

Ansible Automation Platform 2

Execution Environments

Command line tools for Content Creators

Brian Dumont, RHCA

Lt. Principal Solution Architect



Red Hat

Ansible Automation
Platform

Agenda

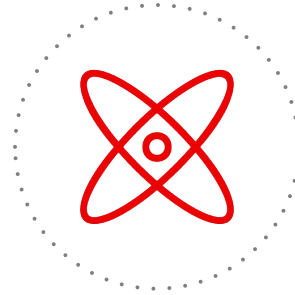
- Quick Platform Overview
- Content Creators Persona
- Execution Environments Overview
- Deep Dive on Command Line tools
- Demo

What makes a platform?



Red Hat Ansible Automation Platform

Combining the universal automation language with cloud services and certified content for automating, deploying, and operating applications, infrastructure and services securely at enterprise scale.



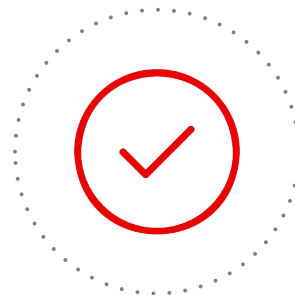
Ansible automation

Providing scalable, secure implementation for describing, building, and managing the deployment of enterprise IT applications across diverse enterprise architectures.



Cloud services

Cloud services that facilitate team collaboration and provide operational analytics for automating heterogeneous, hybrid environments.



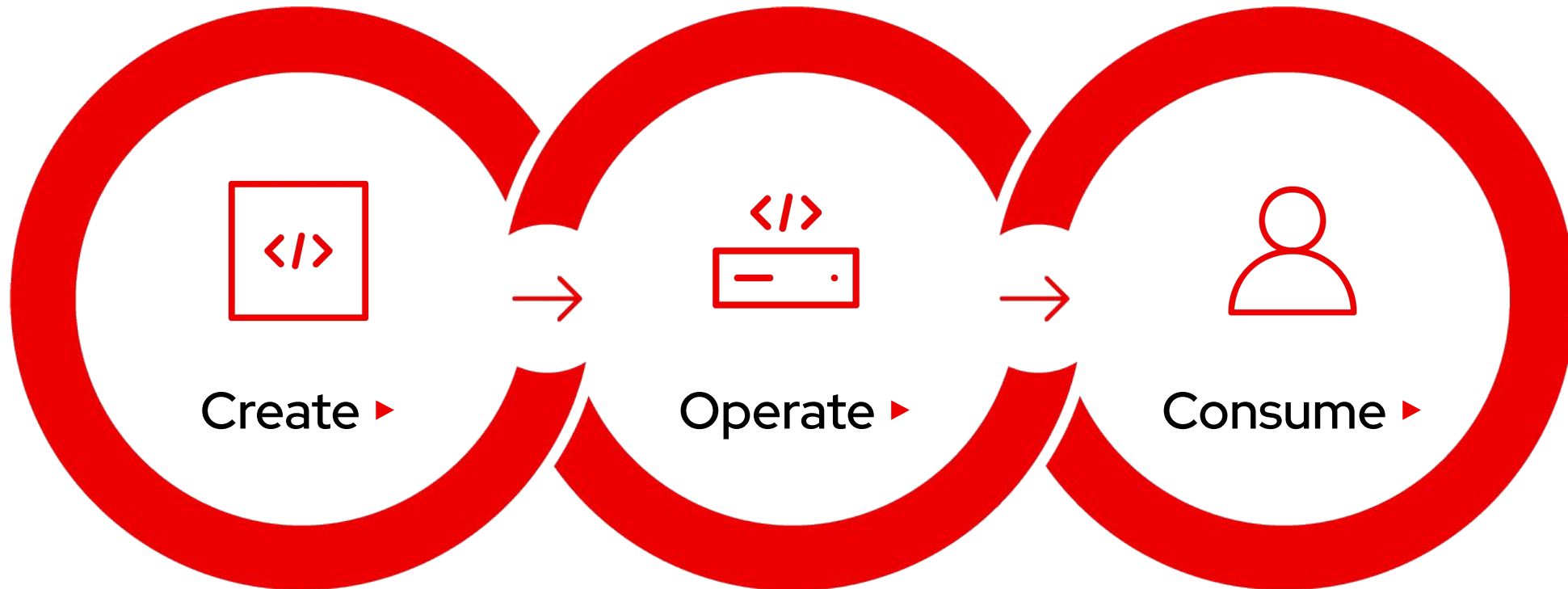
Certified content

Extends native platform capabilities with certified, supported content designed to expand the automation domain and accelerate adoption for enterprise customers.



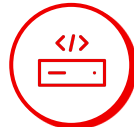
Red Hat Ansible Automation Platform

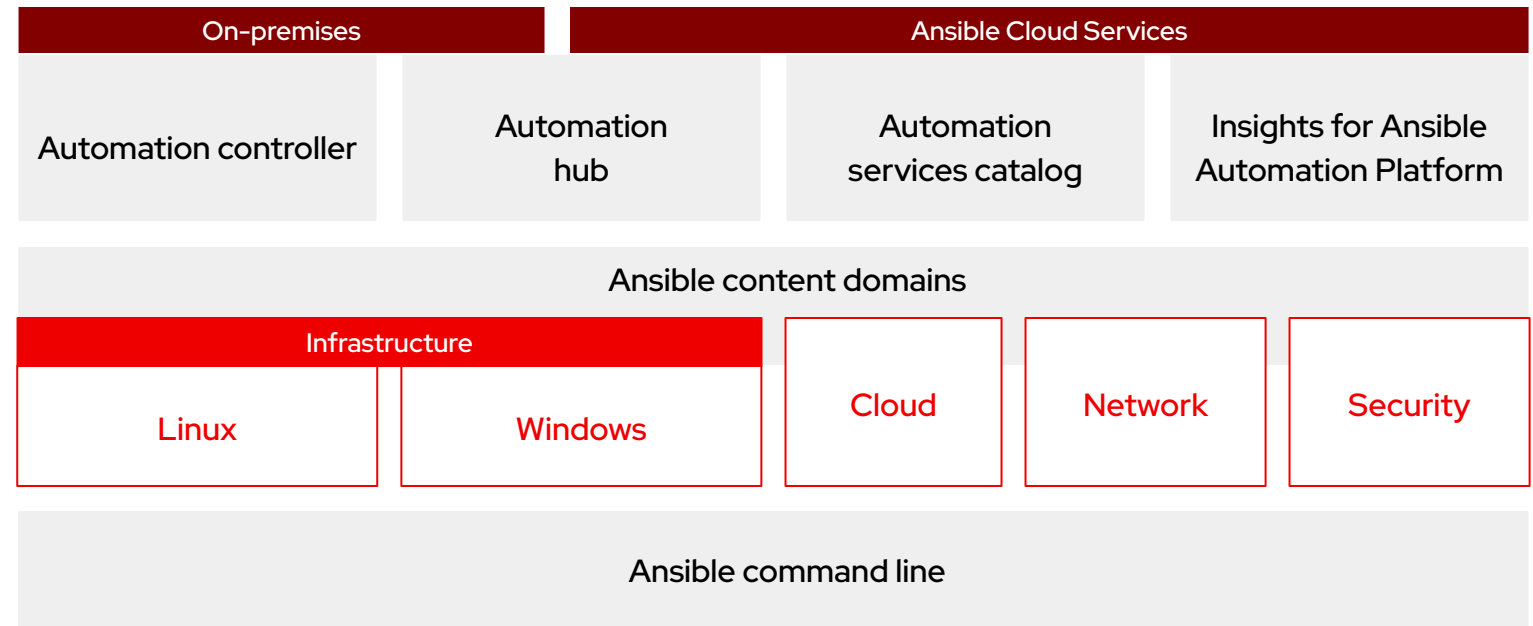
Holistic automation for your enterprise



What makes a platform?

Red Hat Ansible Automation Platform

-  Content creators
-  Operators
-  Domain experts
-  Users



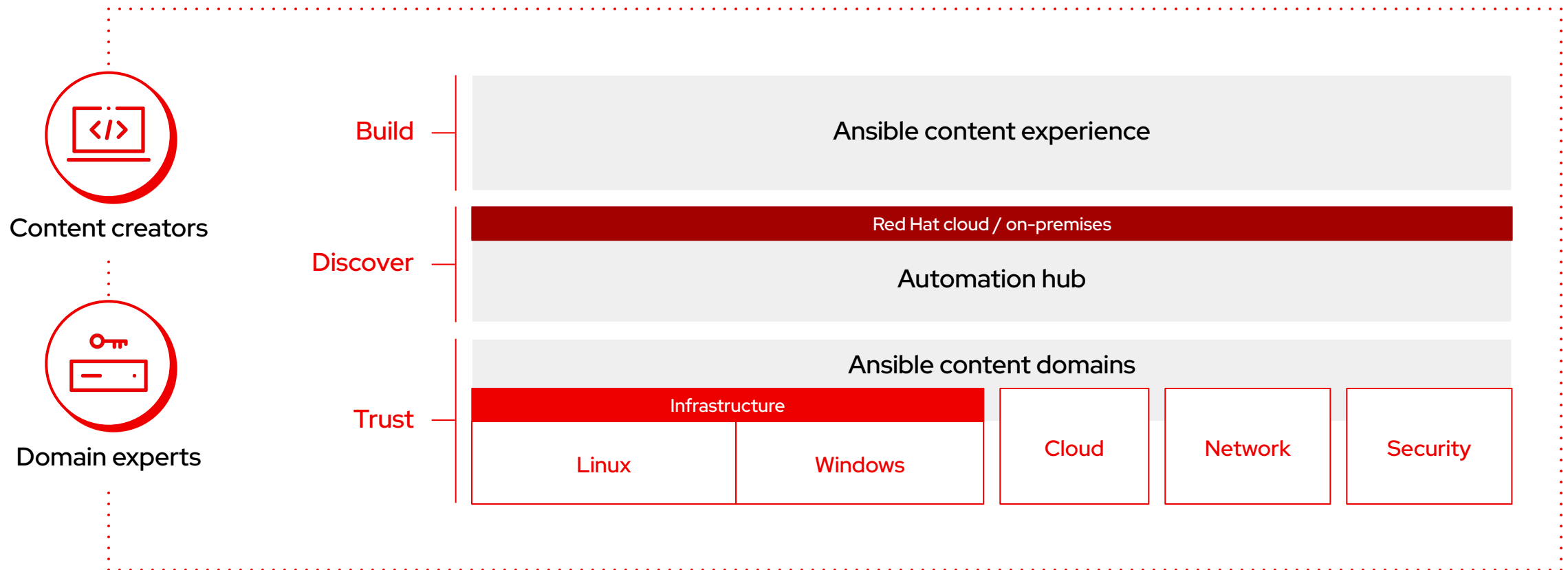
Fueled by an open source community



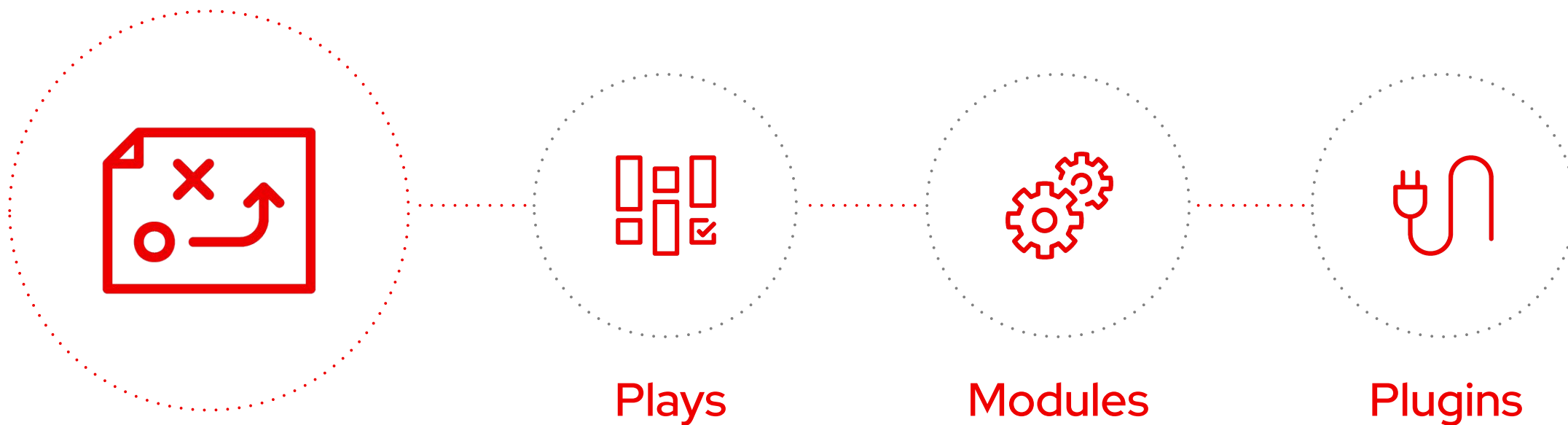
Create

Create

The automation lifecycle



What makes up an Ansible playbook?



Collections

Simplified and consistent content delivery



What are they?

Collections are a data structure containing automation content:

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





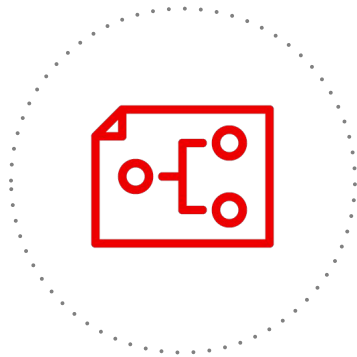
```
nginx_core
├── MANIFEST.json
├── playbooks
│   └── deploy-nginx.yml
│       ...
├── plugins
├── README.md
├── roles
│   ├── nginx
│   │   ├── defaults
│   │   ├── files
│   │   │   └── ...
│   │   ├── tasks
│   │   └── templates
│   │       └── ...
│   ├── nginx_app_protect
│   └── nginx_config
```

deploy-nginx.yml

```
---
- name: Install NGINX Plus
  hosts: all
  tasks:
    - name: Install NGINX
      include_role:
        name: nginxinc.nginx
      vars:
        nginx_type: plus
    - name: Install NGINX App Protect
      include_role:
        name: nginxinc.nginx_app_protect
      vars:
        nginx_app_protect_setup_license: false
        nginx_app_protect_remove_license: false
        nginx_app_protect_install_signatures: false
```

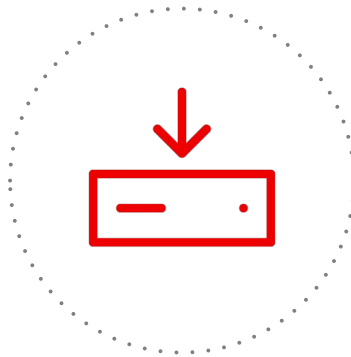
Accessing collections

How to get them



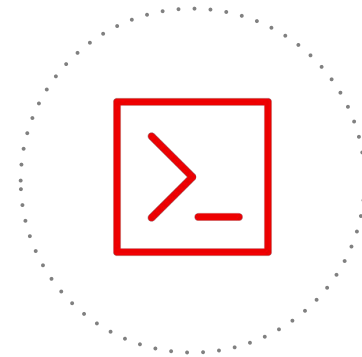
Requirements file

Requirements file defines the required collections for a playbook



Pull via controller

Automation controller pulls the collections from Automation Hub automatically

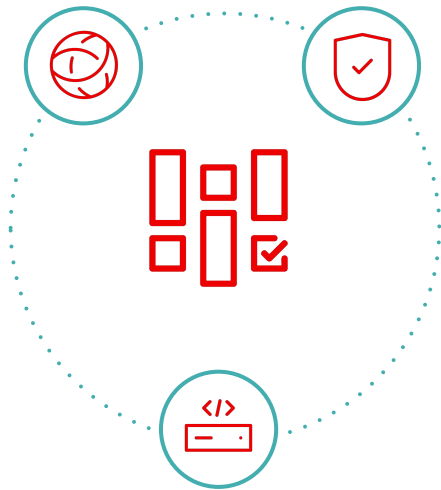


Command line

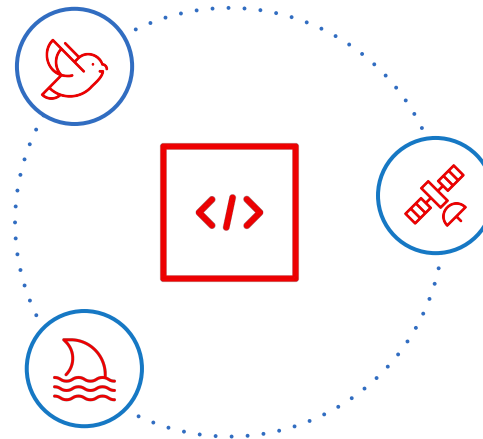
CLI access is also possible via `ansible-galaxy` command

Many technologies, different life cycles

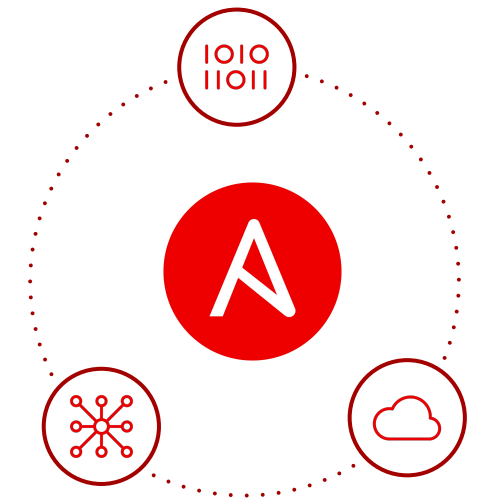
How to keep runtime environment, collections and dependencies aligned?



Collections



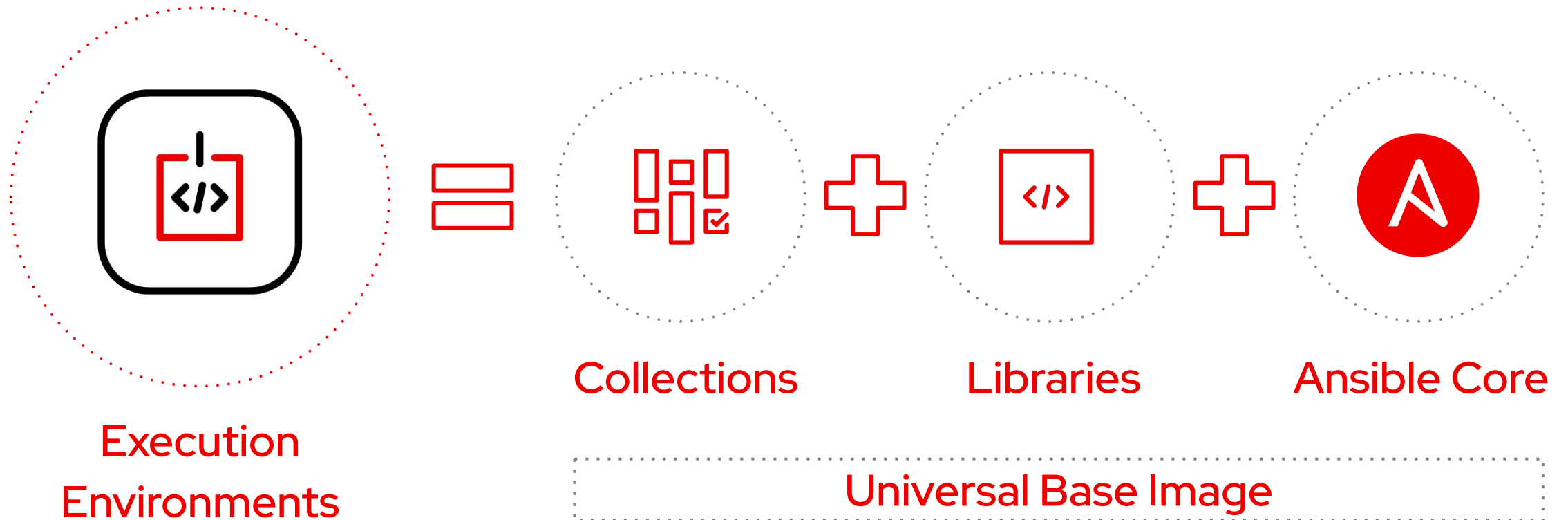
Dependencies



Runtime

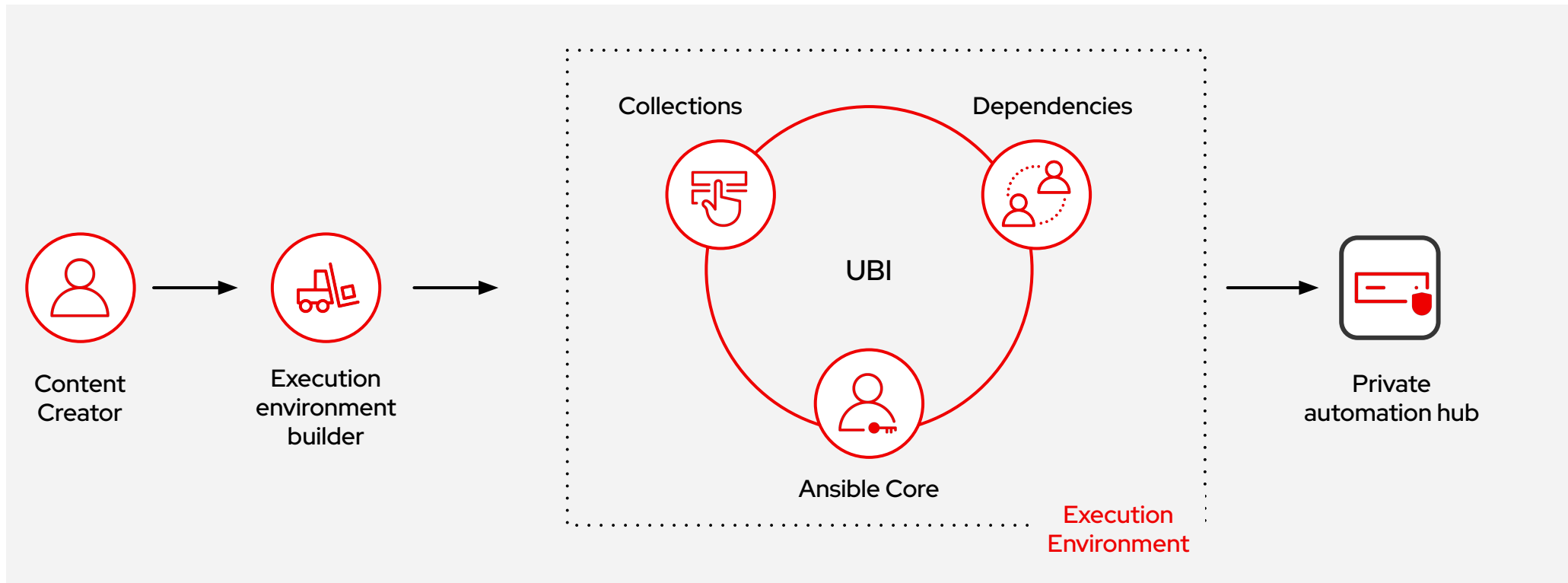
Automation Execution Environments

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



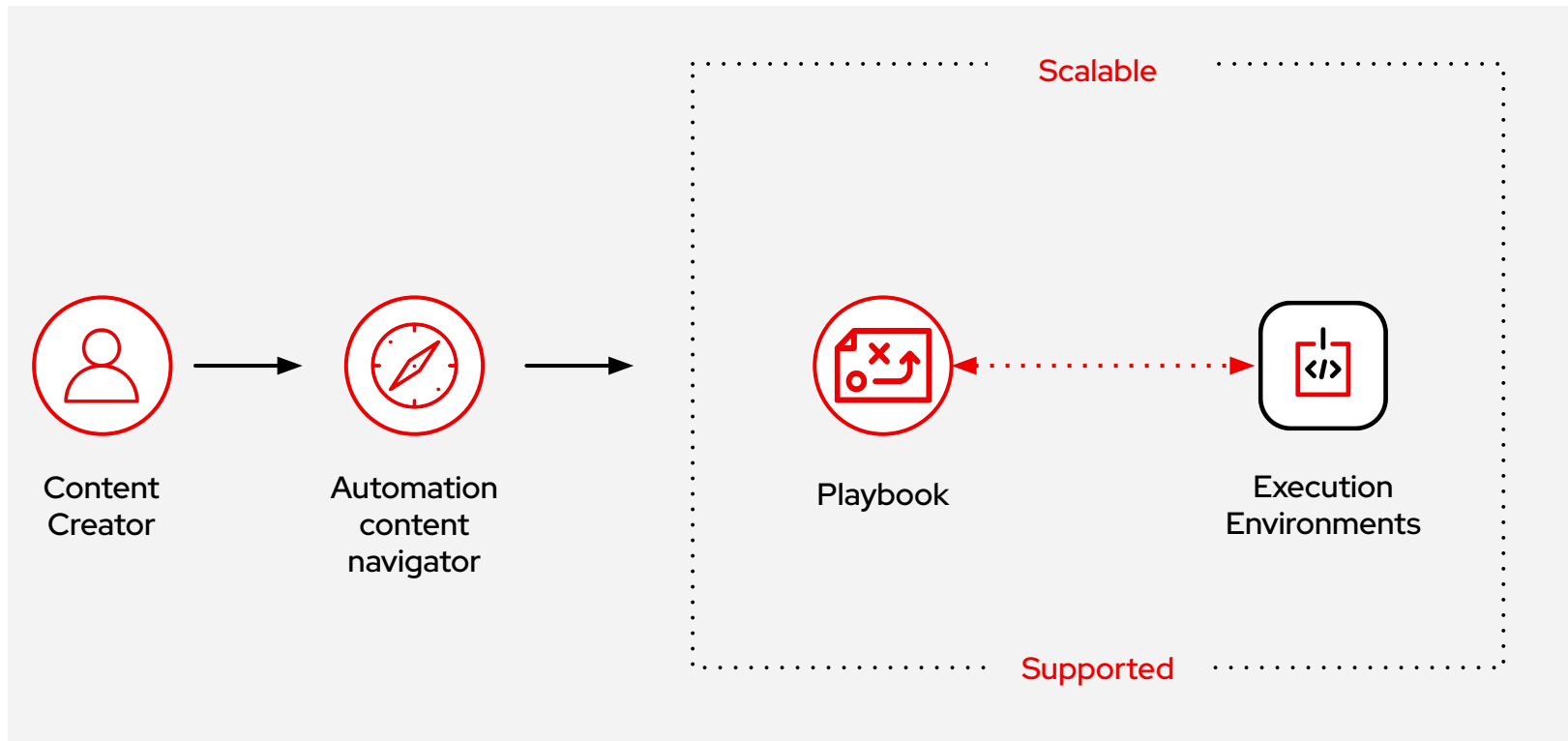
Build, create, publish

Development cycle of an automation execution environment



Develop, test, run

How to develop, test and run containerized Ansible content

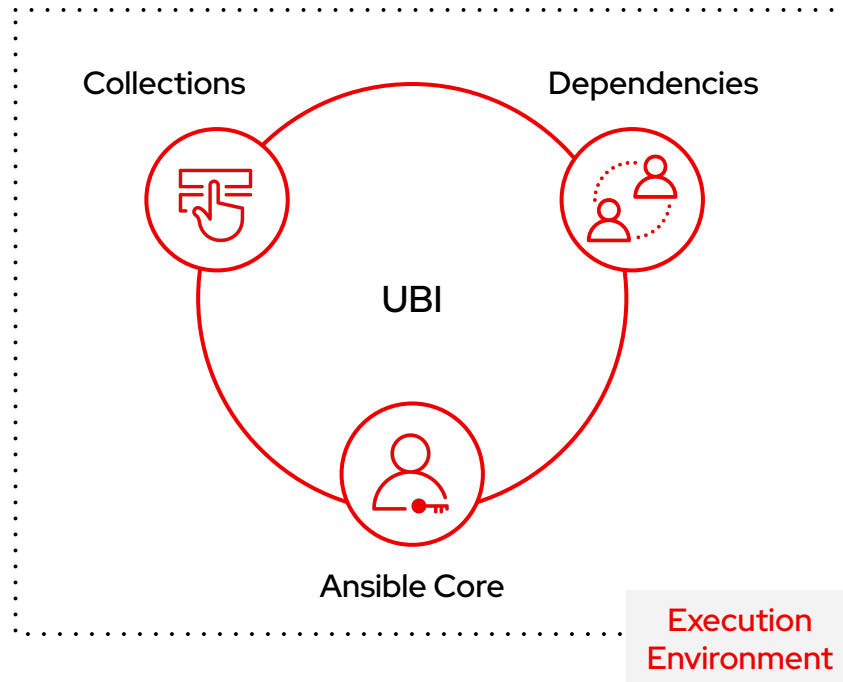


Deep Dive



Execution Environments

Common artifact between content creator and automation architect (operate)



- ▶ Container images that contain the following
 - RHEL UBI 8
 - Ansible 2.9 or Ansible Core 2.12
 - Python 3.8
 - Any content Collections
 - Collection python or binary dependencies.

Replaces python virtual environments



Execution Environments Provided with the Platform



Minimal

- Ansible
- Ansible runner
- Ansible Collections
 - Ansible Core

registry.redhat.io/ansible-automation-platform-21/ee-minimal-rhel8



Supported

- Ansible
- Ansible runner
- Ansible Collections
 - Ansible Core
 - Red Hat

Supported Collections

- Default EE for Ansible Controller
- registry.redhat.io/ansible-automation-platform-21/ee-supported-rhel8



Ansible 2.9

- Ansible 2.9
- Used for compatibility with playbooks and content written for previous versions of Ansible Automation Platform.

registry.redhat.io/ansible-automation-platform-21/ee-29-rhel8

New Command Lines Tools

Content Creators

```
Repo ID: ansible-automation-platform-2.1-for-rhel-8-x86_64-rpms  
Repo Name: Red Hat Ansible Automation Platform 2.1 for RHEL 8 x86_64 (RPMs)
```

Ansible Builder



- ▶ Automates the process of building automation execution environments
- ▶ Uses metadata defined in various Ansible Collections, as well as by the user.

```
dnf install ansible-builder
```

Ansible Navigator



Text-based user interface that allows you to:

- ▶ Launch and watch jobs and playbooks.
- ▶ Share stored, completed playbook and job run artifacts in JSON format.
- ▶ Browse and introspect automation execution environments.
- ▶ Browse your file-based inventory.
- ▶ Render Ansible module documentation and extract examples you can use in your playbooks.

```
dnf install ansible-navigator
```



Ansible Builder

execution-environment.yml

```
version: 1

build_arg_defaults:
  ANSIBLE_GALAXY_CLI_COLLECTION_OPTS: "-c"
  EE_BASE_IMAGE: registry.redhat.io/ansible-automation-platform-21/ee-minimal-rhel8
ansible_config: 'ansible.cfg'

dependencies:
  galaxy: requirements.yml
  python: requirements.txt
# system: bindep.txt

additional_build_steps:
  prepend: |
    RUN pip3 install --upgrade pip setuptools
  append:
    - RUN ls -la /etc
```

Potential Adjustment points for each ee

Ansible.cfg

- Set location for content
 - Enterprise
- Identify any unique plugins or other configuration items

Requirements.yml

- Identify specific collections to include

Requirements.txt

- Identify specific python modules that need to be included

Bindep.txt

- Identify system level dependencies



Ansible Builder

ansible.cfg

```
[galaxy]
server_list = rh-certified_repo,community_repo
```

```
[galaxy_server.rh-certified_repo]
url=https://aap-hub-ansible-automation-platform.apps.ocp-cluster.dumont-lab.lan/api/galaxy/content/rh-certified/
token=xxxxxxxxxxxxxxxx
```

Content locations - Enterprise-wide setting

```
[galaxy_server.community_repo]
url=https://aap-hub-ansible-automation-platform.apps.ocp-cluster.dumont-lab.lan/api/galaxy/content/community/
token=xxxxxxxxxxxxxxxx
```

```
#[public-galaxy]
#url=https://galaxy.ansible.com/
#token=35735056cbe9688c325b8efaf51b4536edc7d3ea
```

```
[inventory]
enable_plugins = kubernetes.core.k8s
```

A potential change area between ee



Ansible Builder

requirements.yml and requirements.txt

kubevirt-ee

requirements.yml

```
collections:
- name: redhat.openshift
- name: community.kubernetes
  version: 1.2.1
- name: kubernetes.core
- name: community.kubevirt
```

requirements.txt

```
jmespath
requests
ansible-pylibssh
kubernetes
openshift
```

lab-ee

requirements.yml

```
collections:
- name: redhat.rhel_system_roles
- name: redhat.insights
- name: community.general
- name: ansible.posix
- name: ansible.tower
- name: ansible.utils
- name: ansible.controller
- name: redhat.openshift
- name: community.kubernetes
- name: kubernetes.core
- name: amazon.aws
- name: community.aws
- name: pfsensible.core
```

requirements.txt

```
jmespath
ansible-pylibssh
kubernetes
openshift
boto3
botocore
```

In the project folder execute the following
#ansible-builder build -t <ee-name> -v 3



Ansible Navigator

Targeted to Content Creators

- ▶ Real value is to ensure that the playbook created works on my control node and in Ansible Controller
- ▶ Handoff point between Content Creators and Automation Architects
- ▶ Two modes
 - Stdout - look like ansible-playbook output
 - TUI
- ▶ Configuration file available (either json or yaml)
 - `\~/ansible-navigator.<ext>`
 - `./ansible-navigator.<ext>`
- ▶ Replay capability built in



Ansible-Navigator

Commands

```
bdumont@ansible:~/insights_api_project
0 ## Welcome
1 -----
2
3 Some things you can try from here:
4 - `:collections`           Explore available collections
5 - `:config`               Explore the current ansible configuration
6 - `:doc <plugin>`        Review documentation for a module or plugin
7 - `:help`                 Show the main help page
8 - `:images`              Explore execution environment images
9 - `:inventory -i <inventory>` Explore an inventory
10 - `:log`                  Review the application log
11 - `:open`                 Open current page in the editor
12 - `:replay`              Explore a previous run using a playbook artifact
13 - `:run <playbook> -i <inventory>` Run a playbook in interactive mode
14 - `:quit`                 Quit the application
15
16 happy automating,
17
18 -winston
```

`^f/PgUp` page up `^b/PgDn` page down `↑↓` scroll `esc` back `:help` help



Ansible-Navigator

Playbook Run

```
ansible-navigator --eei quay.io/bdumont/lab-ee run soe-rhel-vm.yml --vault-password-file=vault-pass
```

Execution Environment to use

Command

PLAY NAME	OK	CHANGED	UNREACHABLE	FAILED	SKIPPED	IGNORED	IN PROGRESS	TASK COUNT	PROGRESS
get machines to soe	22	14	0	0	2	0	0	24	COMPLETE

^f/PgUp page up ^b/PgDn page down ↑↓ scroll esc back [0-9] goto :help help SUCCESSFUL



Ansible-Navigator

Playbook Drill Down

```
bdumont@ansible:~/soe_project
```

RESULT	HOST	NUMBER	CHANGED	TASK	TASK ACTION	DURATION
OK	vm-2022-03-01-20-23-04-1.dumont-lab.lan	0	False	Gathering Facts	gather_facts	2s
OK	vm-2022-03-01-01-24-06-1.dumont-lab.lan	1	False	Gathering Facts	gather_facts	2s
OK	vm-2022-03-01-20-23-04-1.dumont-lab.lan	2	False	include_tasks	include_tasks	0s
OK	vm-2022-03-01-01-24-06-1.dumont-lab.lan	3	False	include_tasks	include_tasks	0s
OK	vm-2022-03-01-20-23-04-1.dumont-lab.lan	4	False	set activation key rhel 8 virtual	set_fact	0s
OK	vm-2022-03-01-01-24-06-1.dumont-lab.lan	5	False	set activation key rhel 8 virtual	set_fact	0s
SKIPPED	vm-2022-03-01-20-23-04-1.dumont-lab.lan	6	False	set activation key - rhel 7	set_fact	0s
SKIPPED	vm-2022-03-01-01-24-06-1.dumont-lab.lan	7	False	set activation key - rhel 7	set_fact	0s
OK	vm-2022-03-01-20-23-04-1.dumont-lab.lan	8	True	install cert rpm	yum	7s
OK	vm-2022-03-01-01-24-06-1.dumont-lab.lan	9	True	install cert rpm	yum	6s
OK	vm-2022-03-01-20-23-04-1.dumont-lab.lan	10	True	subscribe to sat	redhat_subscription	38s
OK	vm-2022-03-01-01-24-06-1.dumont-lab.lan	11	True	subscribe to sat	redhat_subscription	38s
OK	vm-2022-03-01-20-23-04-1.dumont-lab.lan	12	True	Disable all RHSM repositories	rhsm_repository	12s
OK	vm-2022-03-01-01-24-06-1.dumont-lab.lan	13	True	Disable all RHSM repositories	rhsm_repository	12s
OK	vm-2022-03-01-20-23-04-1.dumont-lab.lan	14	True	Enable RHSM repositories	rhsm_repository	10s
OK	vm-2022-03-01-01-24-06-1.dumont-lab.lan	15	True	Enable RHSM repositories	rhsm_repository	9s
OK	vm-2022-03-01-20-23-04-1.dumont-lab.lan	16	True	add foreman proxy ssh keys	authorized_key	1s
OK	vm-2022-03-01-01-24-06-1.dumont-lab.lan	17	True	add foreman proxy ssh keys	authorized_key	1s
OK	vm-2022-03-01-20-23-04-1.dumont-lab.lan	18	False	include_tasks	include_tasks	0s
OK	vm-2022-03-01-01-24-06-1.dumont-lab.lan	19	False	include_tasks	include_tasks	0s
OK	vm-2022-03-01-20-23-04-1.dumont-lab.lan	20	True	install packages	yum	30s
OK	vm-2022-03-01-01-24-06-1.dumont-lab.lan	21	True	install packages	yum	28s
OK	vm-2022-03-01-20-23-04-1.dumont-lab.lan	22	True	start firewalld	service	2s
OK	vm-2022-03-01-01-24-06-1.dumont-lab.lan	23	True	start firewalld	service	2s

```
^f/PgUp page up    ^b/PgDn page down    ↑↓ scroll    esc back    [0-9] goto    :help help    SUCCESSFUL
```



Ansible Navigator

Task Drill Down

```
bdumont@ansible:~/soe_project
PLAY [get machines to soe:4] *****
TASK [set activation key rhel 8 virtual] *****
OK: [vm-2022-03-01-20-23-04-1.dumont-lab.lan]
0 ---
1 duration: 0.044096
2 end: '2022-03-01T21:06:14.705156'
3 event_loop: null
4 host: vm-2022-03-01-20-23-04-1.dumont-lab.lan
5 play: get machines to soe
6 play_pattern: kurtvirt_hosts
7 playbook: /home/bdumont/soe_project/soe-rhel-vm.yml
8 remote_addr: vm-2022-03-01-20-23-04-1.dumont-lab.lan
9 res:
10  _ansible_no_log: false
11  ansible_facts:
12    subscribe_activation_key: rhel8-vm-ak
13    changed: false
14  start: '2022-03-01T21:06:14.661060'
15  task: set activation key rhel 8 virtual
16  task_action: set_fact
17  task_args: ''
18  task_path: /home/bdumont/soe_project/tasks/subscribe_systems.yml:1

^f/PgUp page up  ^b/PgDn page down  ^↕ scroll  esc back  - previous  + next  [0-9] goto  :help help  SUCCESSFUL
```



Ansible Navigator

~/.ansible-navigator.yml

```
---
ansible-navigator:
  execution-environment:
    image: quay.io/bdumont/lab-ee:latest
  logging:
    #   append: False
    level: critical
    #   file: /tmp/log.txt
    #   mode: stdout
  playbook-artifact:
    enable: True
#   replay: {playbook_dir}/{playbook_name}-artifact-{ts_utc}.json
    save-as: artifacts/{playbook_name}-artifact-{ts_utc}.json
```

Demo Time

Thank you

 [linkedin.com/company/red-hat](https://www.linkedin.com/company/red-hat)

 [youtube.com/user/RedHatVideos](https://www.youtube.com/user/RedHatVideos)

 [facebook.com/redhatinc](https://www.facebook.com/redhatinc)

 twitter.com/RedHat

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 that execute on different hosts.

```
---  
- name: install and start apache  
  hosts: web  
  become: yes
```


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.

```
- name: latest index.html file ...  
  template:  
    src: files/index.html  
    dest: /var/www/html/
```

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.

Example become plugin:

```
---  
- name: install and start apache  
  hosts: web  
  become: yes
```

Example filter plugins:

```
{{ some_variable | to_nice_json }}  
{{ some_variable | to_nice_yaml }}
```

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.

```
---  
- name: install and start apache  
  hosts: web  
  roles:  
    - common  
    - webserver
```