
TASK [show version and show interfaces]
*********************************************************************************************
ok: [rtr1]

TASK [set_fact]
*********************************************************************************************
ok: [rtr1]

TASK [debug] ******************************************************************************
ok: [rtr1] => {
    "ver": [
        "16.06.01"
    ]
}

PLAY RECAP ********************************************************************************
rtr1                       : ok=3    changed=0    unreachable=0    failed=0
```

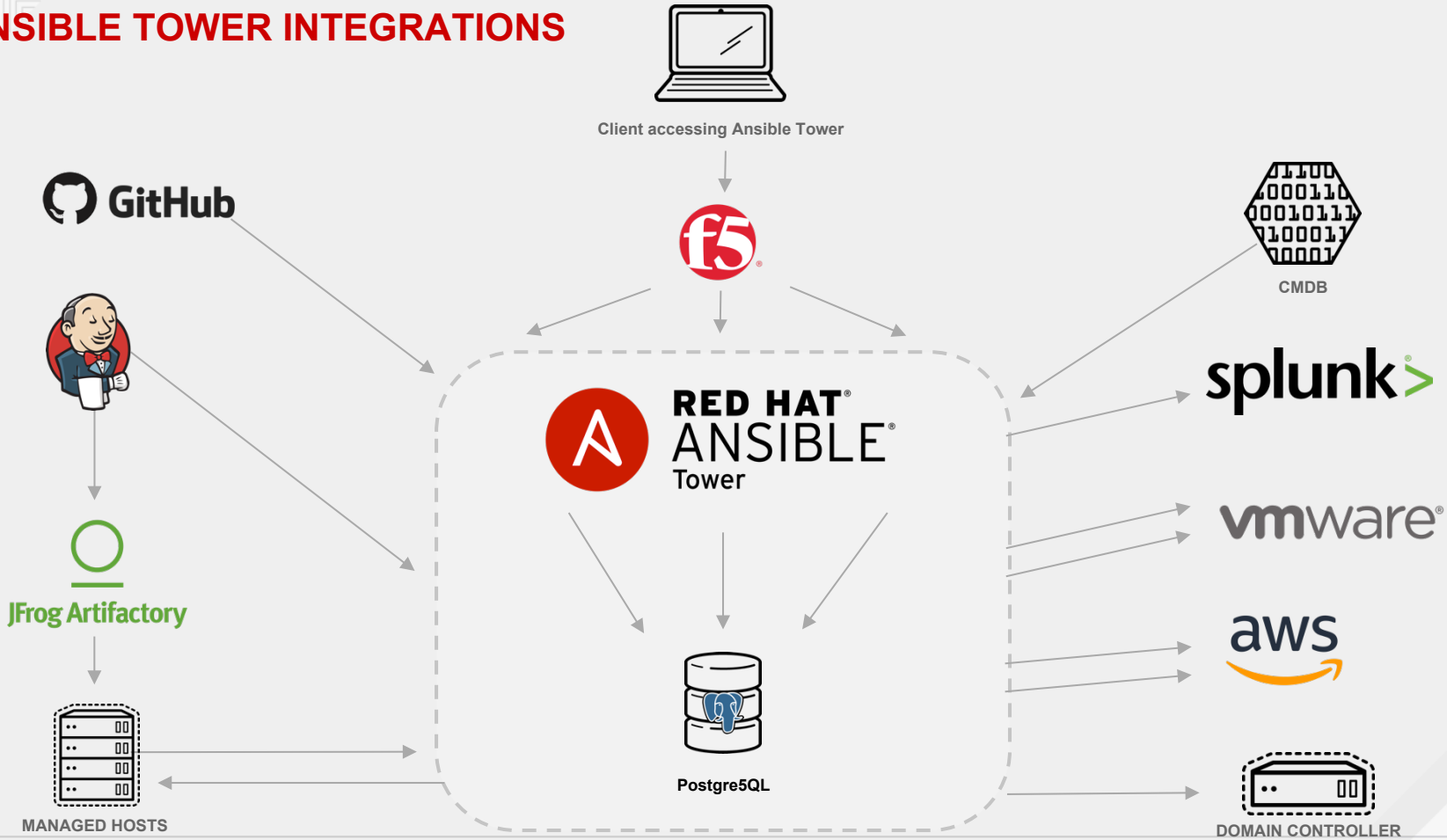

RED HAT®
ANSIBLE®
Automation
RED HAT ANSIBLE TOWER
Scale + operationalize your automation
CONTROL
KNOWLEDGE
DELEGATION
RED HAT ANSIBLE ENGINE
Support for your Ansible automation
SIMPLE
POWERFUL
AGENTLESS
FUELED BY AN INNOVATIVE OPEN SOURCE COMMUNITY

API-Driven Infrastructure

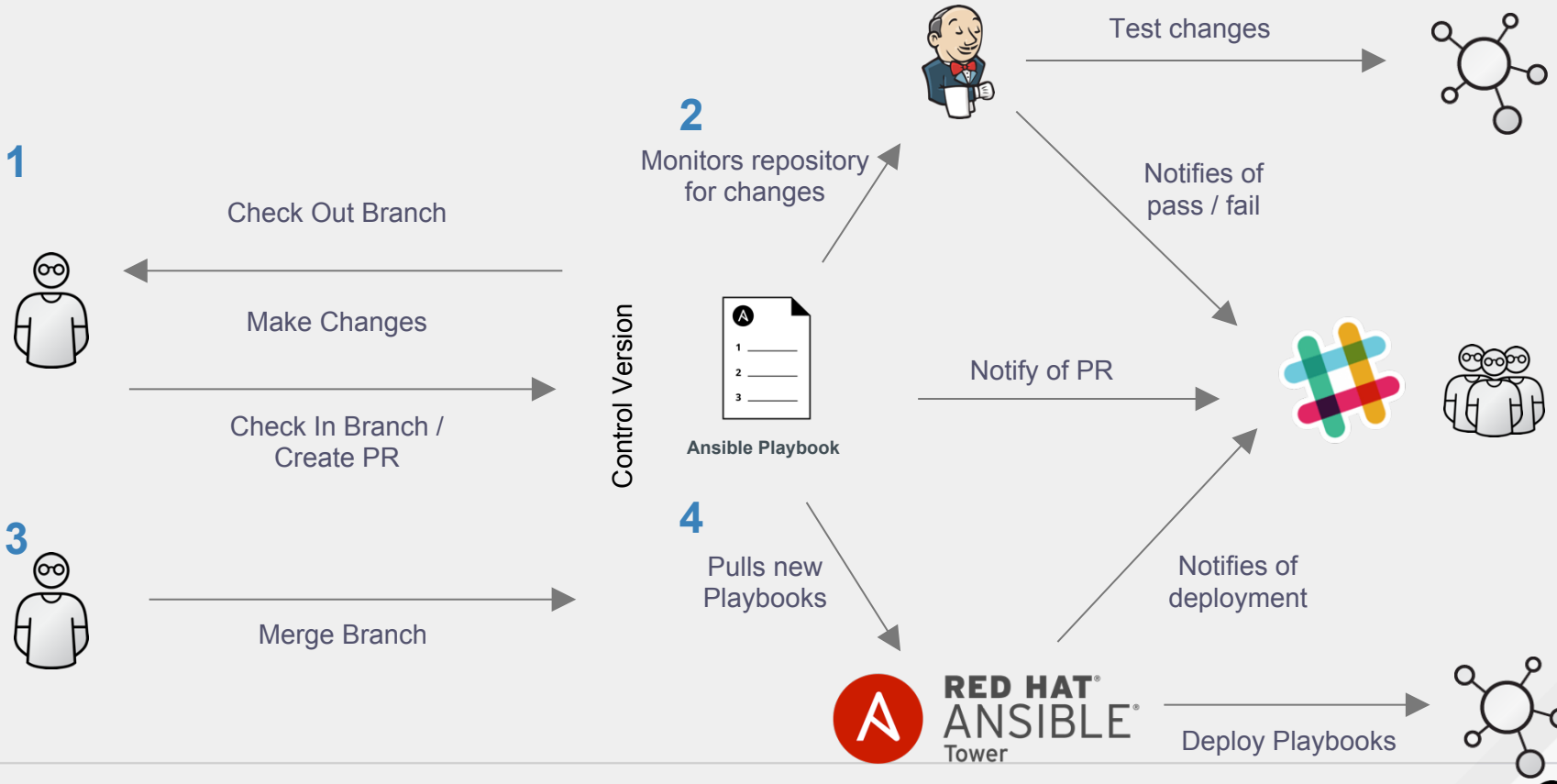
Well Defined, Role Based API

Easily Customizable Back End

ANSIBLE TOWER INTEGRATIONS



Network CI Workflow



redhat.

Use Case Examples

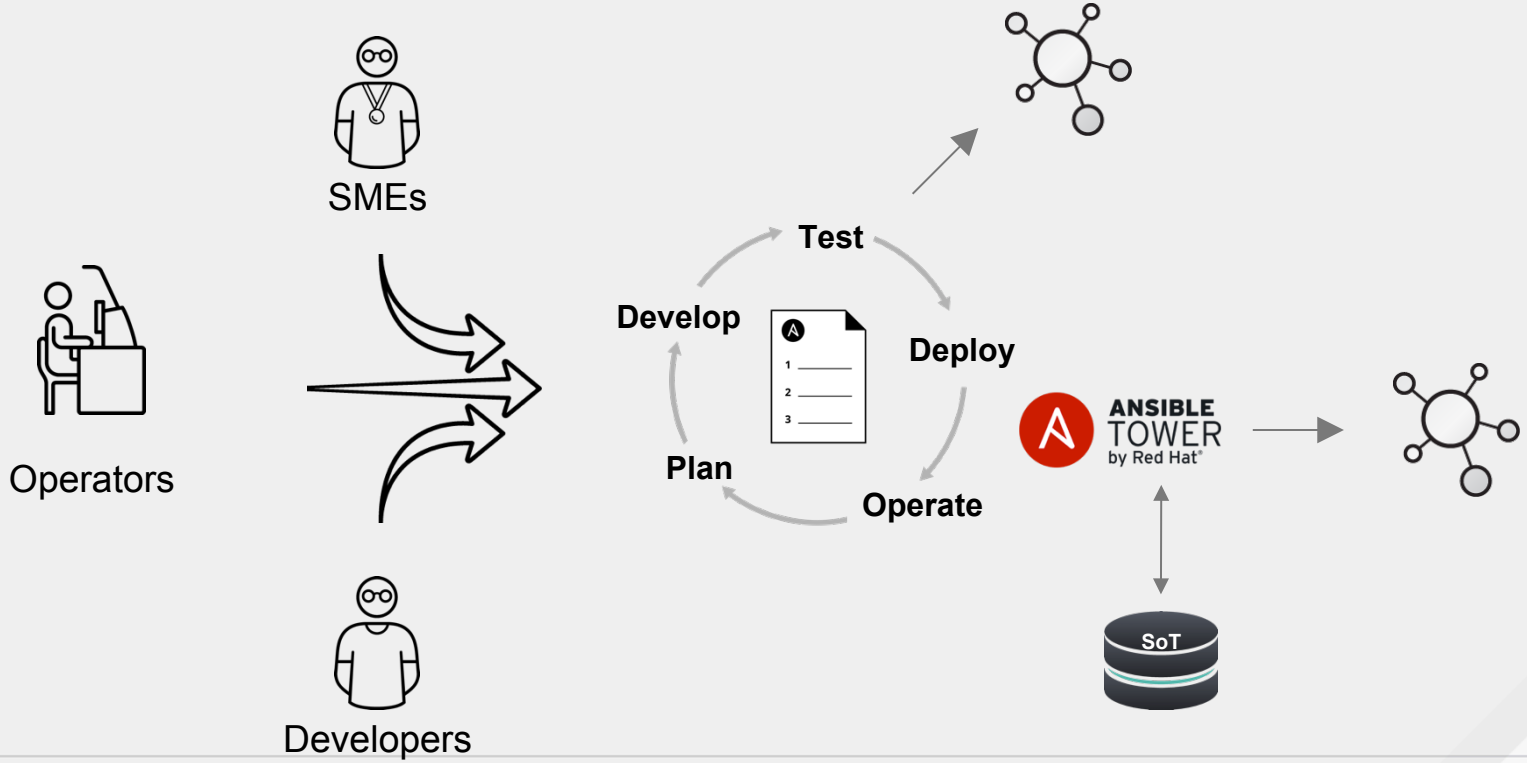
- **Information / Inventory Retrieval and Configuration**
  - Ad hoc or bulk
  - Iteration over specific network segments, VLANs, VRFs
  - Credential management with Tower Vault
- **State Checking and Validation**
  - Compare running configs to desired configs
- **Invocation of Tasks/Playbooks**
  - Manually, API via Tower, Scheduled via Tower

Use Case Examples

- **Continuous Compliance**
  - Combining stateful validation with schedules
  - Logging and Aggregation

- **Integrations**
  - ZTP post-install NOS handoff to Ansible
  - External APIs using Tower-CLI
    - Splunk, ServiceNow, VMware, Elastic
    - Atlassian, GitLab, Jenkins, and most all Red Hat products

The Automated Enterprise



Where Do I Begin?

Learn Ansible

- Join existing Ansible network automation communities
- Take Ansible training courses from Red Hat or elsewhere

Develop success criteria

- Create specific goals that require planning, tailored to your organization
- Create phases to ensure people and processes aren't alienated

Start small!

- Create playbooks that read or check only
- Create simple jobs that eliminate the most annoying tasks
- Leverage existing knowledge internally

Automation Is Not Just a Tool

It's a strategy, it's a journey

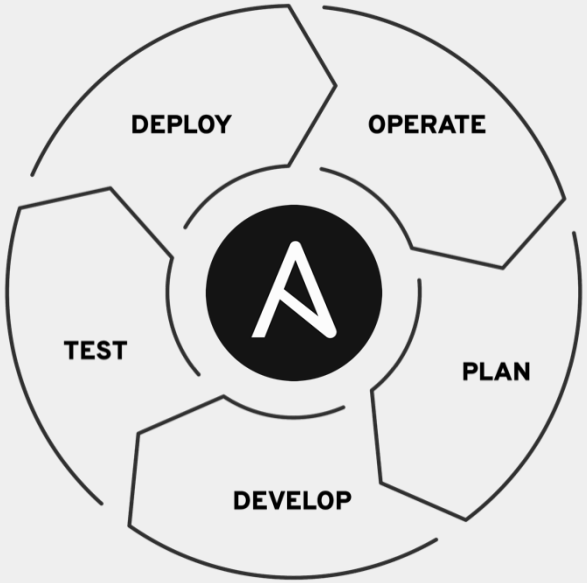
No need to abandon or redefine network operations

- Build with Ansible for bridges between legacy and modern networks

Foster and leverage tribal knowledge

START SMALL, THINK BIG

Three high-level benefits for successful network operations



INFRASTRUCTURE AS YAML

- Automate backup & restores
- Manage “golden” versions of configurations

CONFIGURATION MANAGEMENT

- Changes can be incremental or wholesale
- Make it part of the process: agile, waterfall, etc.

ENSURE AN ONGOING STEADY STATE

- Schedule tasks daily, weekly, or monthly
- Perform regular state checking and validation

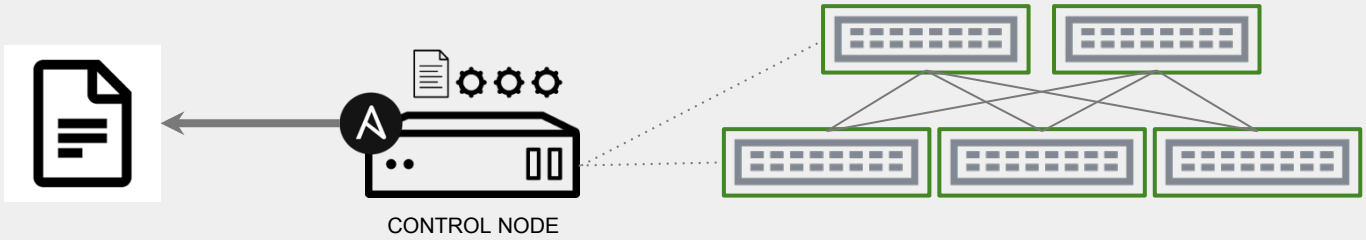
YOUR NETWORK AUTOMATION JOURNEY

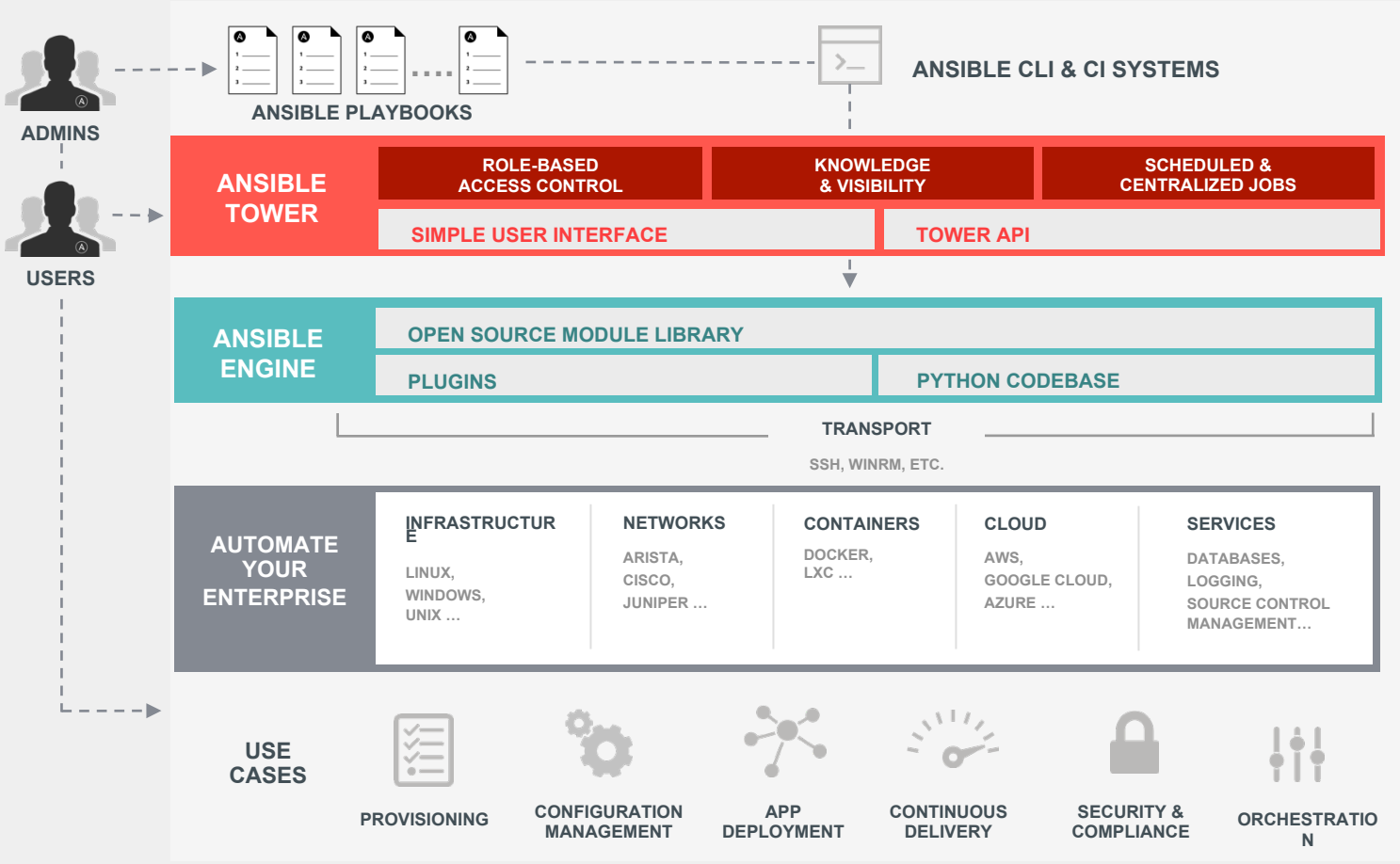
Example Use Cases

BUILD
with Red Hat Ansible Engine

RED HAT® ANSIBLE® Engine

- “Do you know what’s in your network?”
- “What is connected to what?”
- “Which OS versions are installed?”




ADMINS
USERS
ANSIBLE PLAYBOOKS
ANSIBLE CLI & CI SYSTEMS
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
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…
USE CASES
PROVISIONING
CONFIGURATION MANAGEMENT
APP DEPLOYMENT
CONTINUOUS DELIVERY
SECURITY & COMPLIANCE
ORCHESTRATION

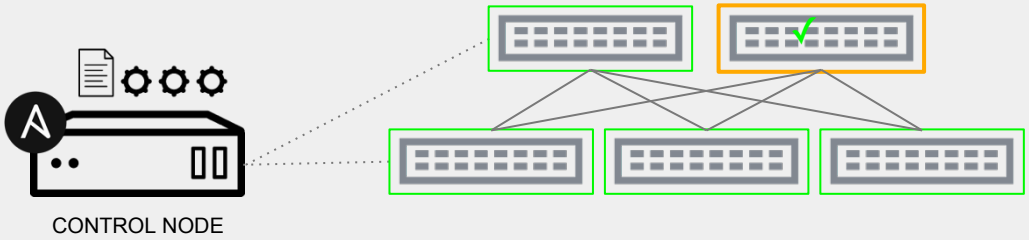
YOUR NETWORK AUTOMATION JOURNEY

Example Use Cases

MANAGE
with Red Hat Ansible Engine

RED HAT® ANSIBLE® Engine

- “Have configurations changed at all?”
- “I need to backup all my network configurations.”
- “I need to bring up and configure a new pod online quickly.”



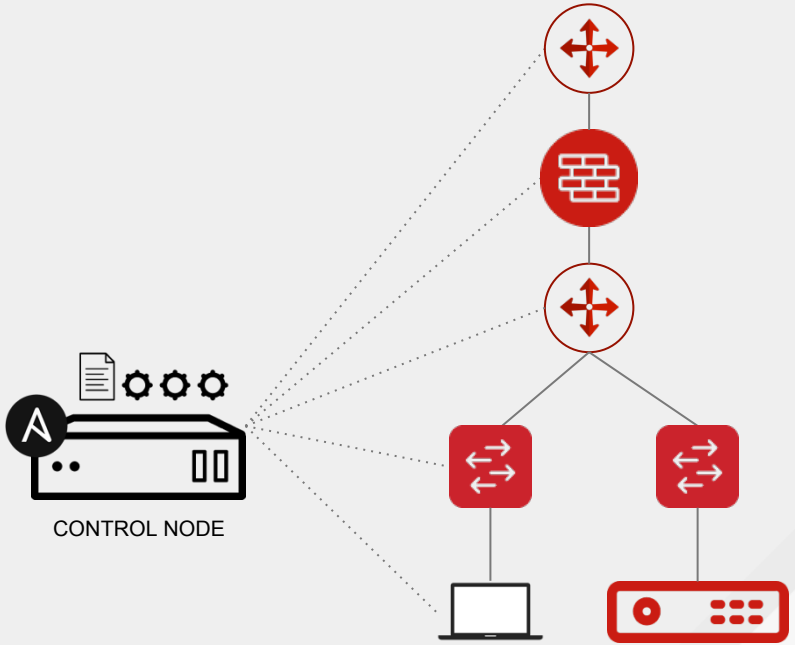
YOUR NETWORK AUTOMATION JOURNEY

Example Use Cases

SCALE
with Red Hat Ansible Tower

RED HAT® ANSIBLE® Tower

- “I need to ensure continuous compliance of configs.”
- “How do large teams operationalize automation globally?”
- “I need to integrate third-party solutions with RESTful API.”



THANK YOU


plus.google.com/+RedHat


facebook.com/redhatinc


linkedin.com/company/red-hat


twitter.com/RedHatNews


youtube.com/user/RedHatVideos

redhat.

RED HAT ANSIBLE ENGINE NETWORKING ADD-ON

redhat. | ANSIBLE

NETWORK MODULES

- **Developed, maintained, tested, and supported** by Red Hat
- **140+ supported modules** and growing*
- Red Hat **reports and fixes problems**
- **Networking modules included** with Ansible Engine offering, but **the Ansible Engine Networking Add-On** SKU purchase is **required** for full support

*take special note of the specific supported platforms

NETWORKING ADD-ON INCLUDED SUPPORT:

Arista EOS

Cisco IOS

Cisco IOS XR

Cisco NX-OS

Juniper Junos

Open vSwitch

VyOS