Red Hat
Ansible Automation
Platform

Network Automation Workshop

Introduction to Ansible for
network engineers and operators

Housekeeping

Understanding the format of this workshop

Topics Covered:

- What is the Ansible Automation Platform?
- What can it do?
- Why Network Automation?
- How Ansible Network Automation works

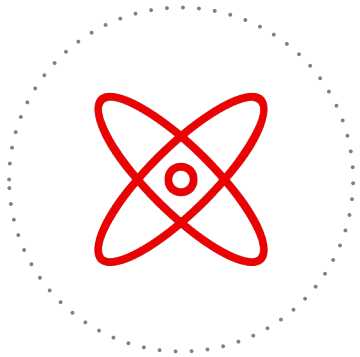
Housekeeping

Understanding the format of this workshop

Timing

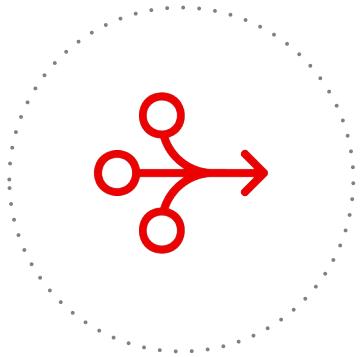
- Review Concepts (~25 min)
- AAP Overview Demo (~10 min)
- Labs with instructors in breakout rooms (~85 min)
- Labs available for 24 hours

Why the Ansible Automation Platform?



Powerful

Orchestrate complex processes at enterprise scale.



Simple

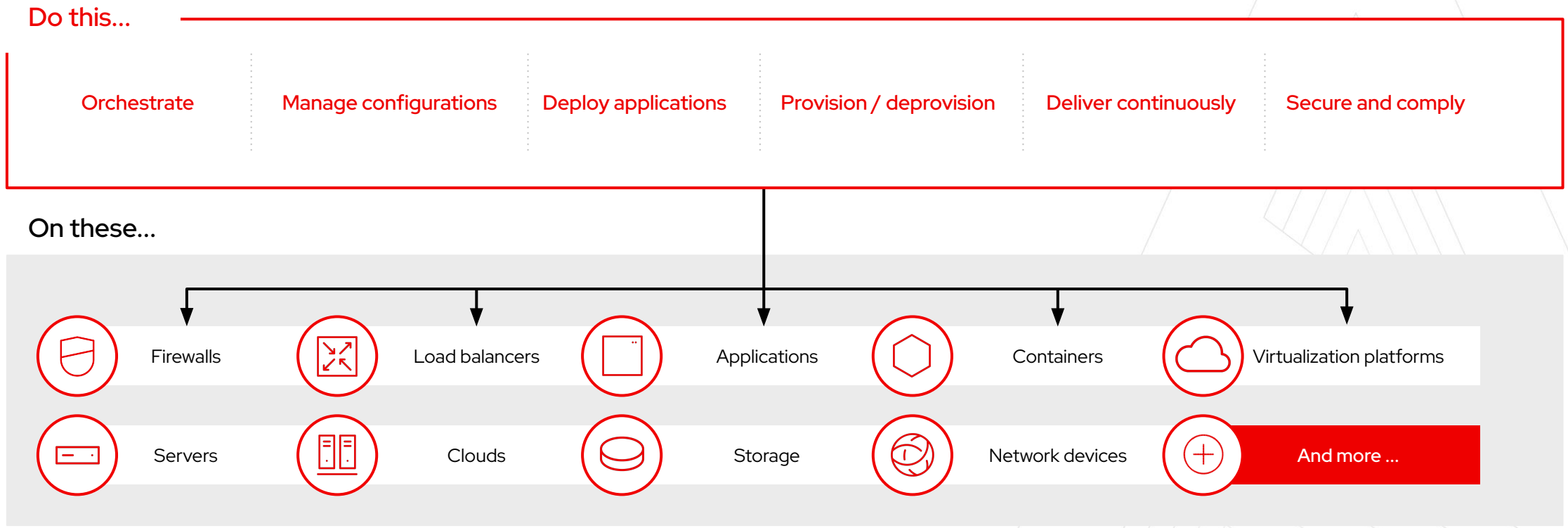
Simplify automation creation and management across multiple domains.

Agentless

Easily integrate with hybrid environments.

Automate the deployment and management of automation

Your entire IT footprint



Red Hat
Ansible Automation
Platform

Supported and certified content you can trust.

140+

Certified Content Collections

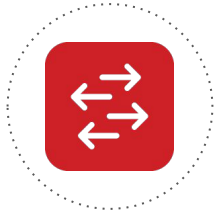
55+

Certified technology partners

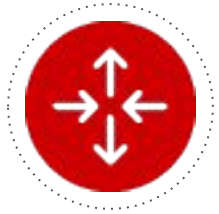
Infrastructure | Cloud | Network | Security | Edge

Red Hat Ansible Automation Platform

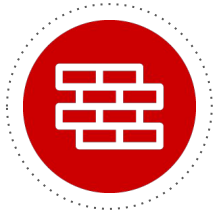
Ansible Network Ecosystem



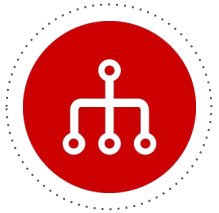
SWITCHES



ROUTERS



ENTERPRISE FIREWALLS



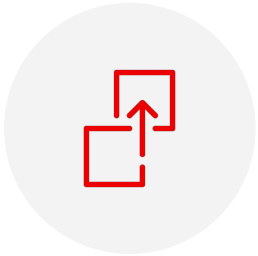
LOAD BALANCERS

CONTROLLERS

IP ADDRESS MGMT

Next generation networking

Automation to effectively manage increasing diversity and scope



Edge / IoT Devices

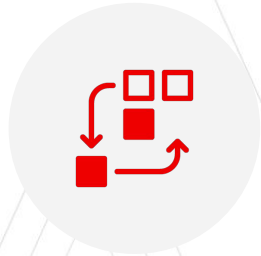
New device types entering networks at scale, with distributed computing.

Hybrid cloud

Numerous deployment forms across the globe

Digital transformation

Responding with new applications is only as fast as the slowest process



Data-intensive computing

Artificial intelligence, digital applications and growing data driving connectivity

Network Automation Journey

Complexity

OPPORTUNISTIC

How can we simplify a task or set of tasks?

- Backup & Restore
- Dynamic Documentation

SYSTEMATIC

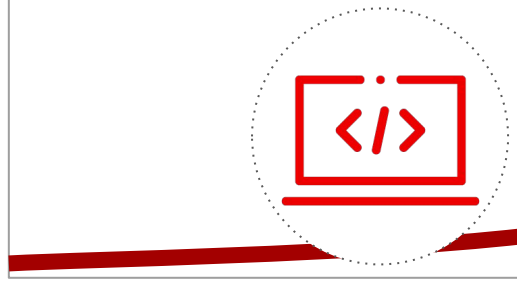
How do we centralise our processes?

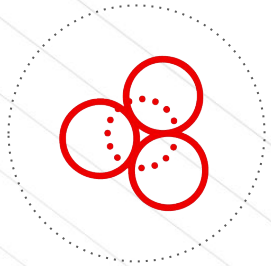
- Scoped Config Management
- Network Compliance

INSTITUTIONALIZED

How do we orchestrate our processes?

- Operational State Validation
- Automated NetOps



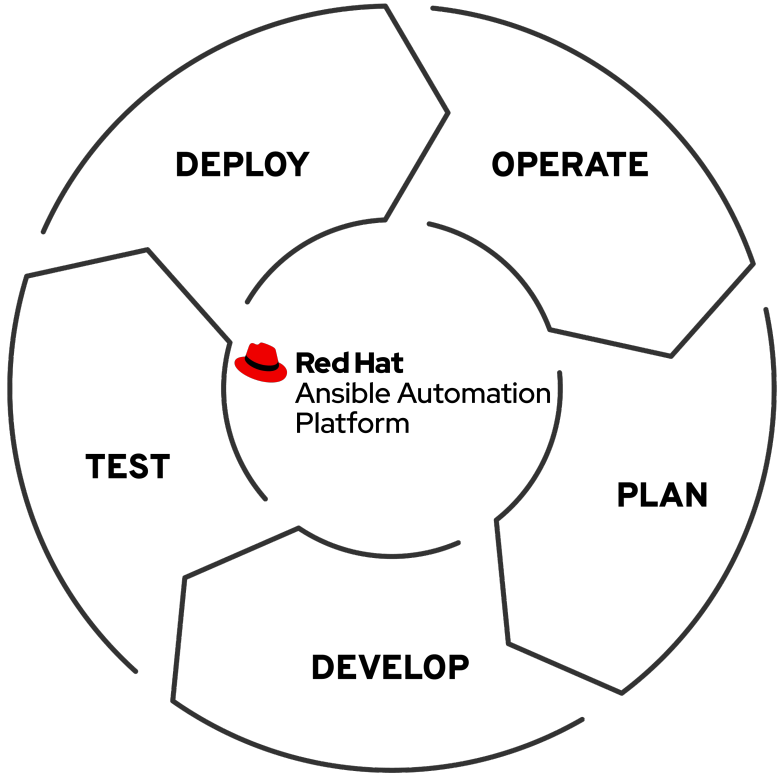


Ansible Network Automation

Red Hat
Ansible Automation
Platform

What does it do?

Three high-level benefits for successful network operations



Infrastructure Awareness

- Dynamic Documentation
- Compliance and traceability

Configuration Management

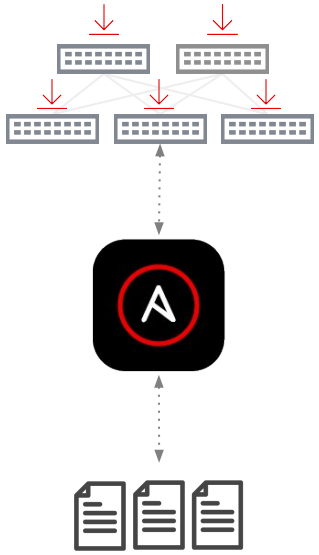
- Automate backup & restores
- Scoped Config Management

Network Validation

- Validate operational steady-state
- Roll back if configuration changes don’t meet goals

Start Small

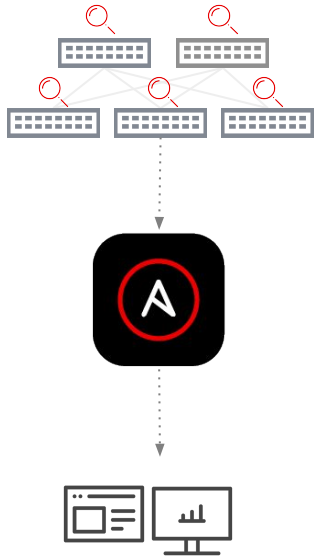
Quick automation victories for network engineers



Config Backup and Restore

Ubiquitous first touch use case

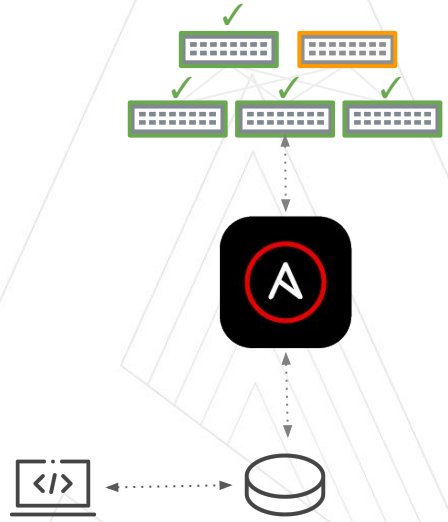
- Gain confidence in automation quickly
- First steps towards network as code
- Quickly recover network steady state



Dynamic Documentation

Use Ansible facts to gain information

- Read-only, no production config change
- Dynamic Documentation and reporting
- Understand your network



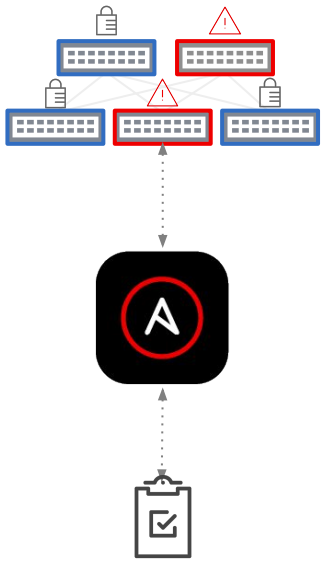
Scoped Config Management

Focus on high yield victories

- Automate VLANs, ACLs and SNMP config
- Introduce source of truth concepts
- Enforce Configuration policy

Think Big

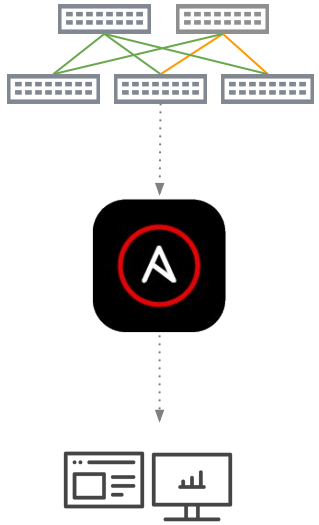
Institutionalizing automation into your organization



Network Compliance

Respond quickly and consistently

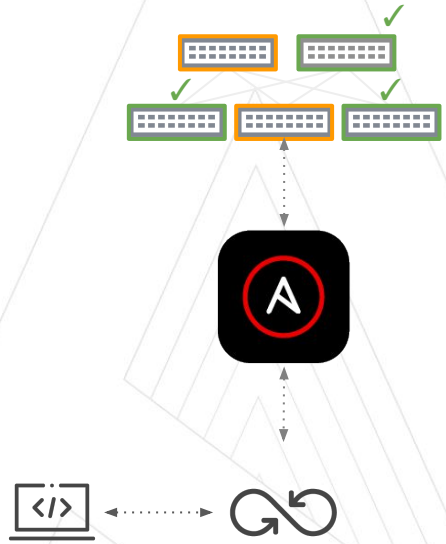
- Security and config compliance for network
- Remove human error from security responses
- Enforce Configuration policies and hardening



Operational State Validation

Going beyond config management

- Parsing operational state to structured values
- Schema validation and verification
- Enhance operational workflows

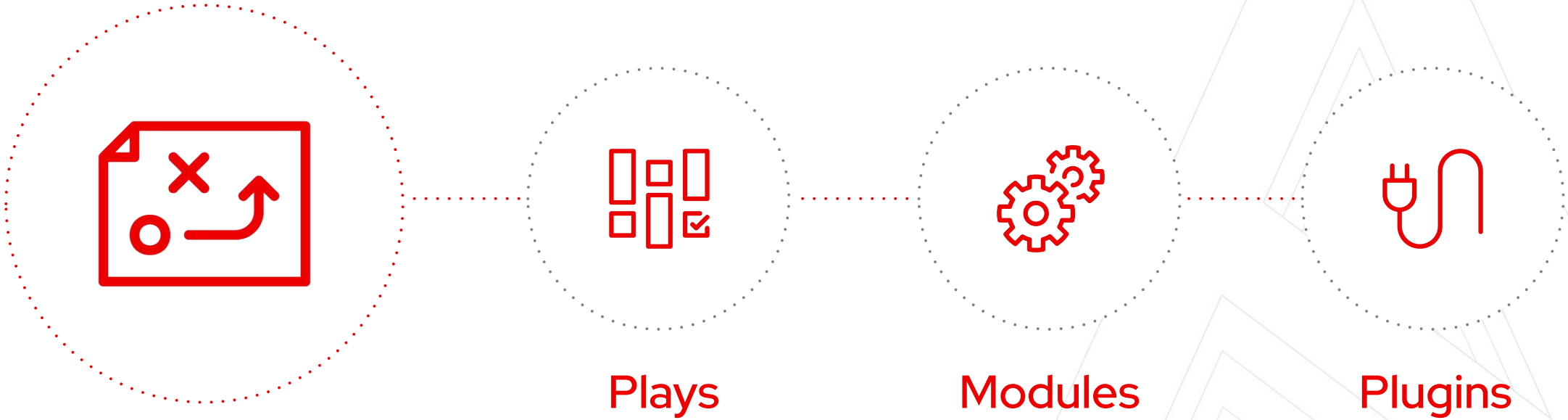


Automated NetOps

Infrastructure as code

- Data centric automation
- Deploy configuration pipelines
- GitOps for Network Automation

What makes up an Ansible playbook?



Ansible plays

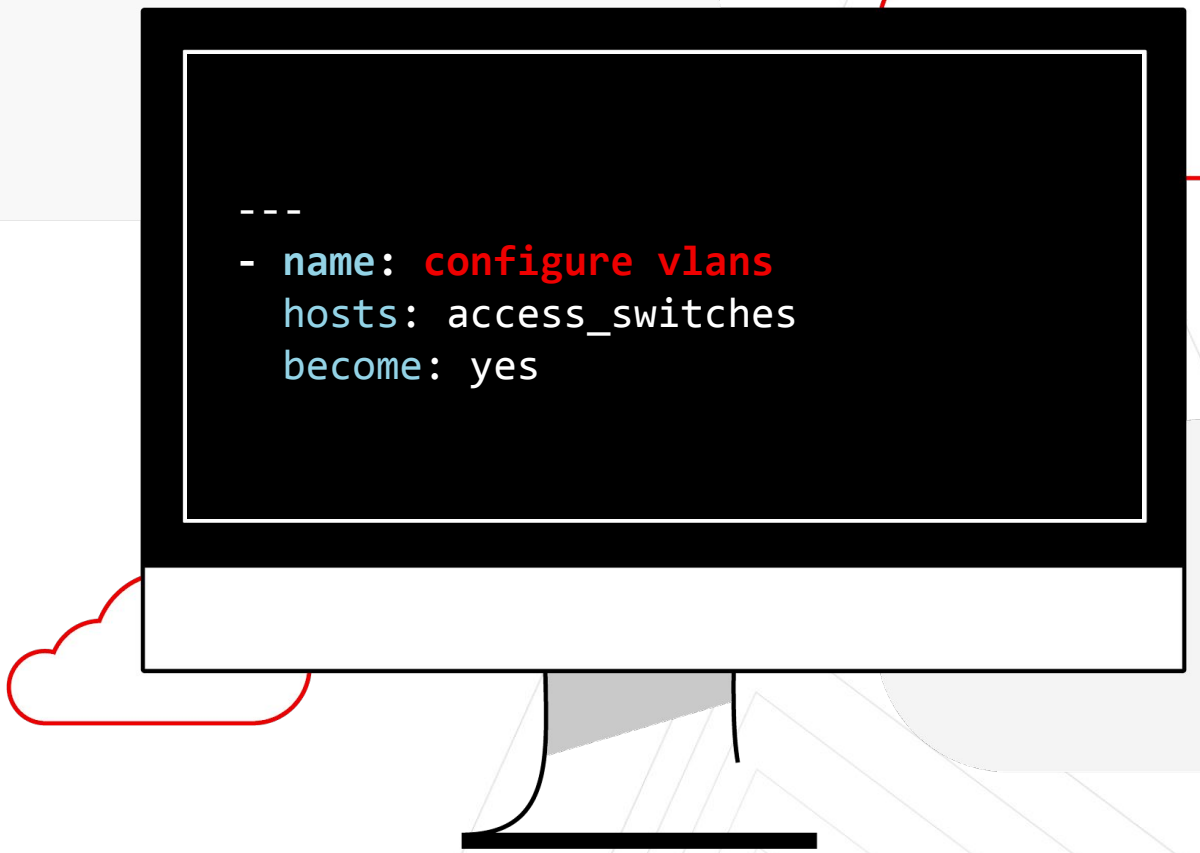
What am I automating?

What are they?

Top level specification for a group of tasks. Will tell that play which hosts it will execute on and control behavior such as fact gathering or privilege level.

Building blocks for playbooks

Multiple plays can exist within an Ansible playbook



Ansible modules

The “tools in the toolkit”

What are they?

Parametrized components with internal logic, representing a single step to be done.
The modules “do” things in Ansible.

Language

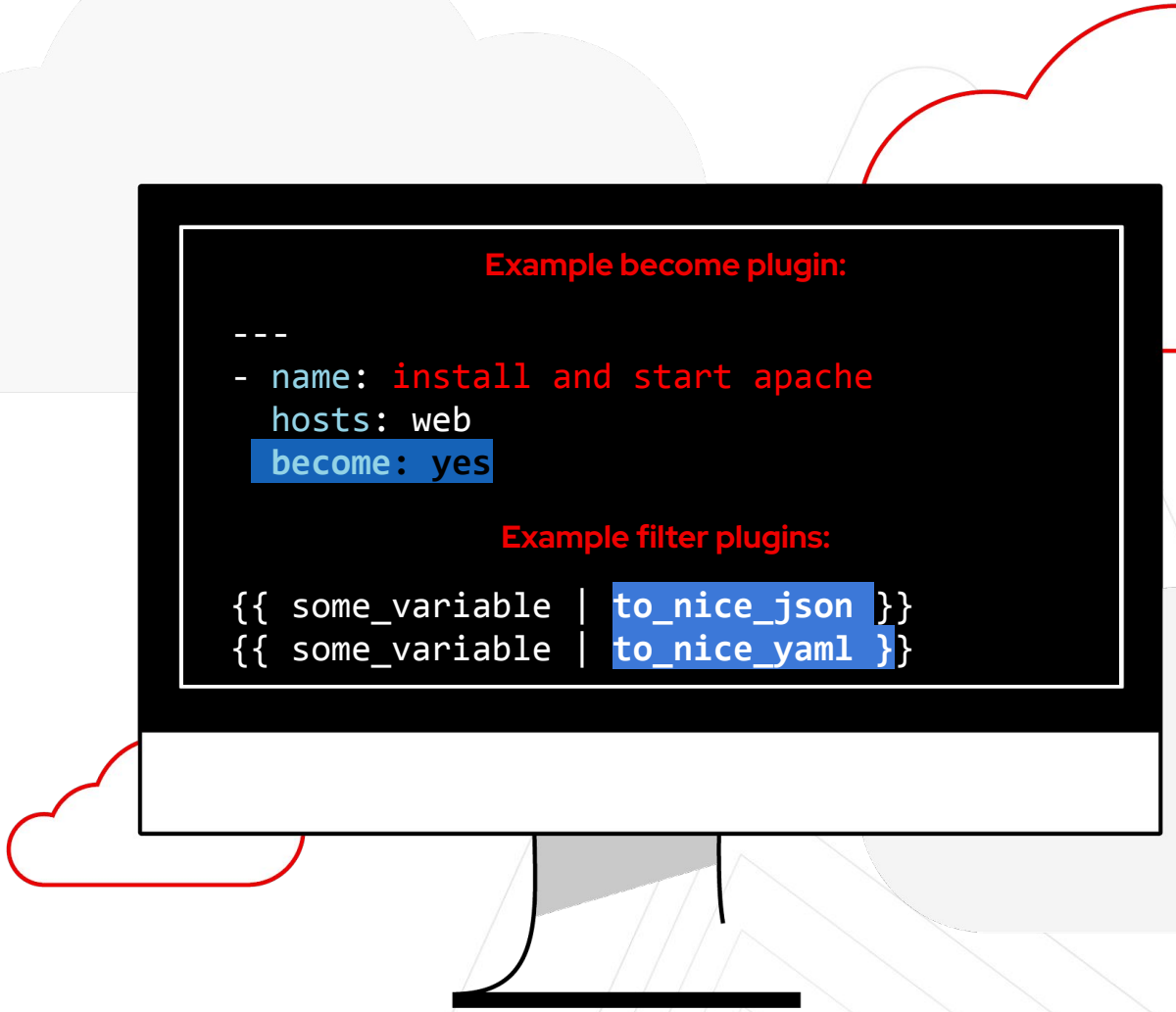
Usually Python, or Powershell for Windows setups. But can be of any language.

Ansible plugins

The “extra bits”

What are they?

Plugins are pieces of code that augment Ansible’s core functionality. Ansible uses a plugin architecture to enable a rich, flexible, and expandable feature set.



Ansible Inventory

The systems that a playbook runs against

What are they?

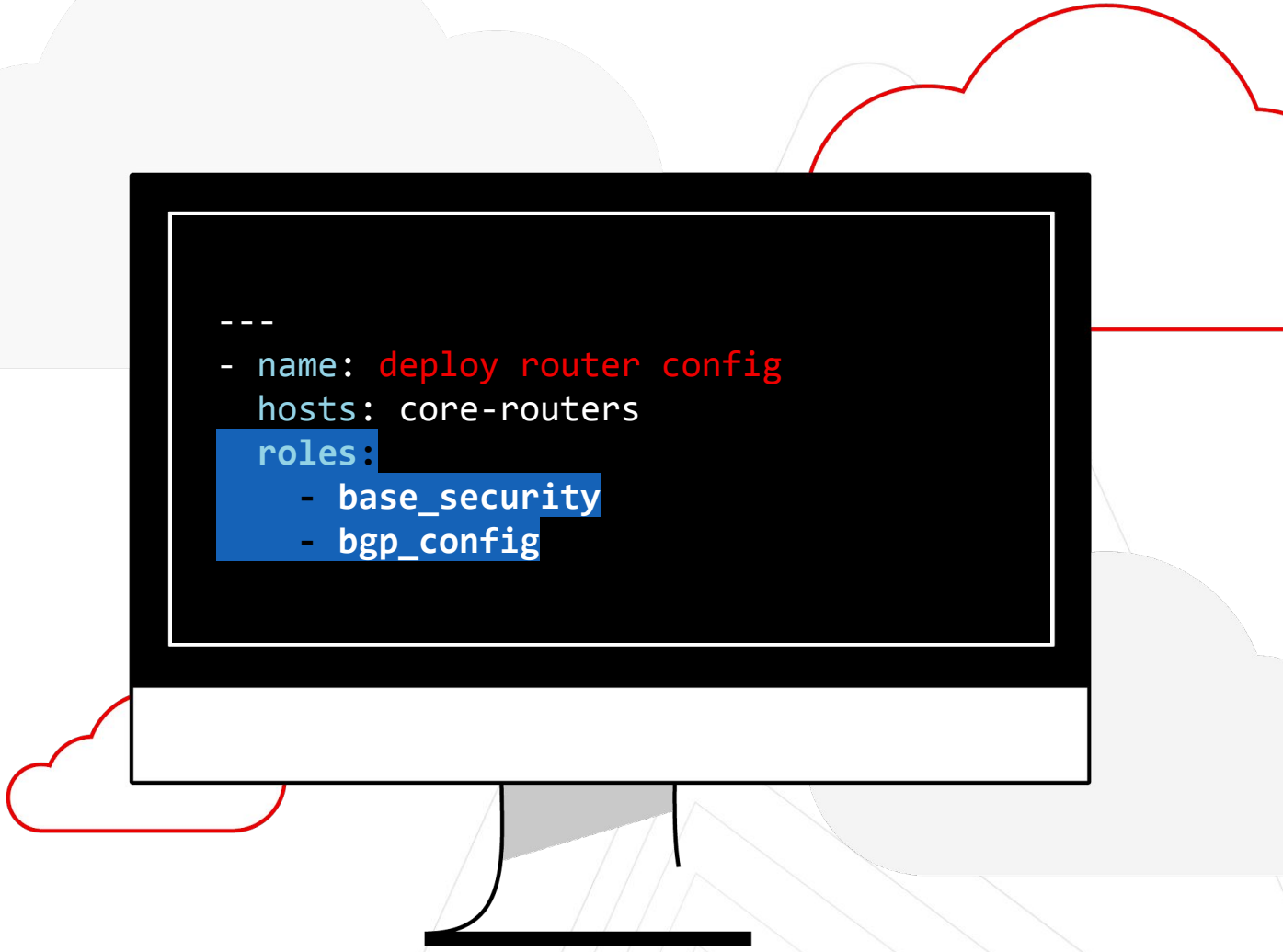
List of systems in your infrastructure that automation is executed against

Ansible roles

Reusable automation actions

What are they?

Group your tasks and variables of your automation in a reusable structure. Write roles once, and share them with others who have similar challenges in front of them.



Collections

Simplified and consistent content delivery

What are they?

Collections are a directory structure containing automation content:

- Modules
- Playbooks
- Roles
- Plugins
- Docs
- Tests

Network Connection Plugins

Use your vendor connection of choice

`ansible_connection`

- **`netconf`** – XML over netconf

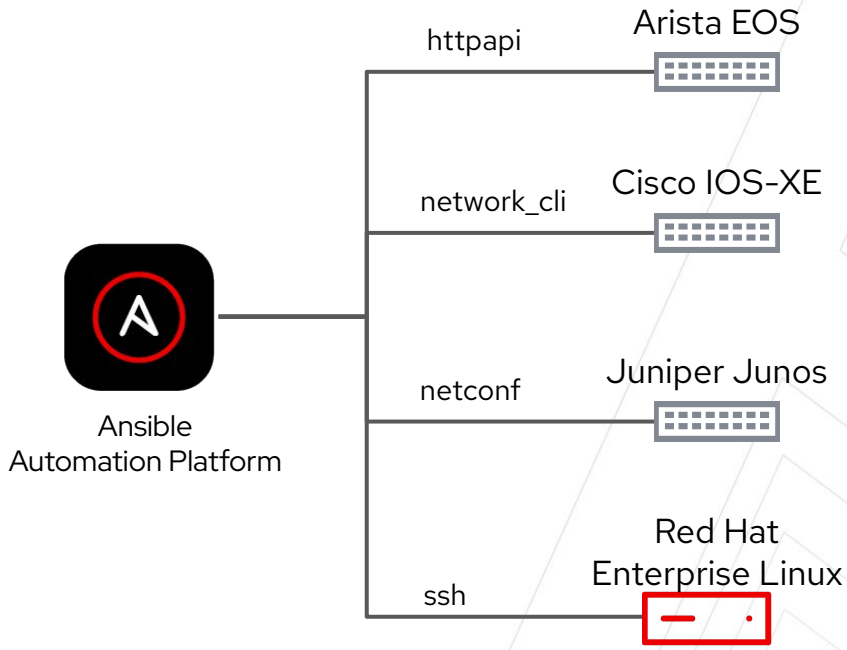
  example: Juniper Junos

- **`network_cli`** – command line over SSH

  example: Cisco IOS-XE, Arista EOS

- **`httpapi`** – vendor API

  example: Arista eAPI, Cisco NX-API



https://docs.ansible.com/ansible/latest/plugins/connection.html

Network modules

Ansible modules for network automation typically references the vendor OS followed by the module name.

- namespace.collection.facts
- namespace.collection.command
- namespace.collection.config
- namespace.collection.resource

More modules depending on platform

Arista EOS = arista.eos.

Cisco IOS/IOS-XE = cisco.ios

Cisco NX-OS = cisco.nxos

Cisco IOS-XR = cisco.iosxr

F5 BIG-IP = f5networks.f5_bigip_bigip.

Juniper Junos = junipsnetworks.junos.

VyOS = vyos.vyos.

Network Automation Modules

How do we interact with network devices?

command

namespace.collection.**command**
Cisco IOS -> `cisco.ios.command`

facts

namespace.collection.**facts**
Arista EOS -> `arista.eos.facts`

config

namespace.collection.**config**
Juniper Junos-> `junipernetworks.junos.config`

resource

namespace.collection.**module**
Cisco IOS-XR-> cisco.iosxr.acls

Red Hat
Ansible Automation
Platform

What are facts?

Structured data, the Ansible way

```
cisco# show version
Cisco IOS XE Software, Version 16.09.02
Cisco IOS Software [Fuji], Virtual XE Software
(X86_64_LINUX_IOSD-UNIVERSALK9-M), Version 16.9.2,
RELEASE SOFTWARE (fc4)
Technical Support: http://www.cisco.com/techsupport
Copyright (c) 1986-2018 by Cisco Systems, Inc.



<<rest of output removed for slide brevity>>
```

Cisco IOS output

```
cisco# ansible -m ios_facts cisco
cisco | SUCCESS => {
    "ansible_facts": {
        "ansible_net_iostype": "IOS-XE",
        "ansible_net_version": "16.09.02",
        "ansible_net_serialnum": "9L8KQ482JFZ",
        "ansible_net_model": "CSR1000V",



<<rest of output removed for slide brevity>>
```

Ansible output

Red Hat
Ansible Automation
Platform

Working with Ansible facts

1. Gather facts

```
- name: gather eos facts
  arista.eos.facts:
    gather_subset: config
    gather_network_resources: vlans
```

2. Use facts

```
- name: print out vlans
  debug:
    var: ansible_network_resources.vlans
```

```
- name: display version
  debug:
    msg: "The IOS version is: {{ ansible_net_version }}"

- name: display serial number
  debug:
    msg: "The serial number is: {{ ansible_net_serialnum }}"
```

Red Hat
Ansible Automation
Platform

Build reports with Ansible Facts

Hostname	Model Type	Mgmt0 IP Address	Code Version
n9k	Nexus9000 9000v Chassis	192.168.2.3	7.0(3)I7(1)
n9k2	Nexus9000 9000v Chassis	192.168.2.4	7.0(3)I7(1)
n9k3	Nexus9000 9000v Chassis	192.168.2.5	7.0(3)I7(1)
n9k4	Nexus9000 9000v Chassis	192.168.2.6	7.0(2)I7(1)
n9k5	Nexus9000 9000v Chassis	192.168.2.7	7.0(3)I7(1)
n9k6	Nexus9000 9000v Chassis	192.168.2.8	7.0(3)I7(1)

A Sample Ansible Playbook

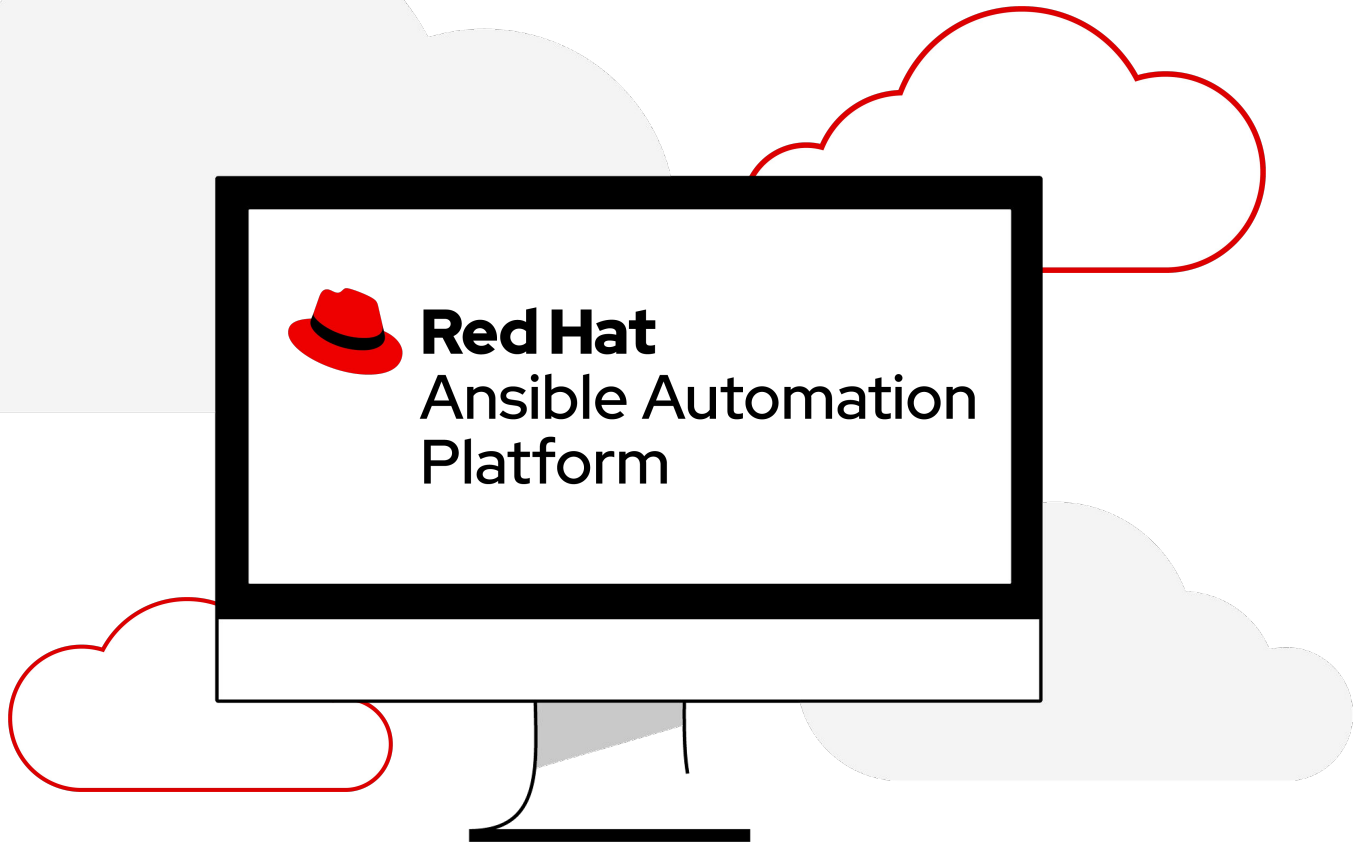
```
---
- name: configure VLANs
  hosts: cisco
  gather_facts: false
  tasks:
    - name: VLANs task
      cisco.nxos.vlans:
        config:
        - vlan_id: 5
          name: WEB
        - vlan_id: 10
```

- A playbook is a list of plays.
- Each play is a list of tasks.
- Tasks invoke modules.
- A playbook can contain more than one play.

Red Hat
Ansible Automation
Platform

What's next?

- Quick demo
- Labs for remainder of workshop



Where to go next

Learn more

- Workshops
- Documents
- Meetup
- Youtube
- Twitter

Get started

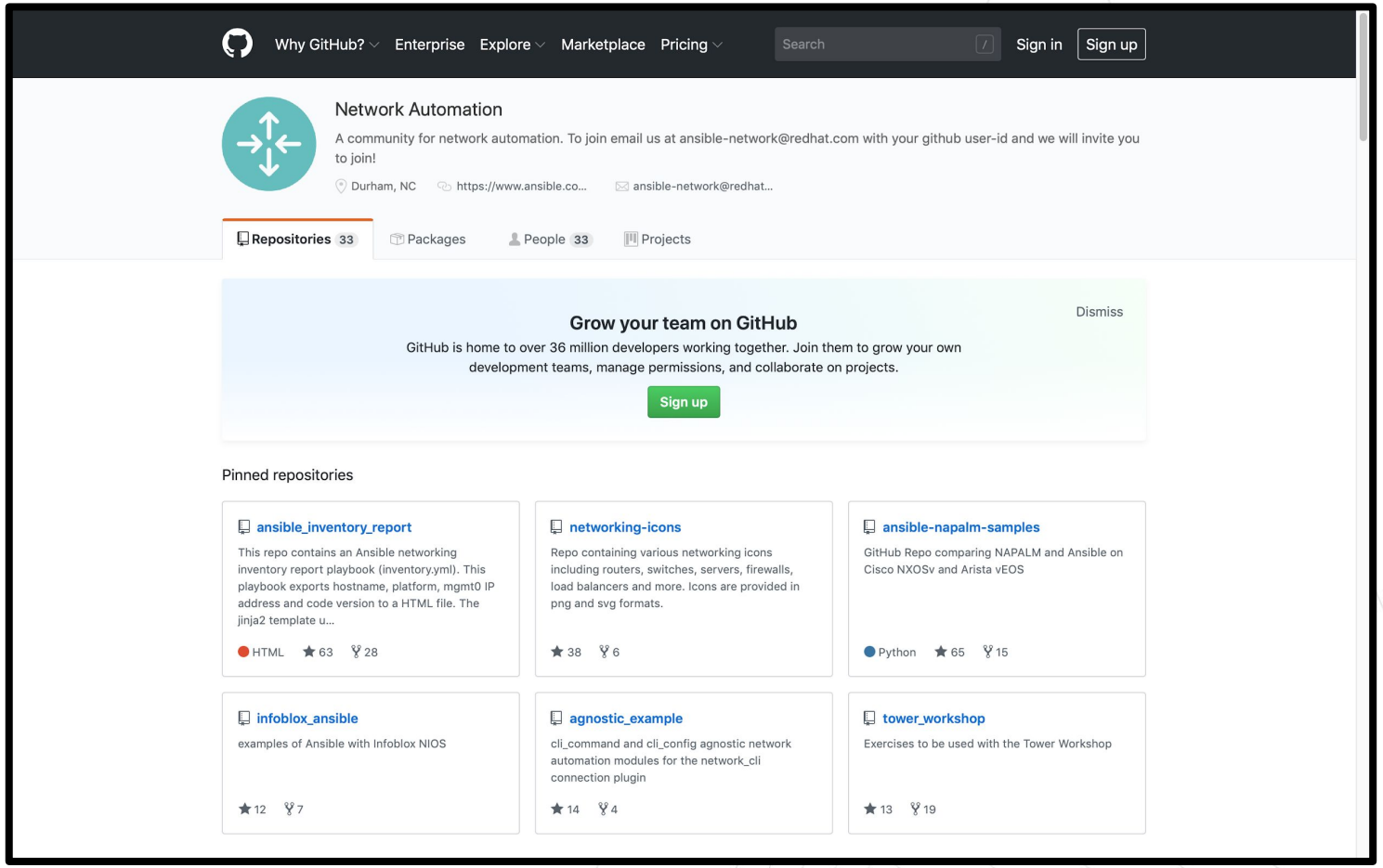
- Evals
- cloud.redhat.com

Get serious

- Red Hat Automation Adoption Journey
- Red Hat Training
- Red Hat Consulting

Bookmark the Github organization

- Examples, samples and demos
- Run network topologies right on your laptop



Red Hat
Ansible Automation Platform

Thank you


linkedin.com/company/red-hat


youtube.com/AnsibleAutomation


facebook.com/ansibleautomation


twitter.com/ansible


github.com/ansible

Red Hat