Red Hat
Ansible Automation
Platform

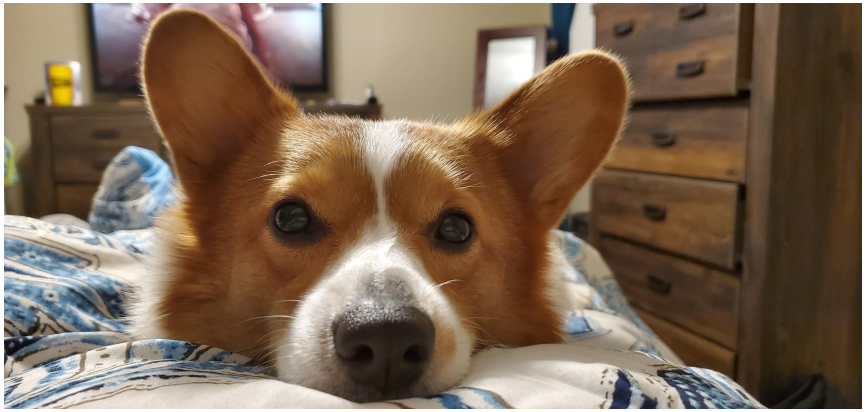
End-to-End Enterprise Automation

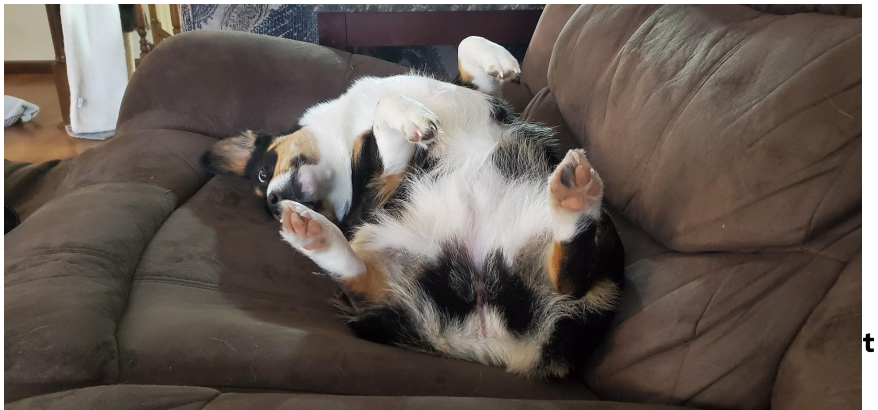
Building, Publishing, and Delivering Ansible Collections

Josh Swanson
Chief Architect
Red Hat
joshswanson@redhat.com

Josh Swanson

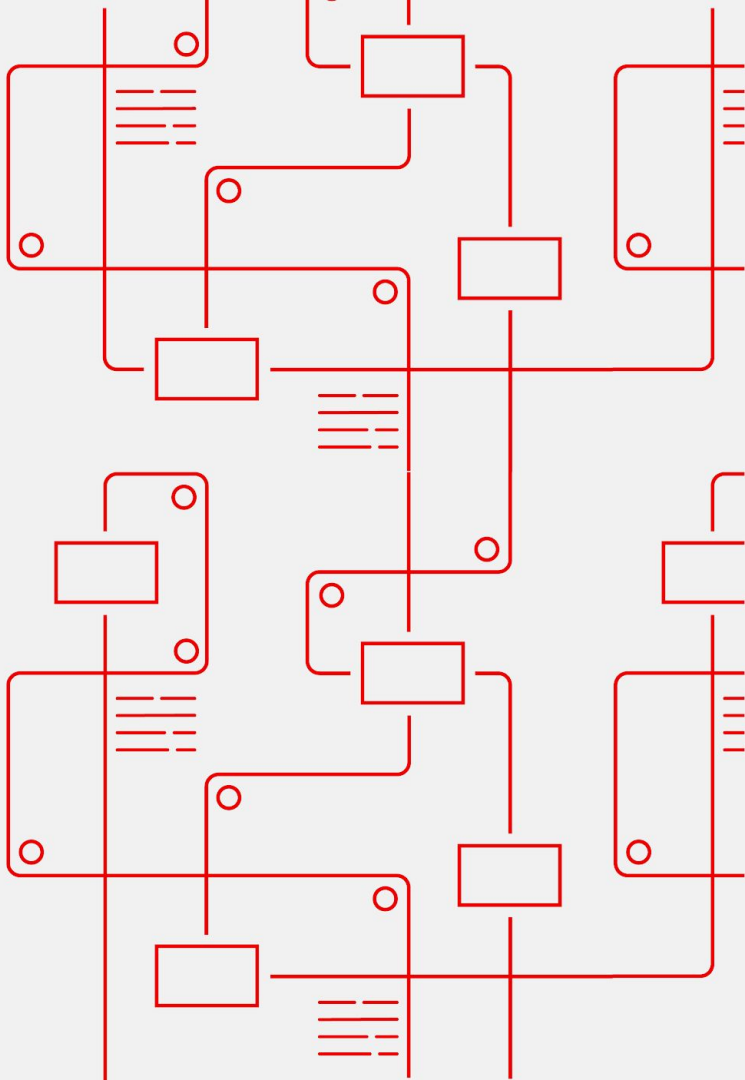
joshswanson@redhat.com



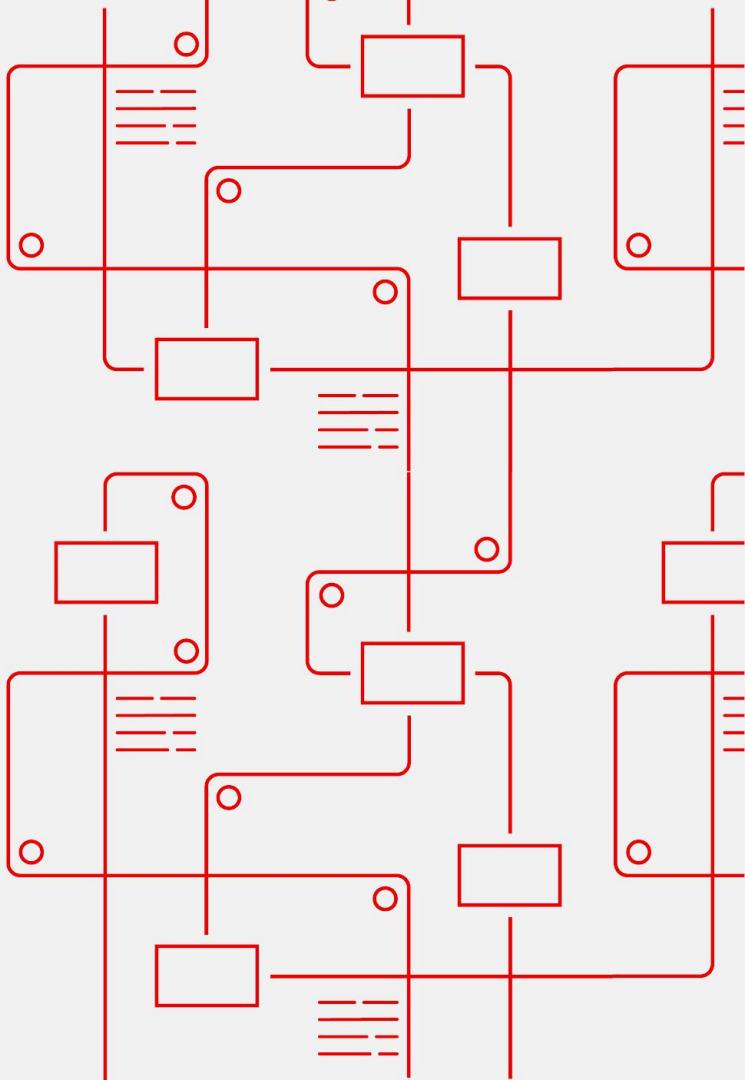


Agenda

- **Why Standardize on a Platform?**
  The story of Red Hat Enterprises
- **Ansible Content Flow**
  Building, publishing, and delivering our automation
- **Quick Review**
  The new state of Red Hat Enterprises
- **Q&A**

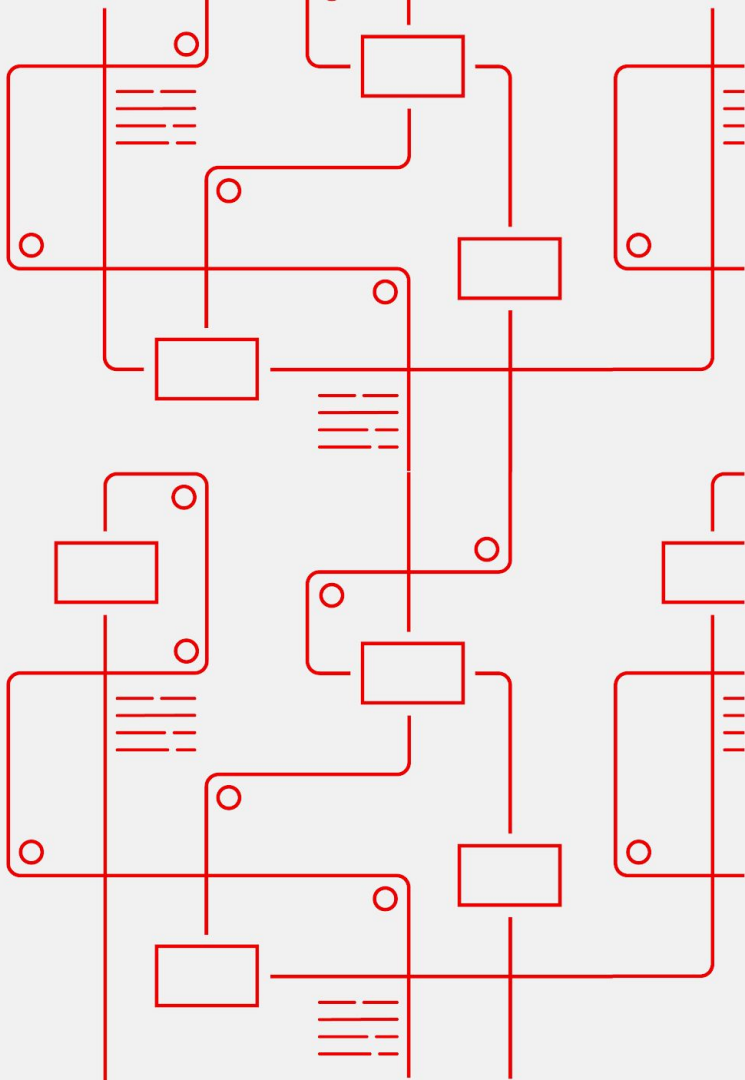


Why Standardize on a Platform?



Red Hat Enterprises

A Startup Hotel Company



Red Hat Enterprises Resource Request

Lines of Business

“We’re entering a new market space and need to quickly deploy an application to capture market share. The faster this is deployed, the more market share we’ll capture, and the more revenue we’ll drive for the business.”

Josh Swanson

Project Manager

Red Hat Enterprises Entering a New Market

Red Hat Enterprises Resource Request

Initiating the Process

Lines of Business

“We’ll need IT assets for integrating with the airline booking systems and for automatically adding reservations to our existing booking systems. We also want to increase the amount of customer information we send to our data warehouse.”

Josh Swanson

Project Manager

Red Hat Enterprises Resource Request

Reviewing the Last Time this Process was Run

Josh Swanson

Project Manager

- 5 total teams involved:
  - CMDB
  - Network
  - Virtualization
  - RHEL Admin/Ops
  - Security
- 6 tickets to those teams in ServiceNow
  - Average closure time of 2 business days
  - Total time: 10 days
- Another 5 days for app deployment
  - Bit of re-work needed
  - Have to be careful: re-requesting resources would take another ~10 business days
- The business wants this done by tomorrow EOD...

Red Hat Enterprises Key Stakeholders

It takes a village to provision infrastructure

Network

Zach Peterson

Operating System

Pete Scurek

Security

Scott Danielson

CMDB

Scott Peters

Virtualization

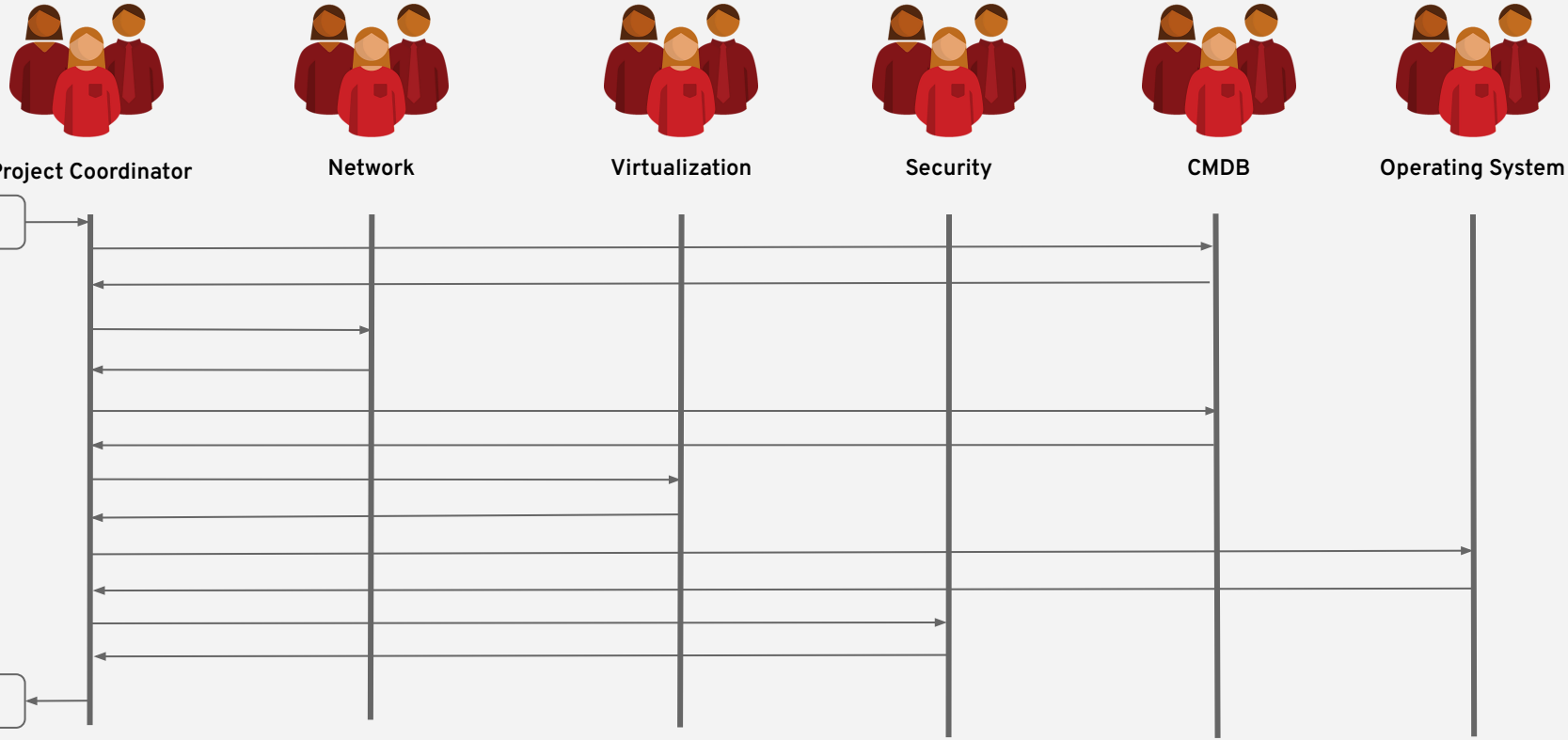
Jimmy Connor

Project Manager

Josh Swanson

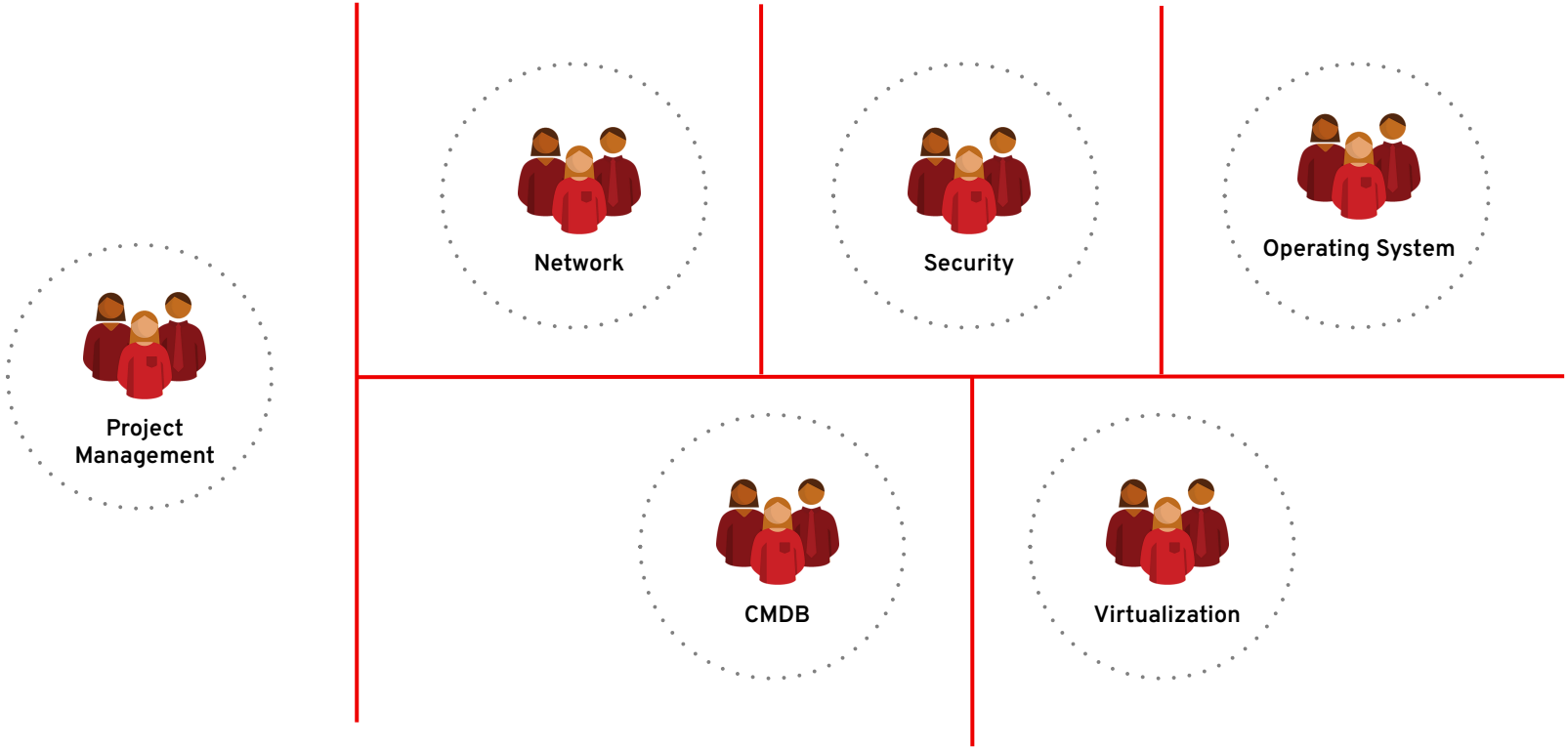
Red Hat Enterprises Provisioning Workflow

It takes a village to provision infrastructure



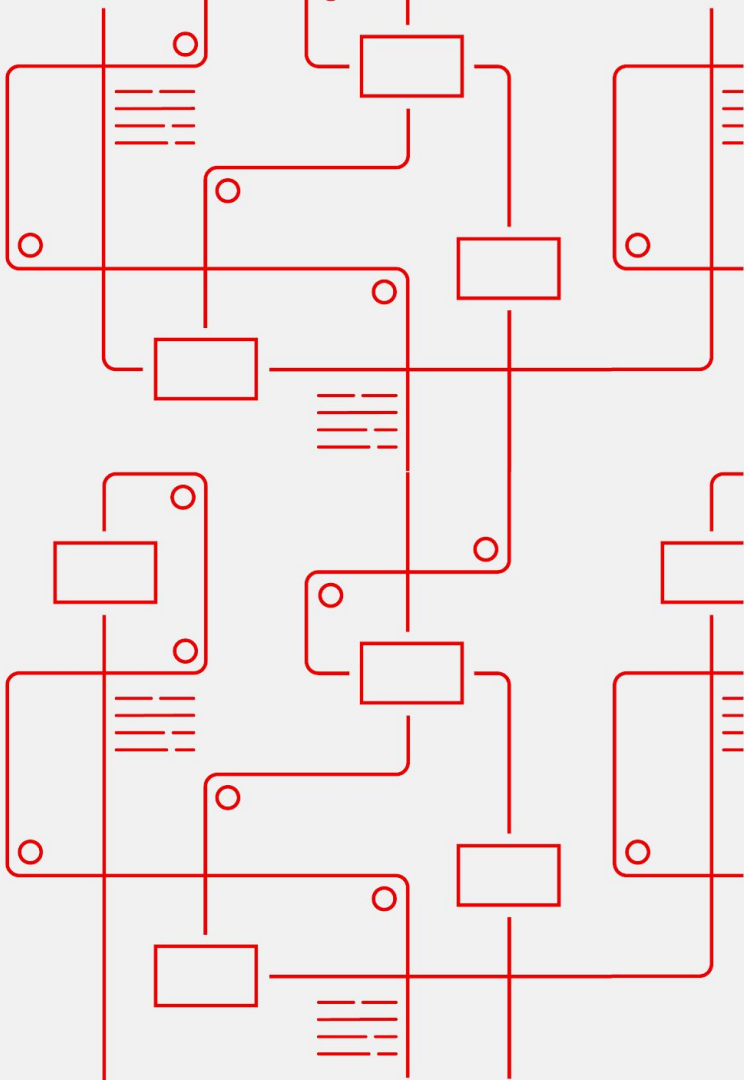
Red Hat Enterprises Key Stakeholders

Here's What We Have Today

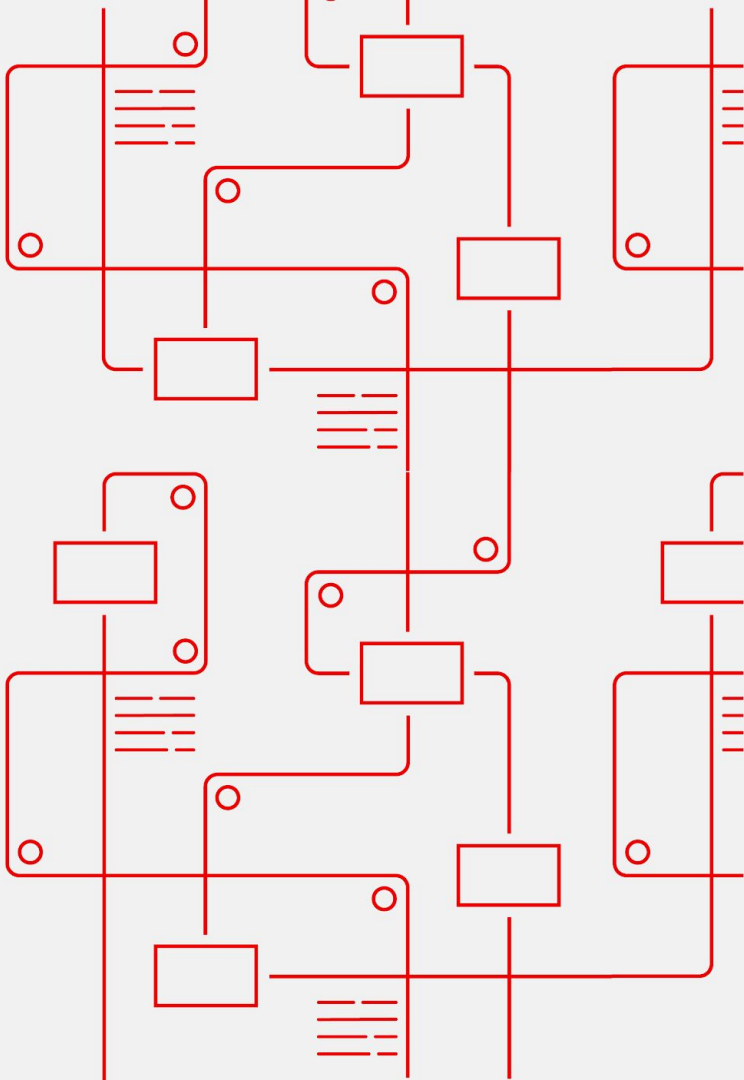


Red Hat

Conclusion: This Isn’t Going to Work

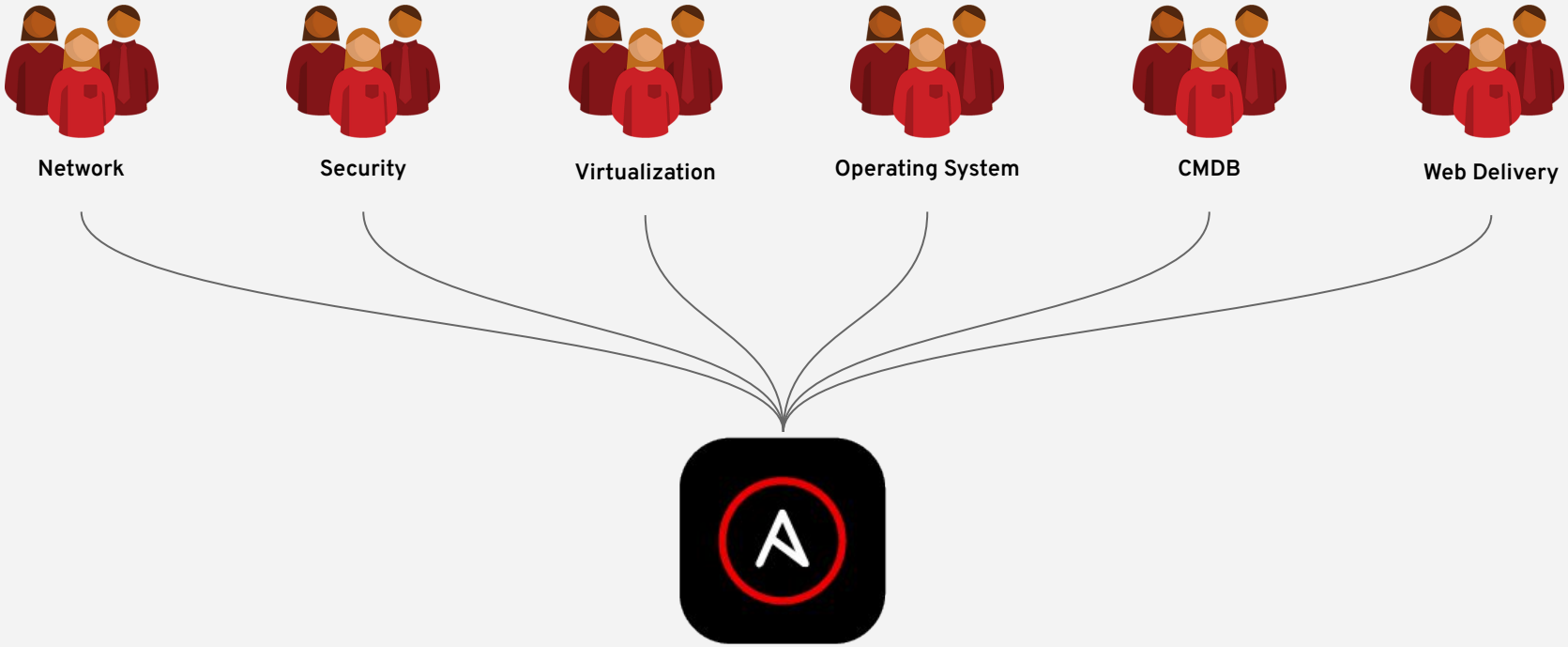


Shifting to a Different Approach

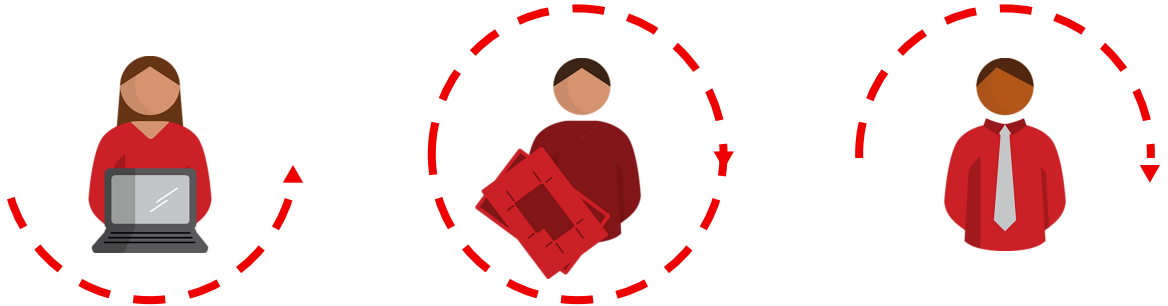


Red Hat Enterprises New Approach

Breaking down barriers by speaking the same language

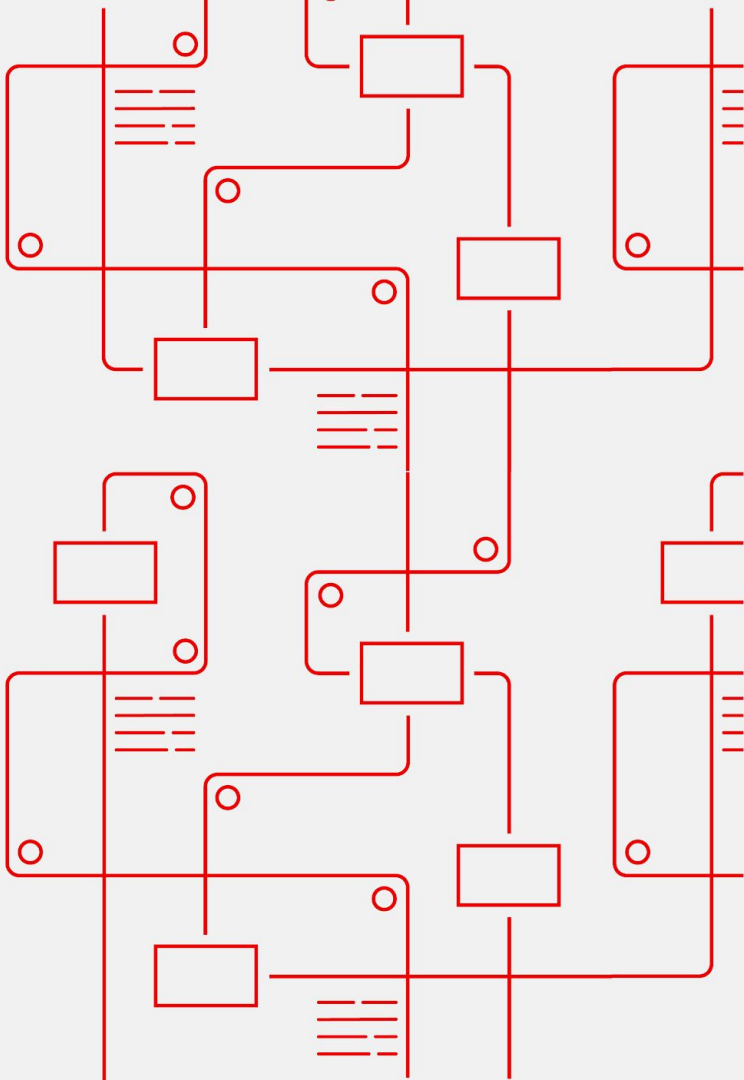


Red Hat Ansible Automation Platform

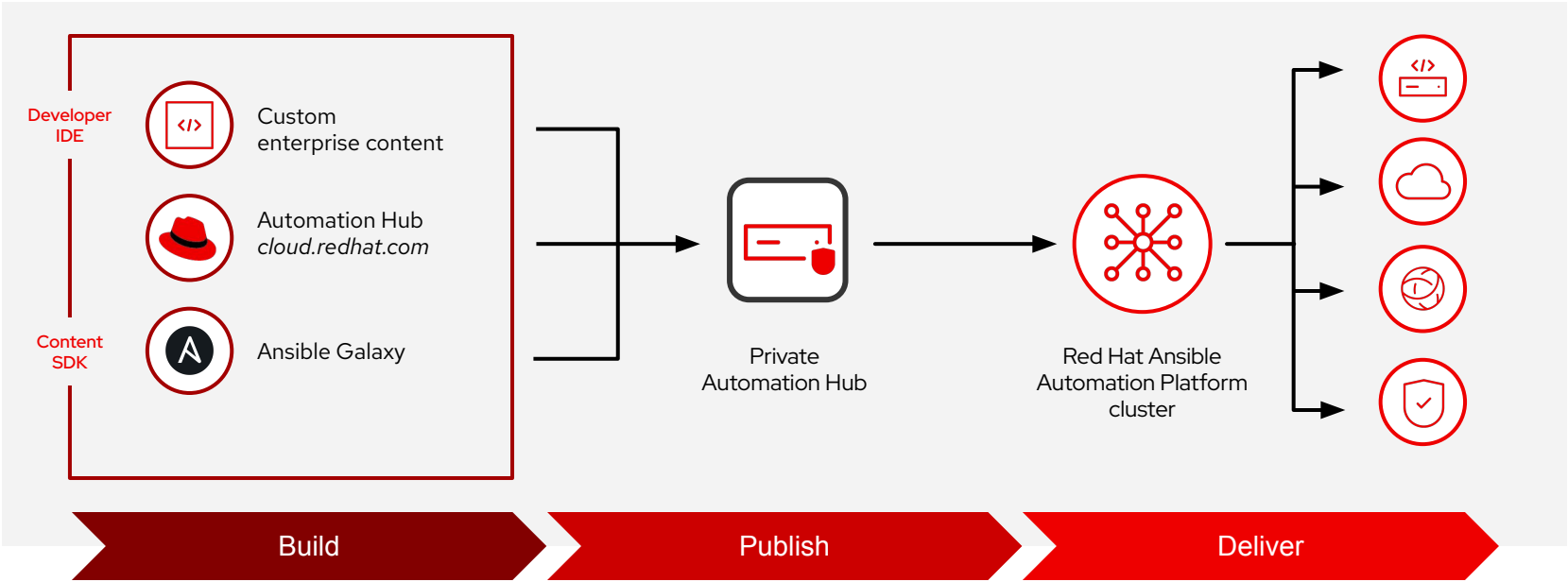


Fueled by an open source community

Ansible Content Flow

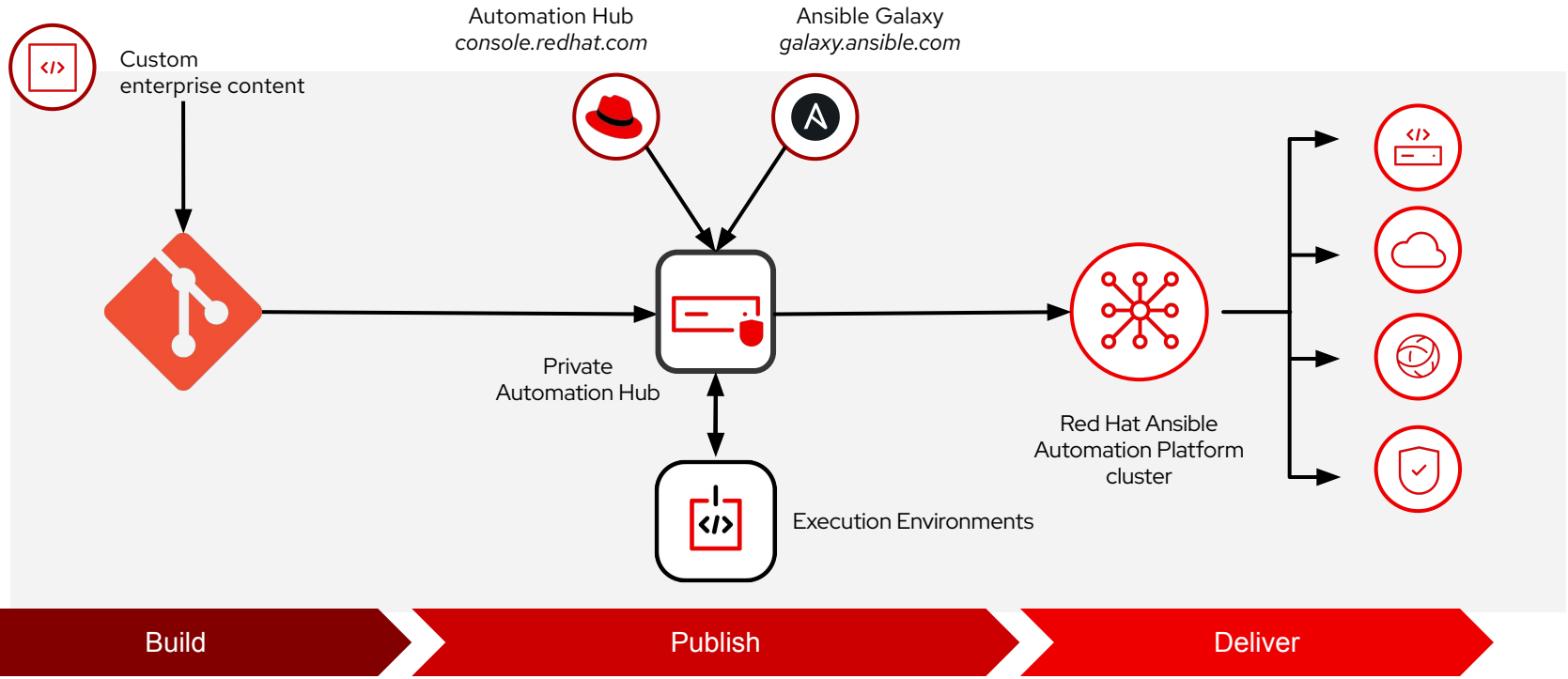


Private Automation Hub architecture



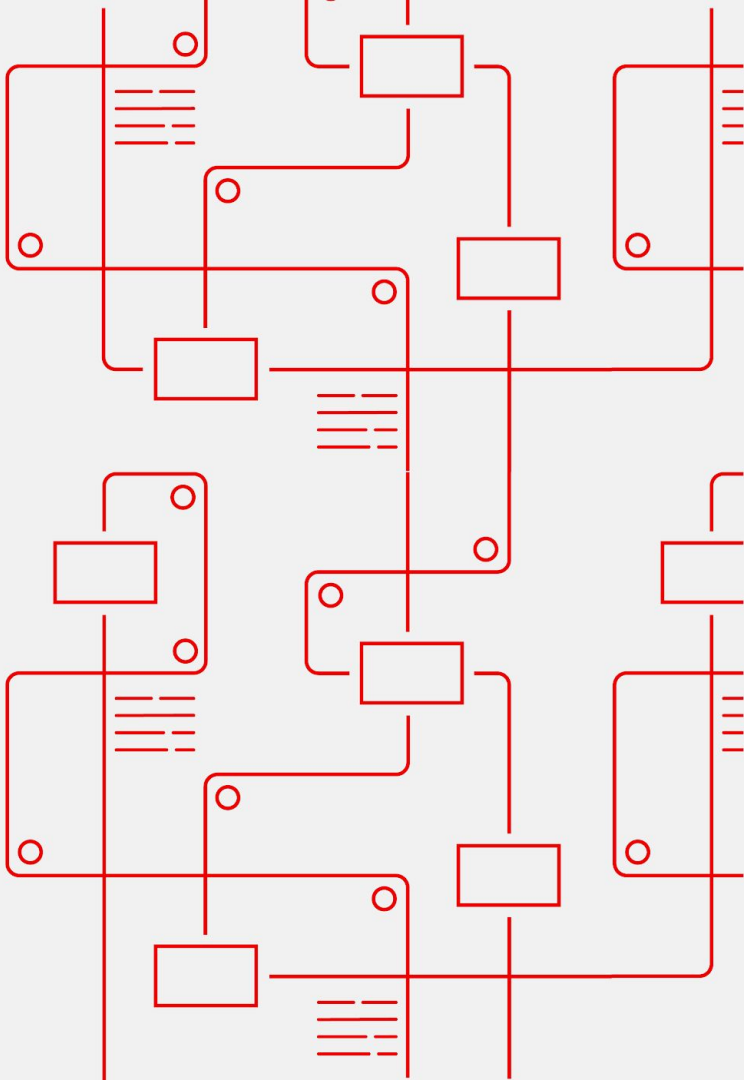
Red Hat

Automation Content Flow



Red Hat

Building Ansible Content

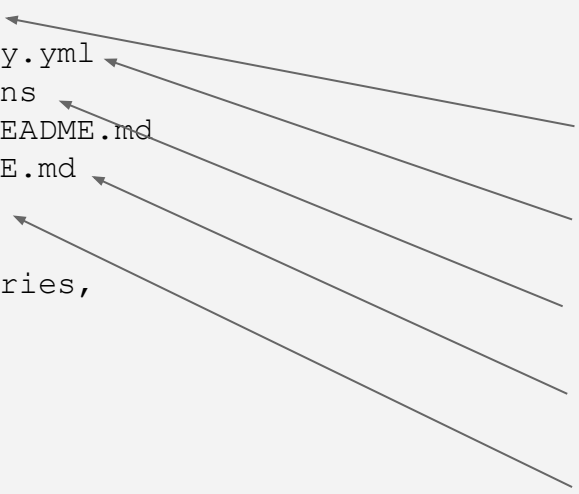


Building Ansible Collections

‘ansible-galaxy collection init’ To Get the Collection Shell

```
[root@DESKTOP-APP17TE ansible-code]# ansible-galaxy collection init my_namespace.my_collection
- Collection my_namespace.my_collection was created successfully
[root@DESKTOP-APP17TE ansible-code]# tree my_namespace/my_collection
my_namespace/my_collection
├── docs
├── galaxy.yml
├── plugins
│   └── README.md
├── README.md
└── roles

3 directories,
3 files
```



docs: Documentation on the collection/plugins

galaxy.yml: Galaxy file for the collection

plugins: Modules, connection plugins, etc

README.md: Readme file for the collection

roles: Ansible roles in the collection

Building Ansible Collections

Leverage GIT Right Away

```
[root@DESKTOP-APP17TE my_collection]# pwd
/root/ansible-code/my_namespace/my_collection
[root@DESKTOP-APP17TE my_collection]# git init
[root@DESKTOP-APP17TE my_collection]# git status
On branch main

No commits yet

Untracked files:
  (use "git add <file>..." to include in what will be committed)
        README.md
        galaxy.yml
        plugins/

nothing added to commit but untracked files present (use "git add" to track)
```

Be in the collection directory

Files/directories created by ‘collection init’

Building Ansible Collections

Suggestion for Namespaces and Collection Names

```
[root@DESKTOP-APP17TE ansible-code]# tree
.
└── my_namespace
    └── my_collection
```

Namespaces indicate a common maintainer or overarching technology, and can contain multiple collections

Collections are organized around a common theme

In an organization, it makes sense to group automation by team or automation domain, and have collections target common tasks or processes

Building Ansible Collections

Suggestion for Namespaces and Collection Names

```
[root@DESKTOP-APP17TE ansible-code]# tree
.
└── network_automation
    └── ip_address_management
```

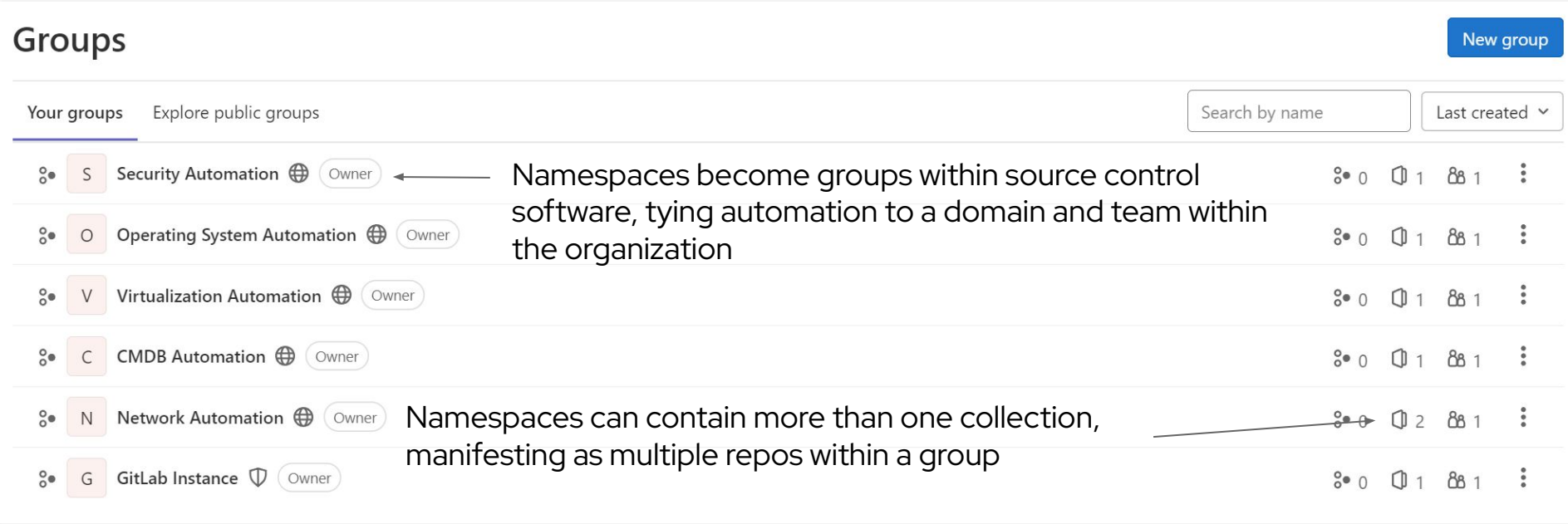
Namespaces indicate a common maintainer or overarching technology, and can contain multiple collections

Collections are organized around a common theme

Here we establish a namespace around automation of networking and network related processes, and have a collection built for IP address management

Building Ansible Collections

Translating Namespaces and Collections to GIT



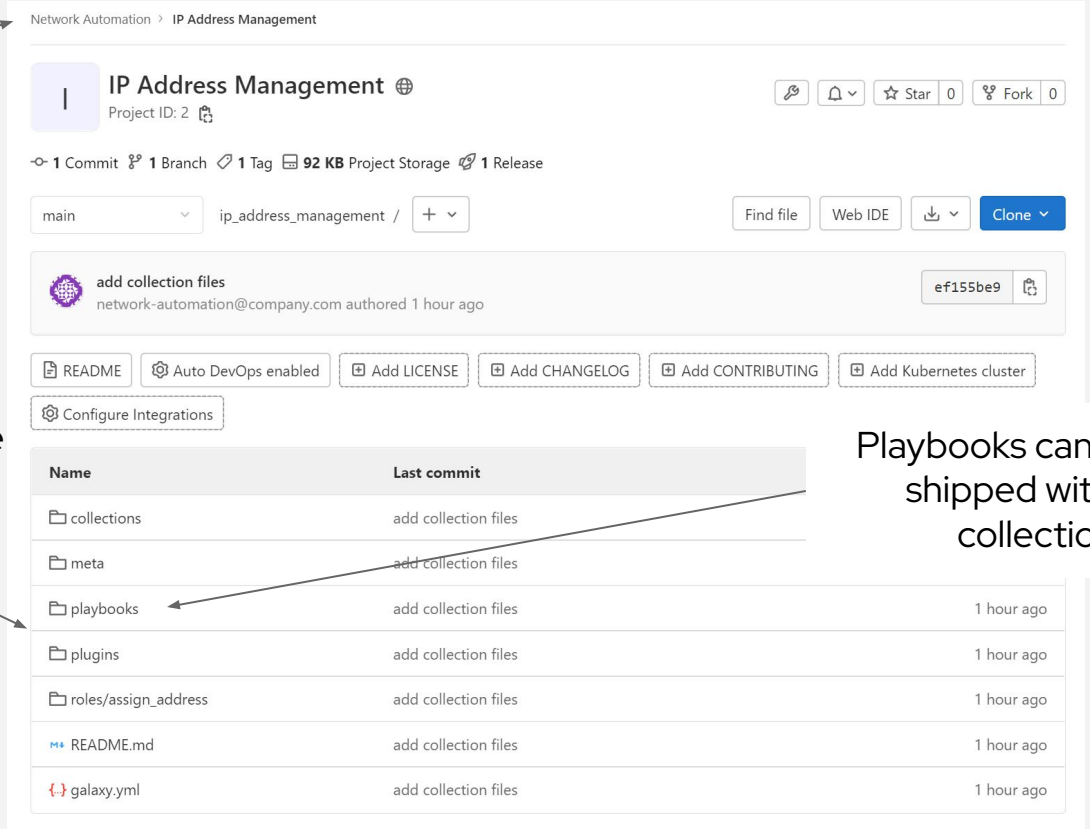
This approach also allows each group to establish their own code standards, apply their own permissions, build their own testing pipelines, and ultimately control the automation they're responsible for

Building Ansible Collections

1:1 Relationship between Collections and Repositories

Namespace and collection names clearly defined

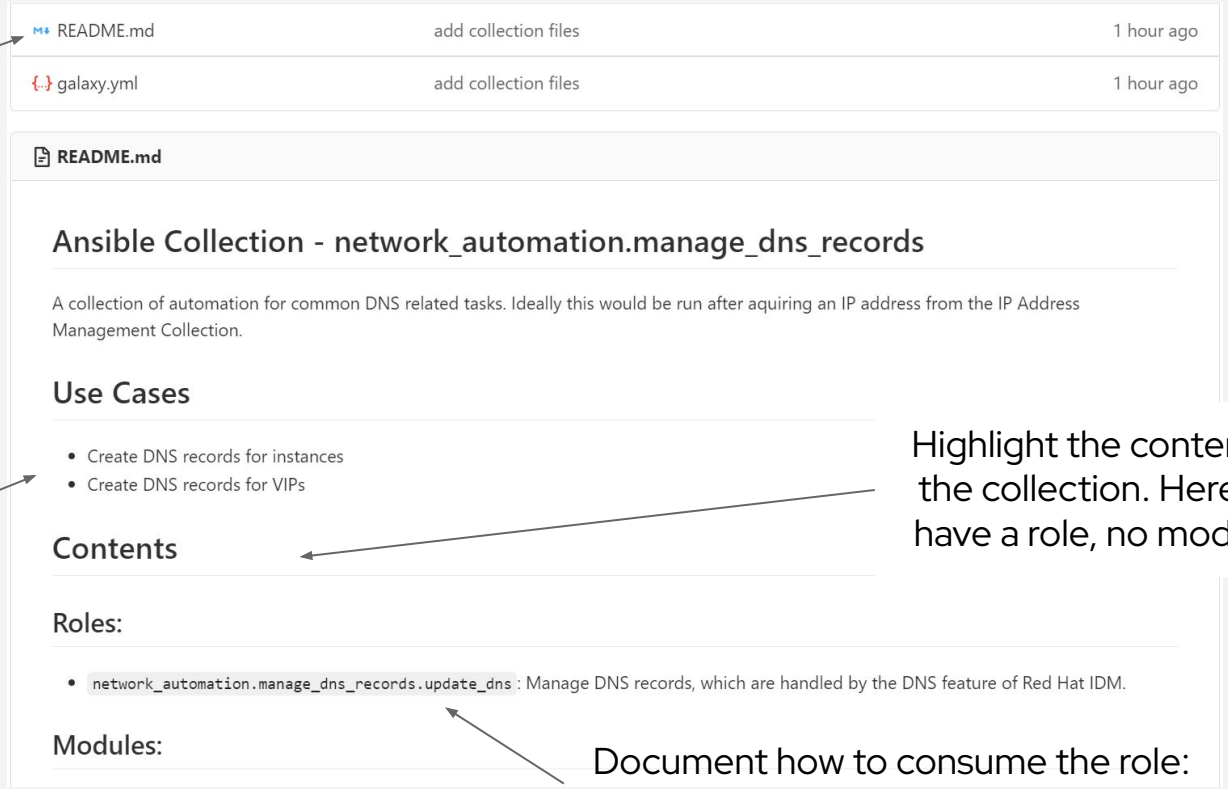
Collection files and directories from our CLI command (plus the automation we added



Playbooks can also be shipped within a collection

Building Ansible Collections

Let The Collection Document Itself



READMEs are required for collections

Clearly outline use cases for the collection

Highlight the content of the collection. Here we have a role, no modules

Document how to consume the role: namespace.collection.role

Building Ansible Collections

Let the Collection Components Document Themselves

README.md add collection files 1 hour ago

README.md

Role Name

network_automation.manage_dns_records.update_dns

Requirements

The following python libraries are required:

- requests

The following collections are required:

- community.general

Role Variables

Some base information should be defined so your records point to the correct place:

```
subnet: 10.15.120.0/22
vlan: 146
address: 10.15.120.123
```

Dependencies

N/A

Example Playbook

```
- name: manage DNS records
  hosts:
    - all
  roles:
    - network_automation.manage_dns_records.update_dns
```

READMEs are required for roles as well

Describe what input or data the role requires

Outline requirements and dependencies on python libraries, other collections, etc

Example playbooks and playbook runs make great documentation as well

Building Ansible Collections

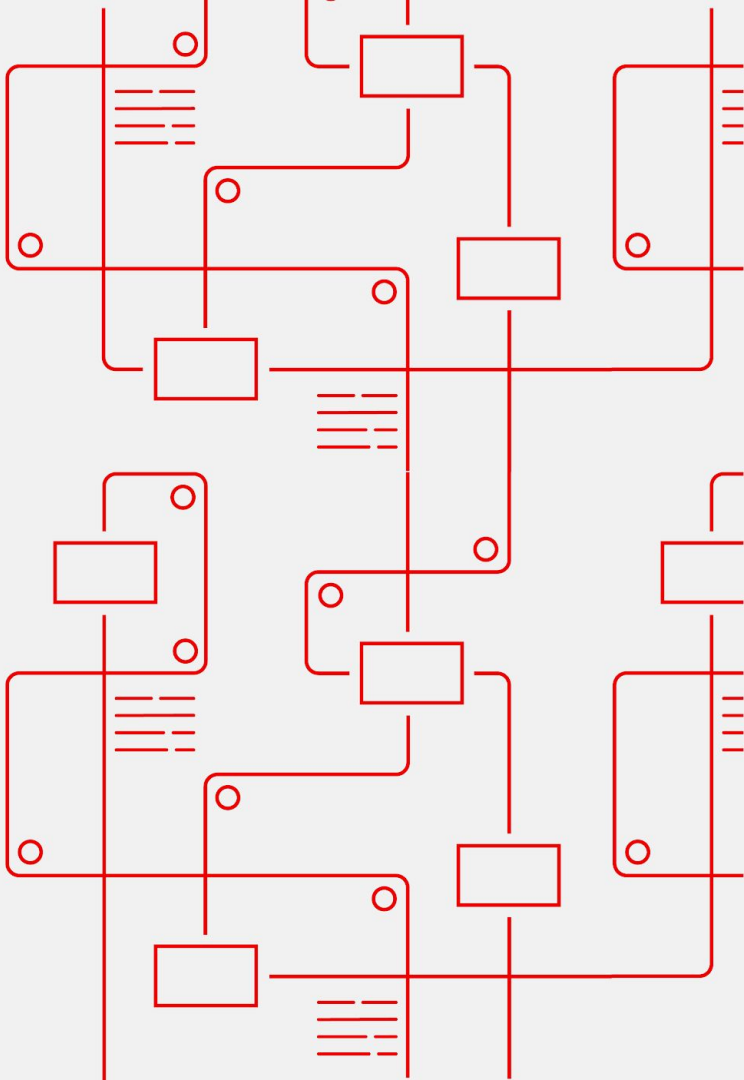
Don't Forget galaxy.yml

```
namespace: network_automation
name: manage_dns_records
version: 1.0.0
readme: README.md
authors:
- Josh Swanson <jswanson@redhat.com>
description: A collection for grabbing available IP address(es)
license:
- GPL-2.0-or-later
tags:
  - dns
  - idm
  - freeipa
dependencies: {}
repository: https://msp-gitlab.rhug.demos.lcl/network_automation/manage_dns_records
documentation:
https://msp-gitlab.rhug.demos.lcl/network_automation/manage_dns_records/-/wikis/home
homepage: https://msp-gitlab.rhug.demos.lcl/network_automation/manage_dns_records
issues: https://msp-gitlab.rhug.demos.lcl/network_automation/manage_dns_records/-/issues
build_ignore:
  - collections/requirements.yml
  - .git
```

Tags help the collection be found in hub

Many source control systems have wikis and issue trackers built in

Publishing Ansible Content



Publishing Ansible Collections

Uploading to Automation Hub

```
[root@DESKTOP-APP17TE my_collection]# pwd
/root/ansible-code/my_namespace/my_collection
[root@DESKTOP-APP17TE my_collection]# ansible-galaxy collection build
Created collection for my_namespace.my_collection at
/root/ansible-code/my_namespace/my_collection/my_namespace-my_collection-1.0.0.tar.gz
```

Be in the collection directory

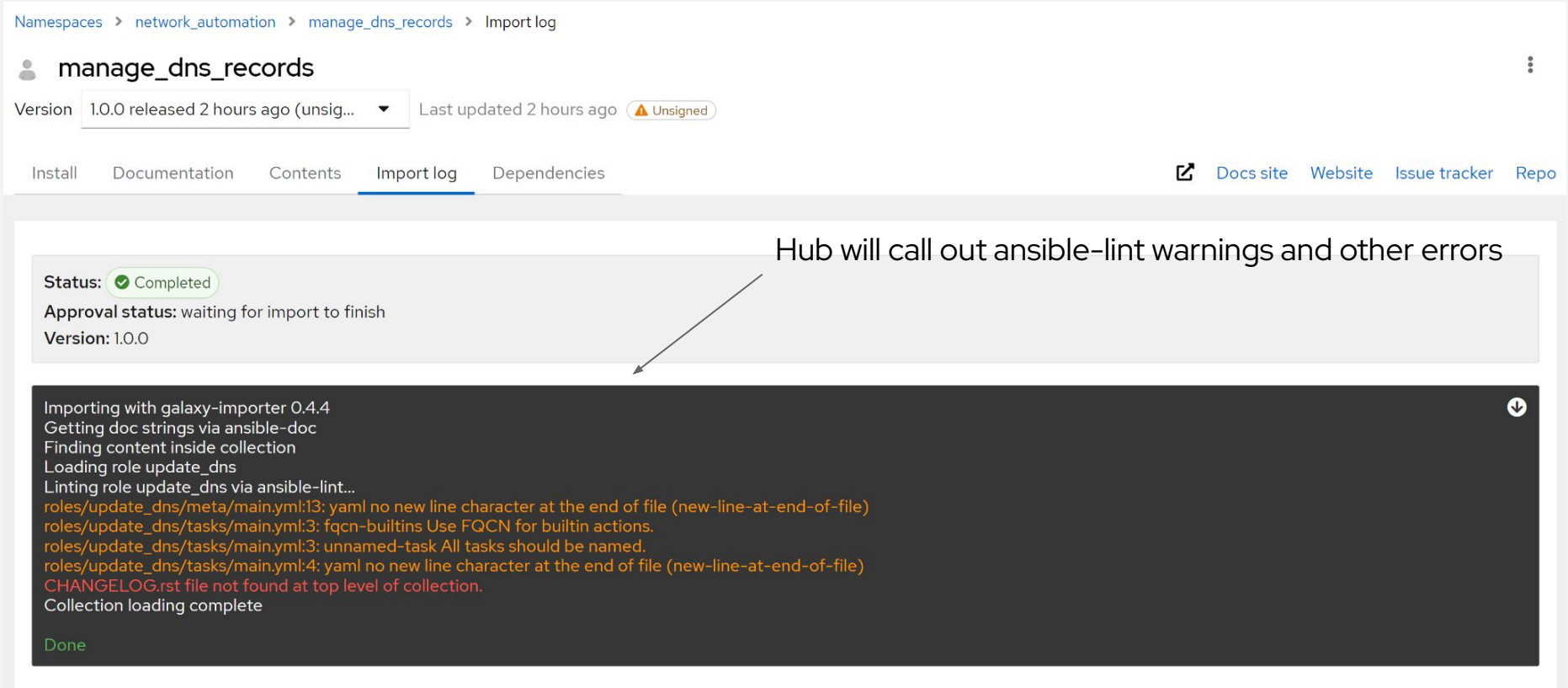
Ansible will build our collection into an archive

Once the archive is created, it can be uploaded to Automation Hub

This is the manual process, however this is a great opportunity to build a testing and publishing pipeline using your favorite pipeline software (Gitlab CI, Github Actions, Jenkins) - Giving your team a more automated approach to developing, testing, and publishing automation

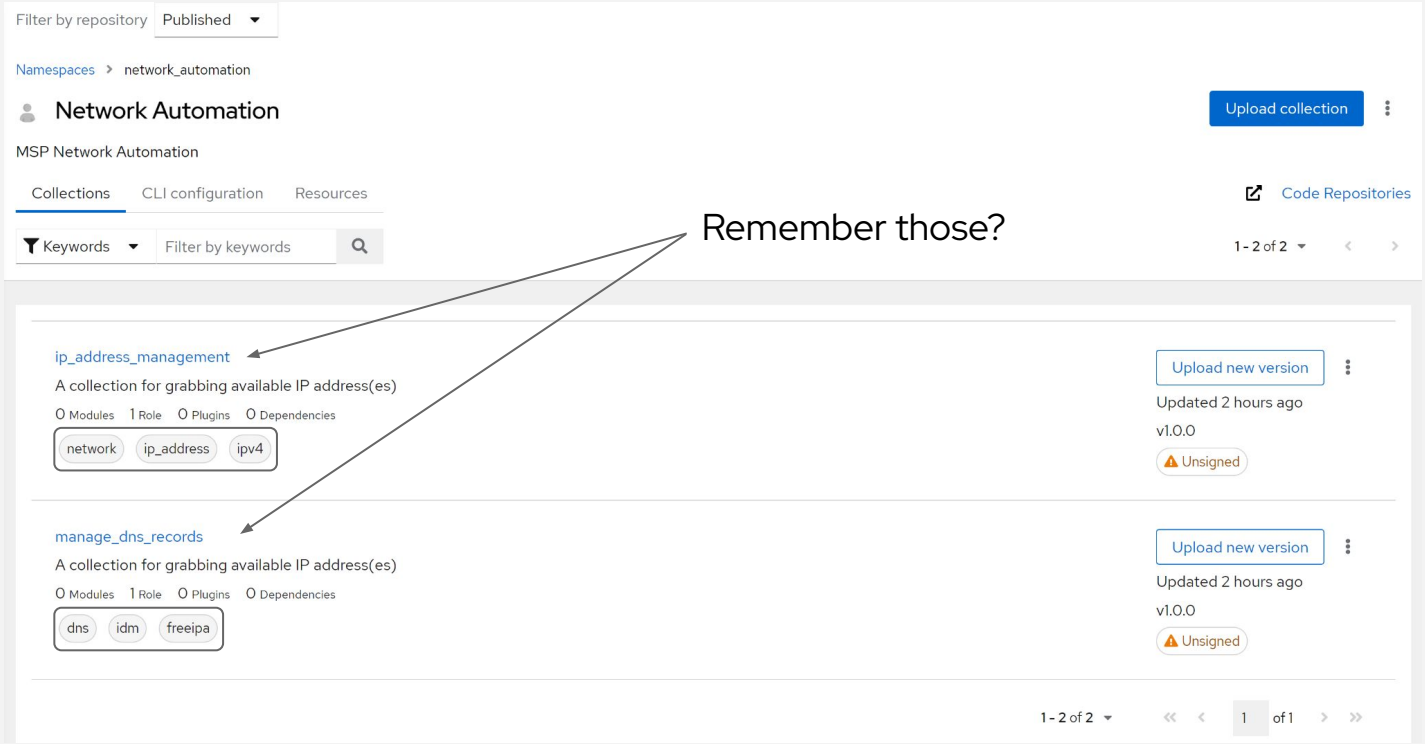
Publishing Ansible Collections

Automation Hub's Importer



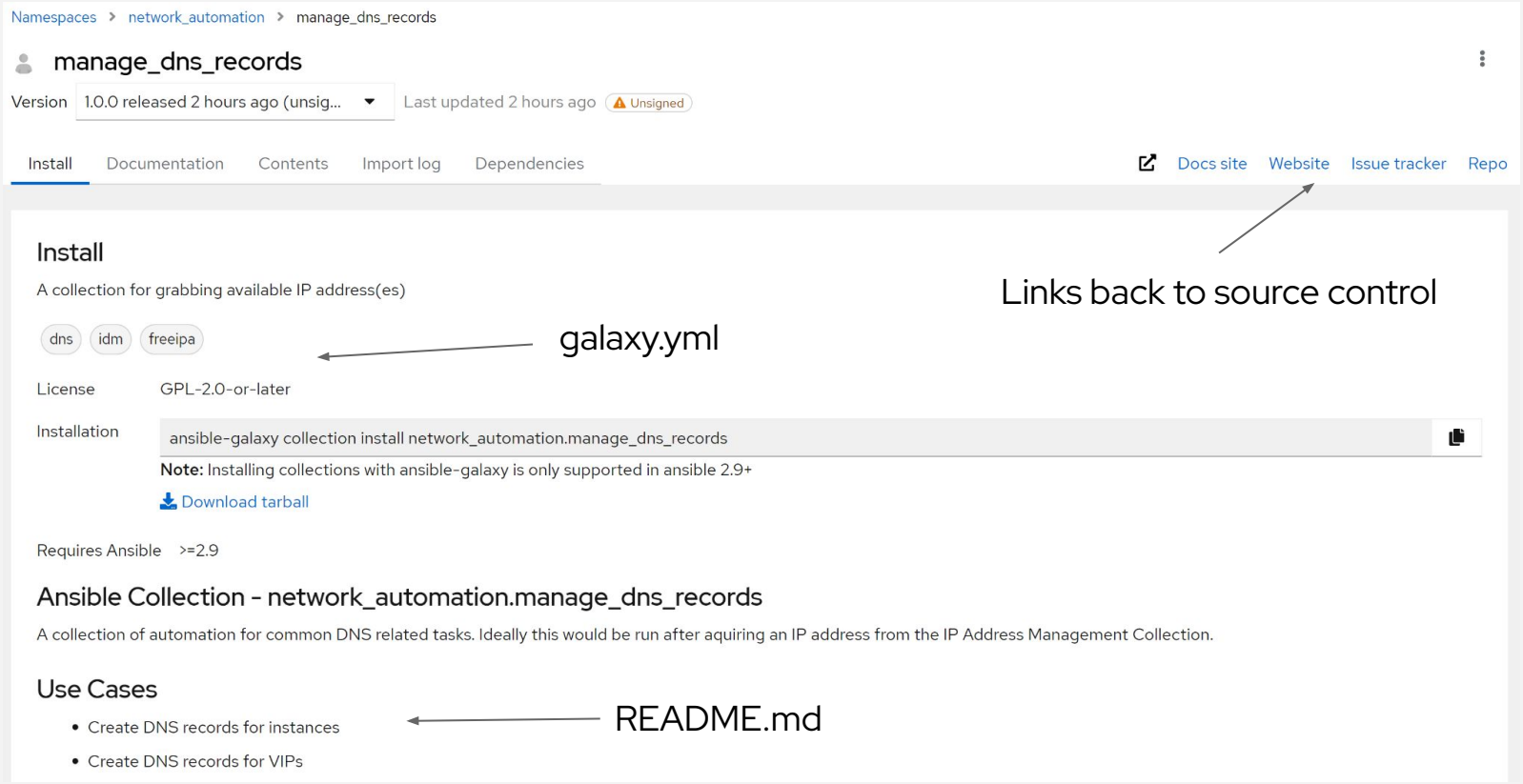
Publishing Ansible Collections

Automation Hub Organization



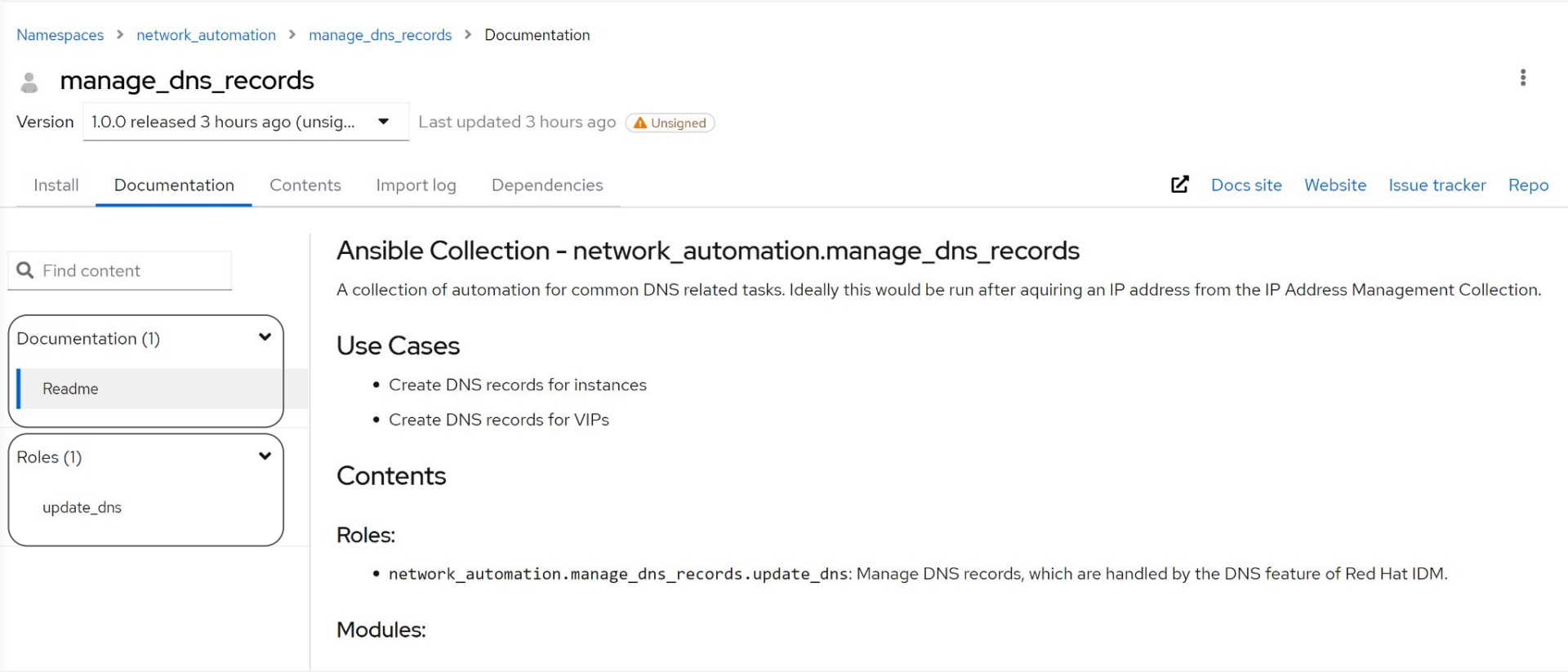
Publishing Ansible Collections

Collection Point of View

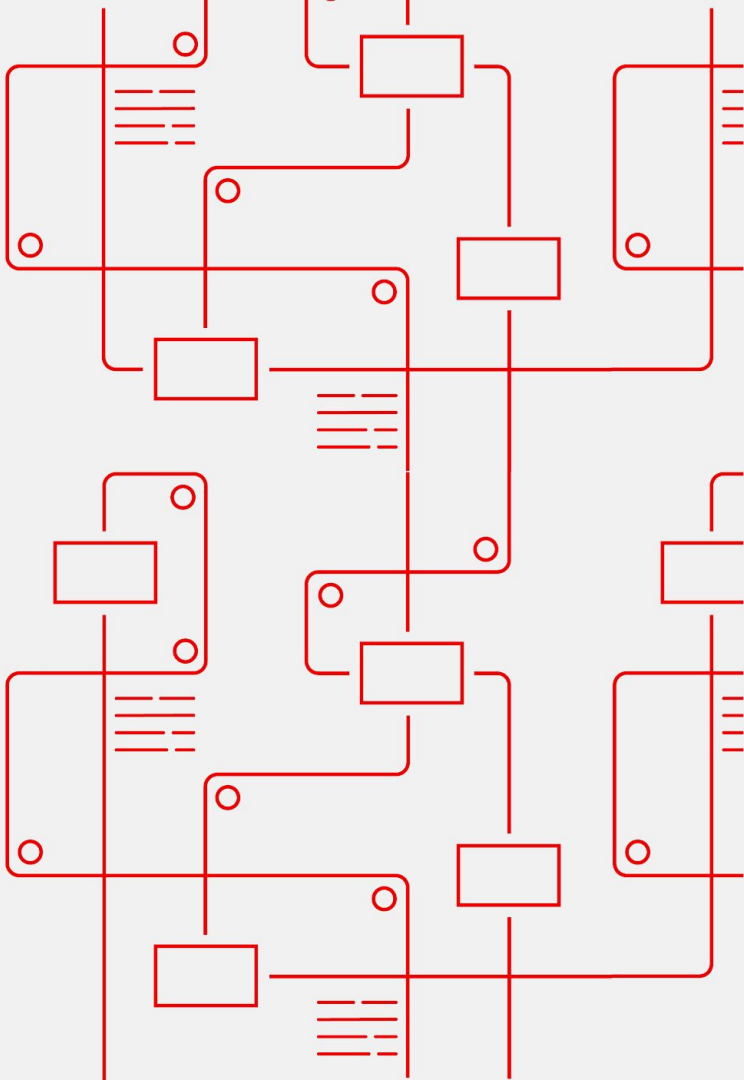


Publishing Ansible Collections

Hub Publishes Our Documentation

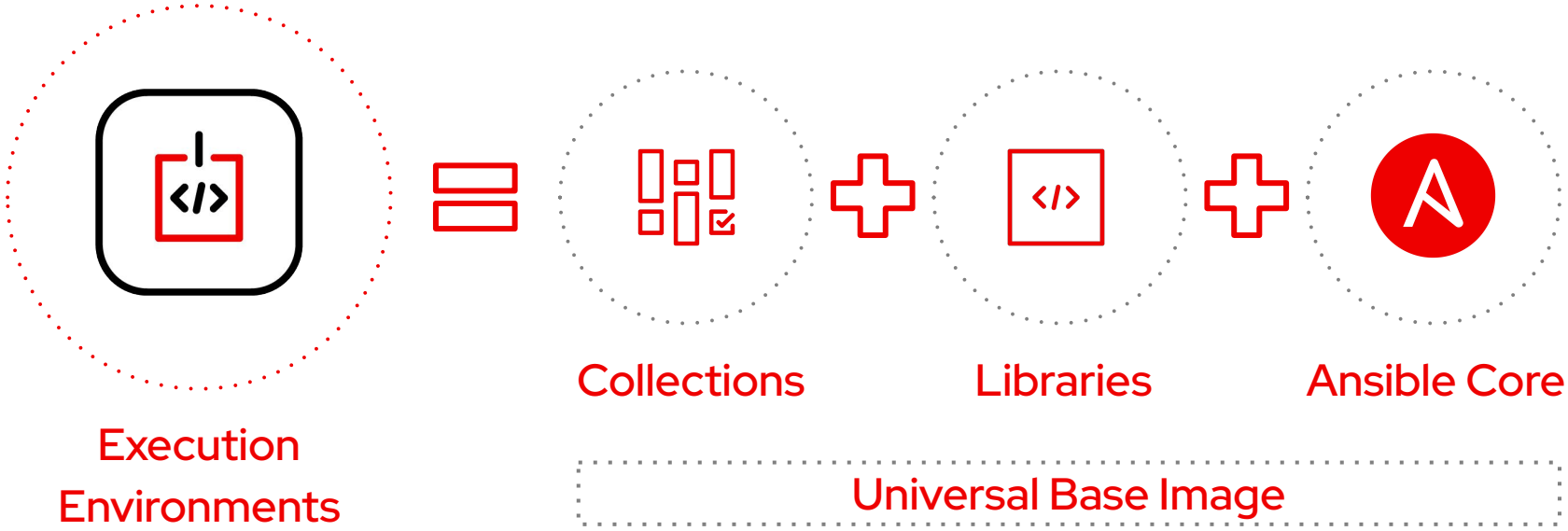


Delivering Ansible Collections



Automation Execution Environments

Components needed for automation, packaged in a cloud-native way



Delivering Ansible Collections

Building our Execution Environment

```
[root@DESKTOP-APP17TE execution-environment]# cat execution-environment.yml
---
version: 1
ansible_config: 'ansible.cfg'
build_arg_defaults:
  EE_BASE_IMAGE: 'registry.redhat.io/ansible-automation-platform-22/ee-minimal-rhel8:latest'
dependencies:
  galaxy: requirements.yml
  python: requirements.txt
  system: bindep.txt
```

Define your instance of Automation Hub

The base image (just like a container, because it is a container) for the execution environment

The additional pieces we want in this execution environment, broken out into files

Delivering Ansible Collections

Building our Execution Environment

```
[root@DESKTOP-APP17TE execution-environment]# cat requirements.yml
---
roles:
collections:
  # Our collections
  - name: security_automation.operating_system
  - name: os_automation.rhel
  - name: virtualization_automation.provision_system
  - name: cmdb_automation.manage_records
  - name: network_automation.manage_dns_records
  - name: network_automation.manage_ip_addresses

  # Supported collections
  - name: redhat.satellite
  - name: ansible.posix
  - name: ansible.windows
  - name: redhat.rhel_system_roles
```

No roles needed, just collections

Pull in collections from Automation Hub

Pull in collections from console.redhat.com
or a downstream Automation Hub

Delivering Ansible Collections

Building our Execution Environment

```
[root@DESKTOP-APP17TE execution-environment]# cat requirements.txt
jmespath
pywinrm
requests
pynetbox
python-gitlab
psutil

[root@DESKTOP-APP17TE execution-environment]# cat bindep.txt
git
unzip
```

Python libraries to be included in the execution environment

System utilities to be installed into the execution environment

Delivering Ansible Collections

Building our Execution Environment

```
[root@DESKTOP-APP17TE execution-environment]# ansible-builder build
```

This may take a bit.

For more information, run with '-v 3' - more information will be printed to the command line as the execution environment is built.

Once complete, treat it like a container image - tag and push it to the image registry built into Automation Hub or your favorite image registry

Automation Mesh

Simple, flexible and reliable scaling of execution capacity

Automate at a global scale

Simple, flexible and reliable way to scale automation of large inventories across diverse network topologies and platforms.

Distributed overlay network

Overlay network which eases distributing automation execution

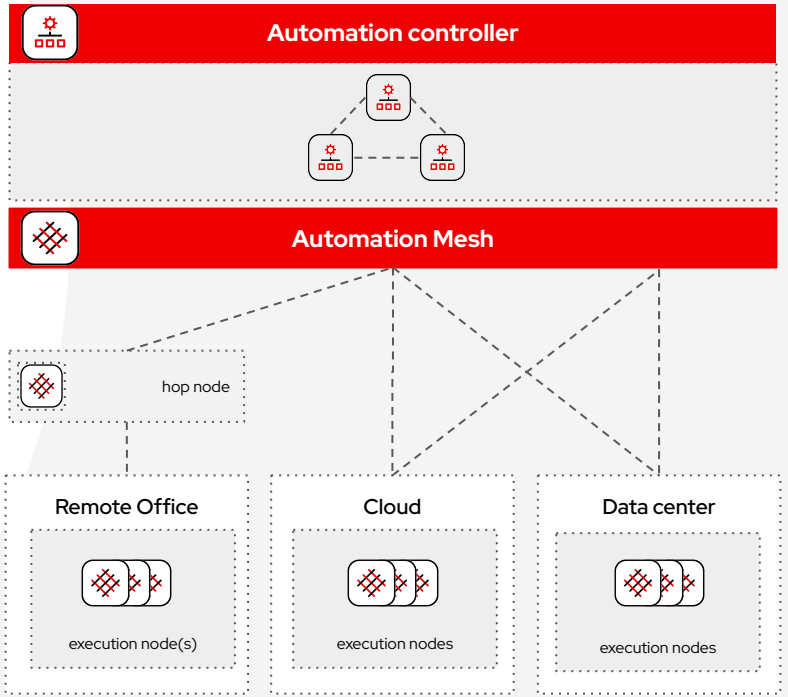
Establishes peer-to-peer connections between execution nodes across existing networks

Flexible architecture

Flexible architecture offers more design choices compared to isolated nodes

Execution node health

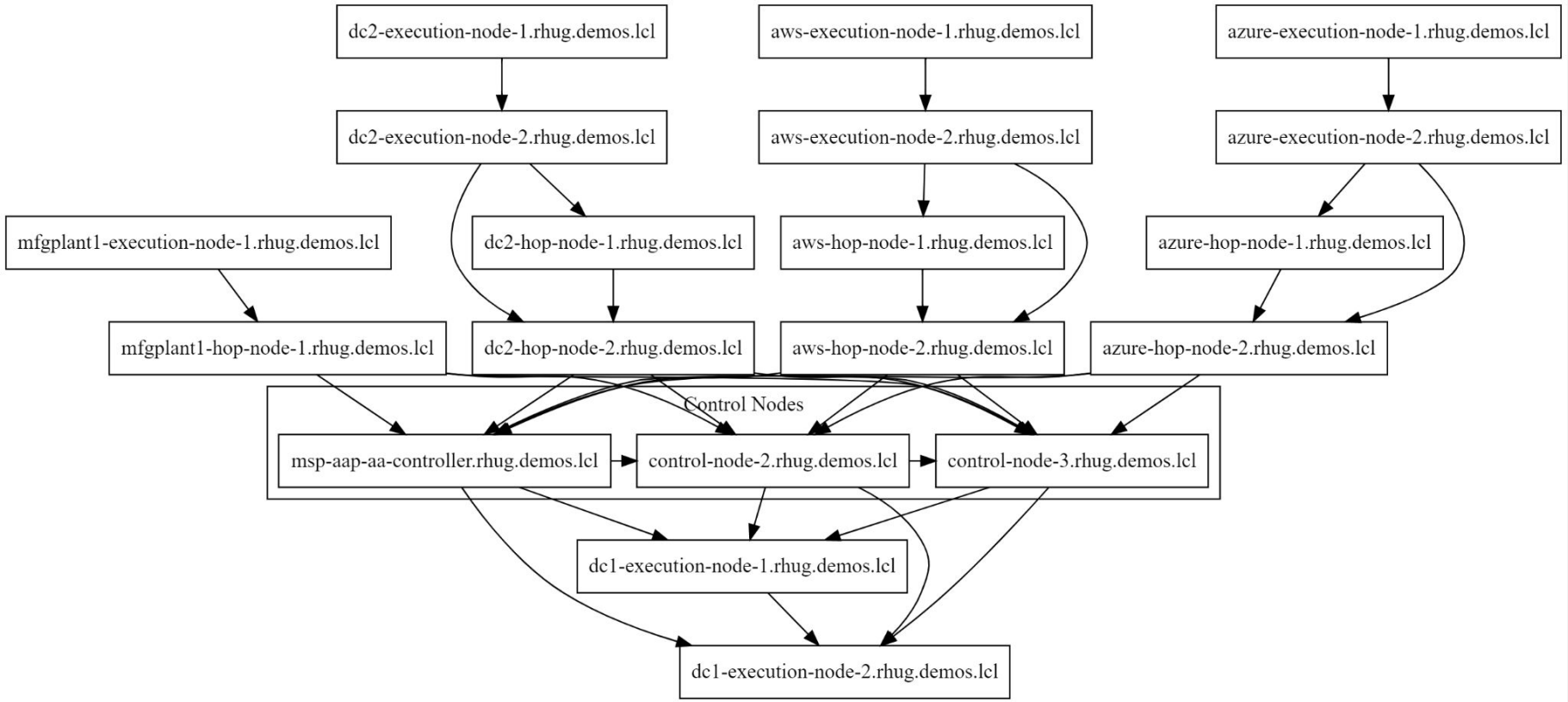
Health checks performed on execution nodes



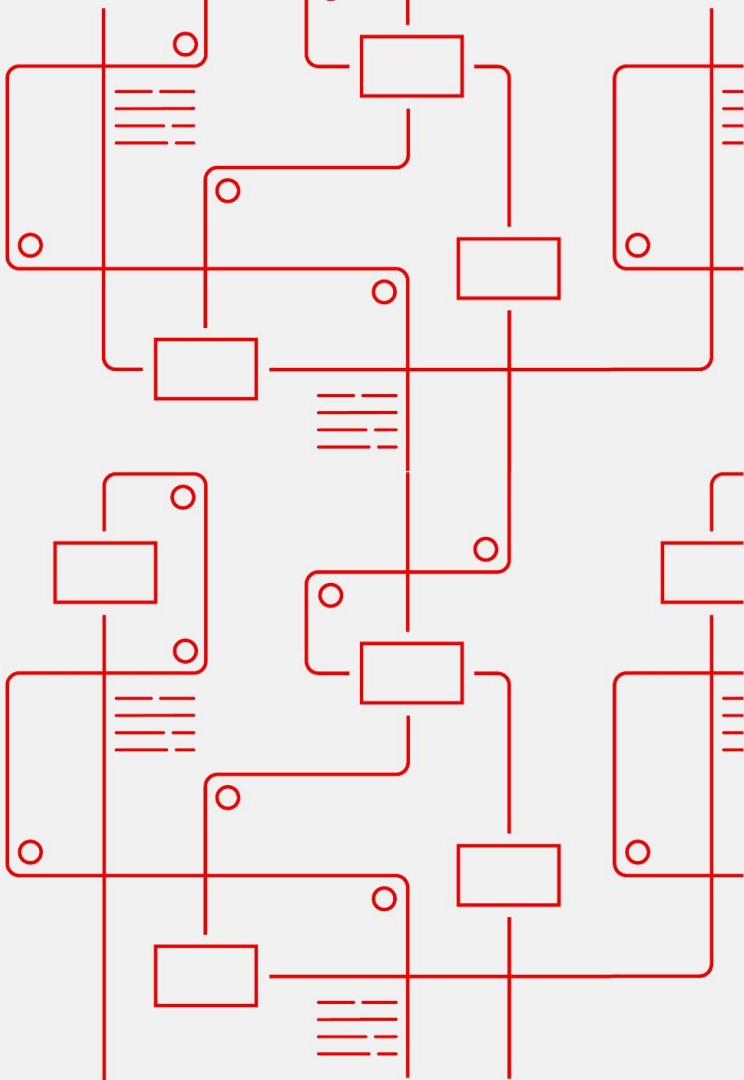
Red Hat

Delivering Ansible Content

Mesh + Execution Environments == Automation Everywhere

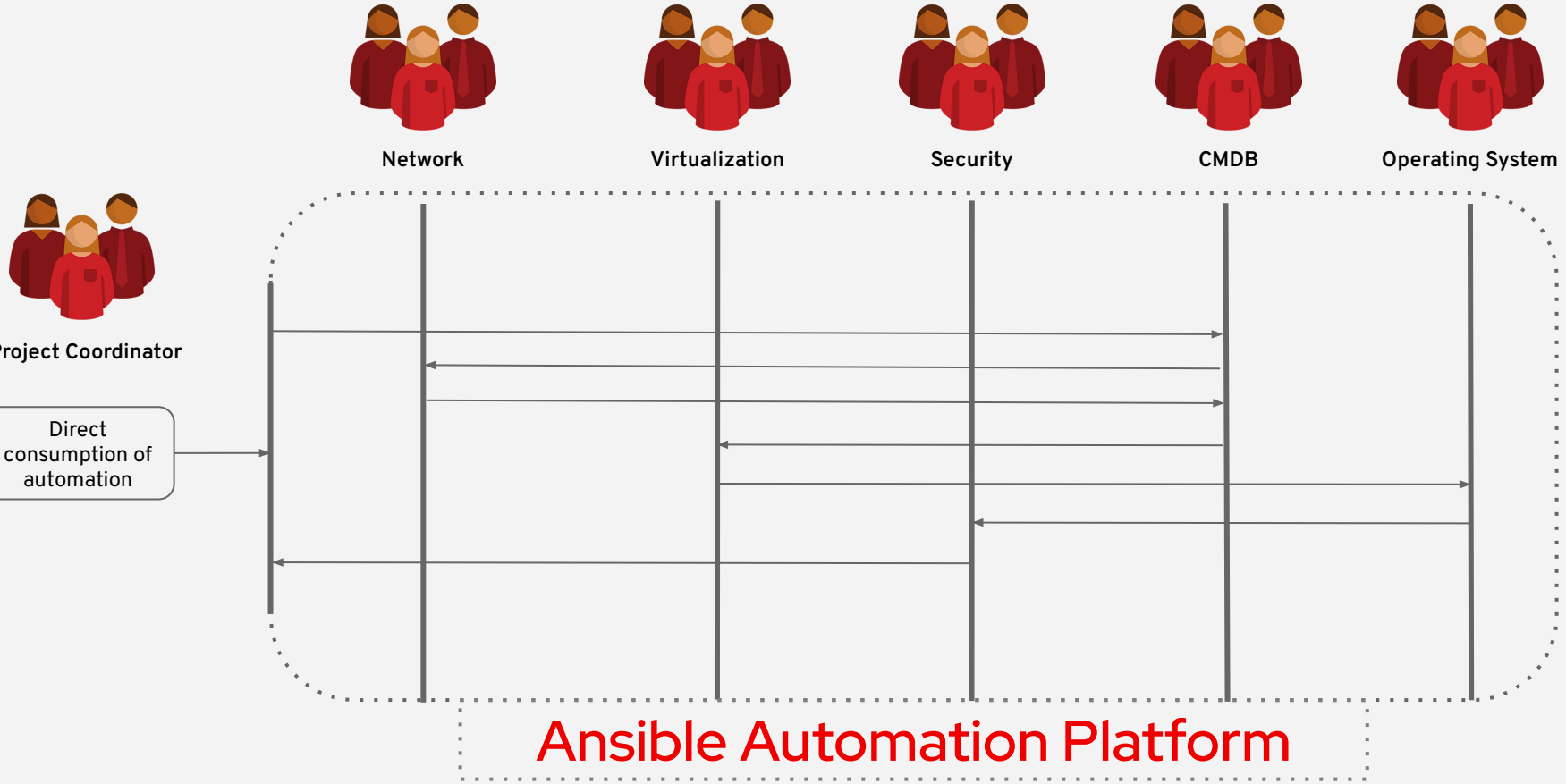


Quick Review: Where We Are Now



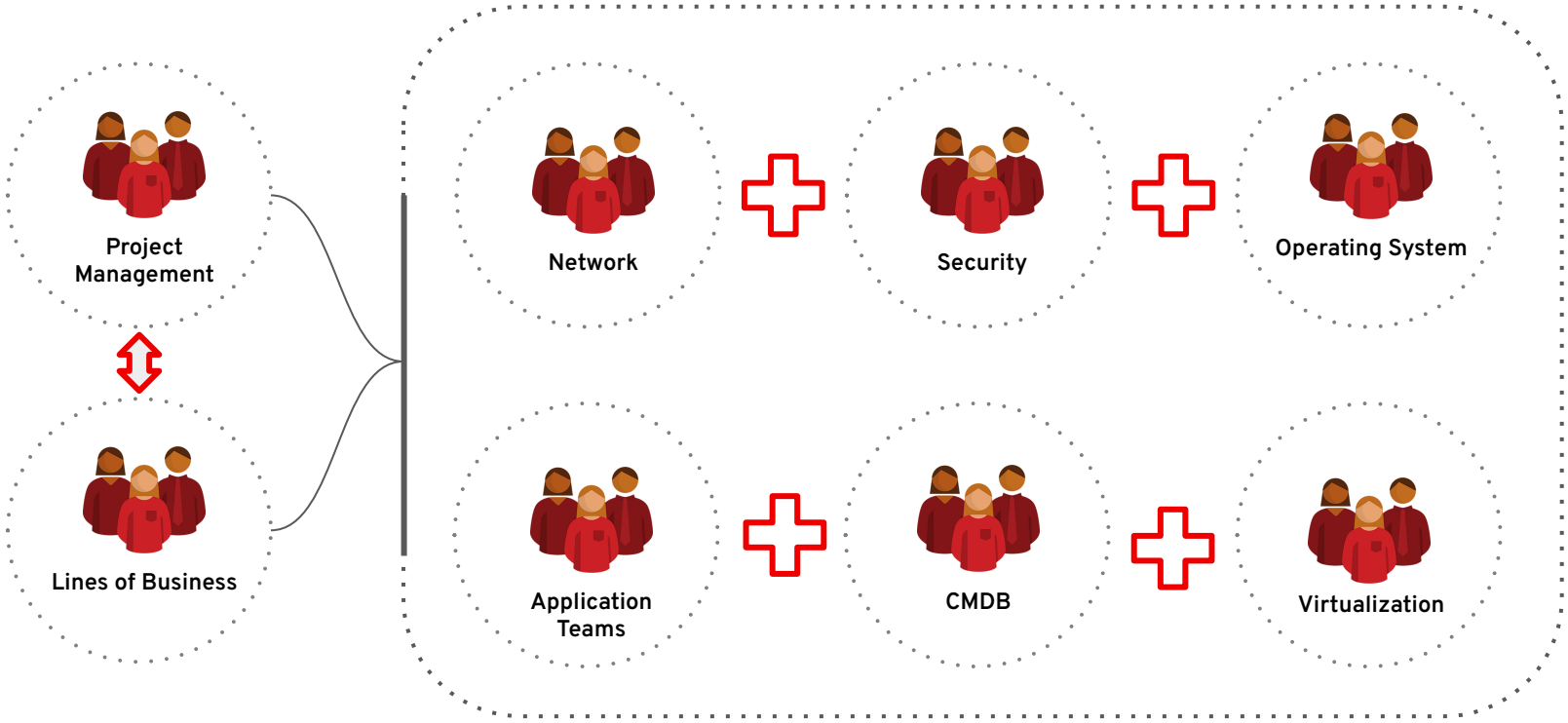
Red Hat Enterprises New Approach

A much more efficient process



Automating the Last Mile

Bringing all stakeholders together



Red Hat

Red Hat
Ansible Automation
Platform

Q+A

Red Hat
Ansible Automation
Platform

Thank You!

The journey of a thousand automated processes begins with a single playbook.

- *Confucius, probably*