



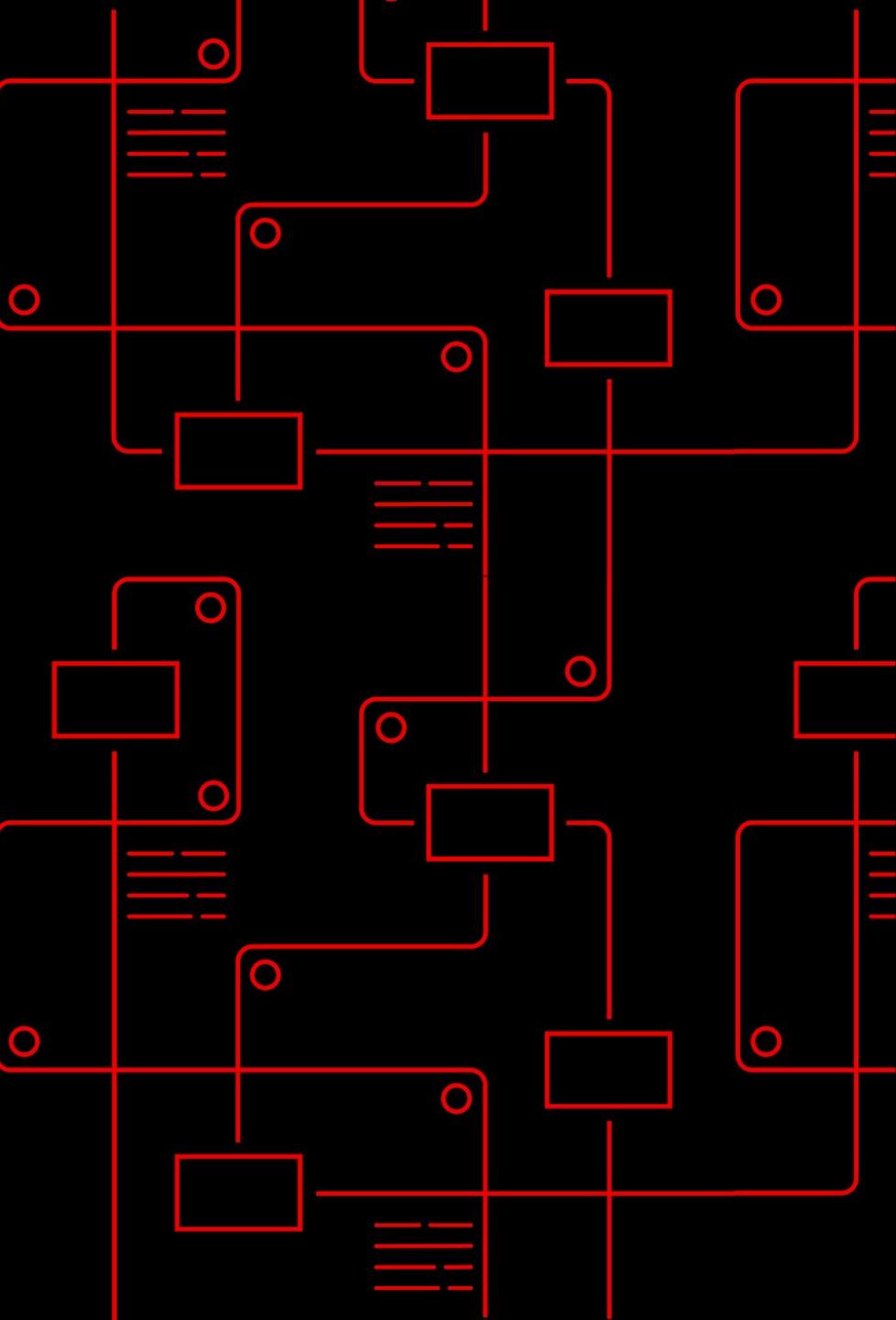
Red Hat Ansible Automation Platform

Developer Tools

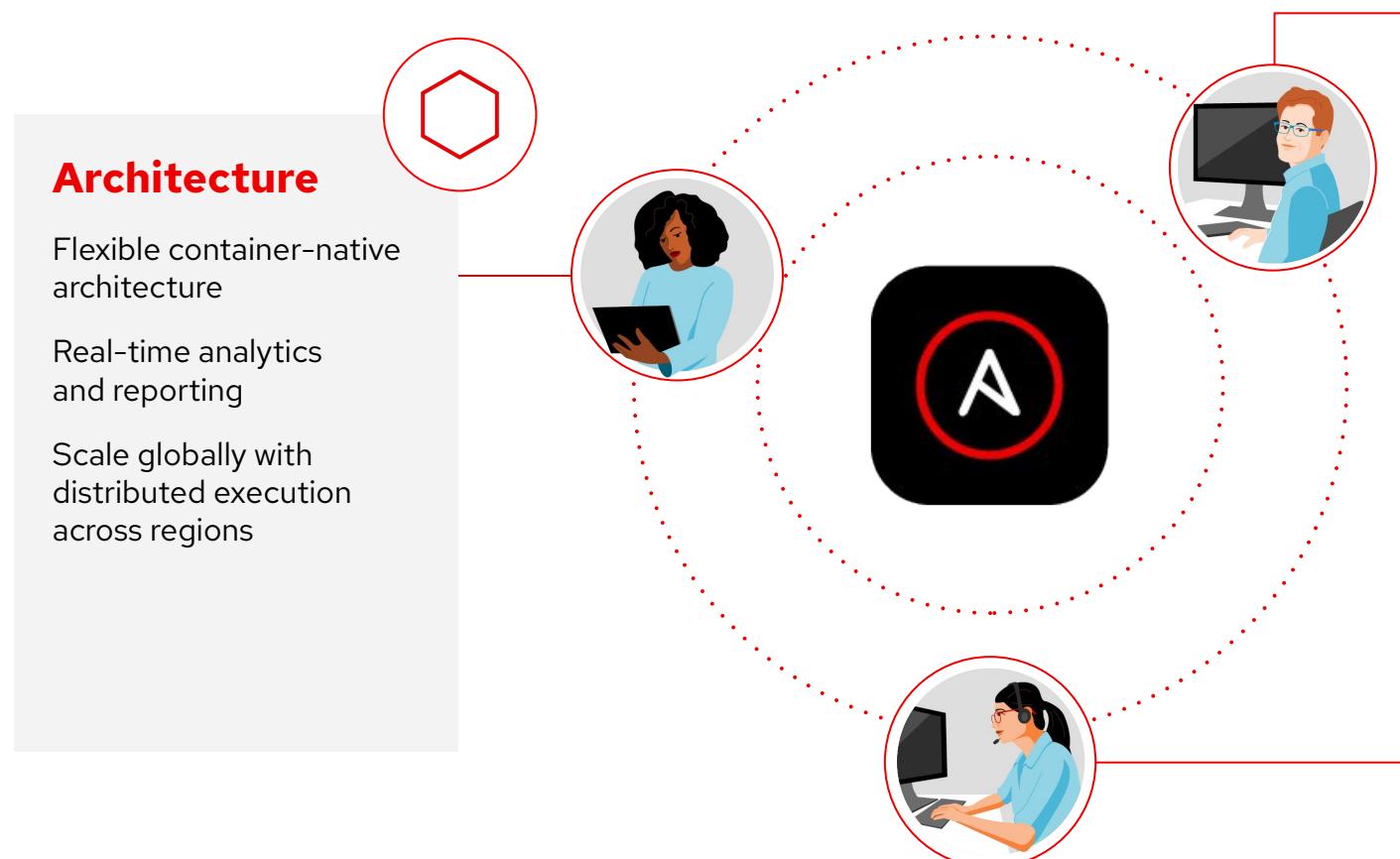
Brad Krumme
Senior Specialist Solution Architect - Automation



What is Red Hat® Ansible® Automation Platform?



A platform for the **entire automation team**.



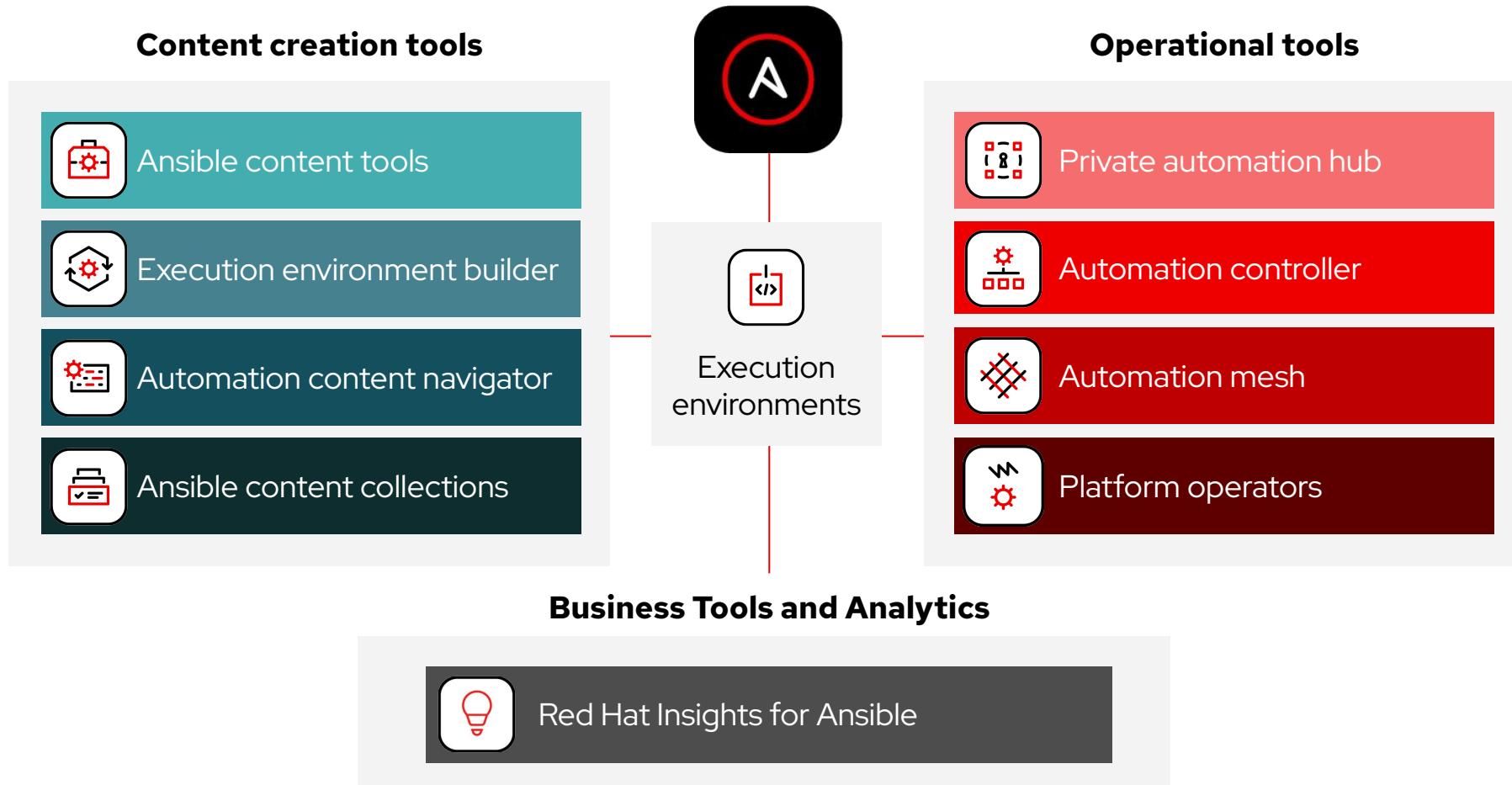
Content creation

Content creation tooling
Portable distribution and reliable execution
Large ecosystem of certified automation

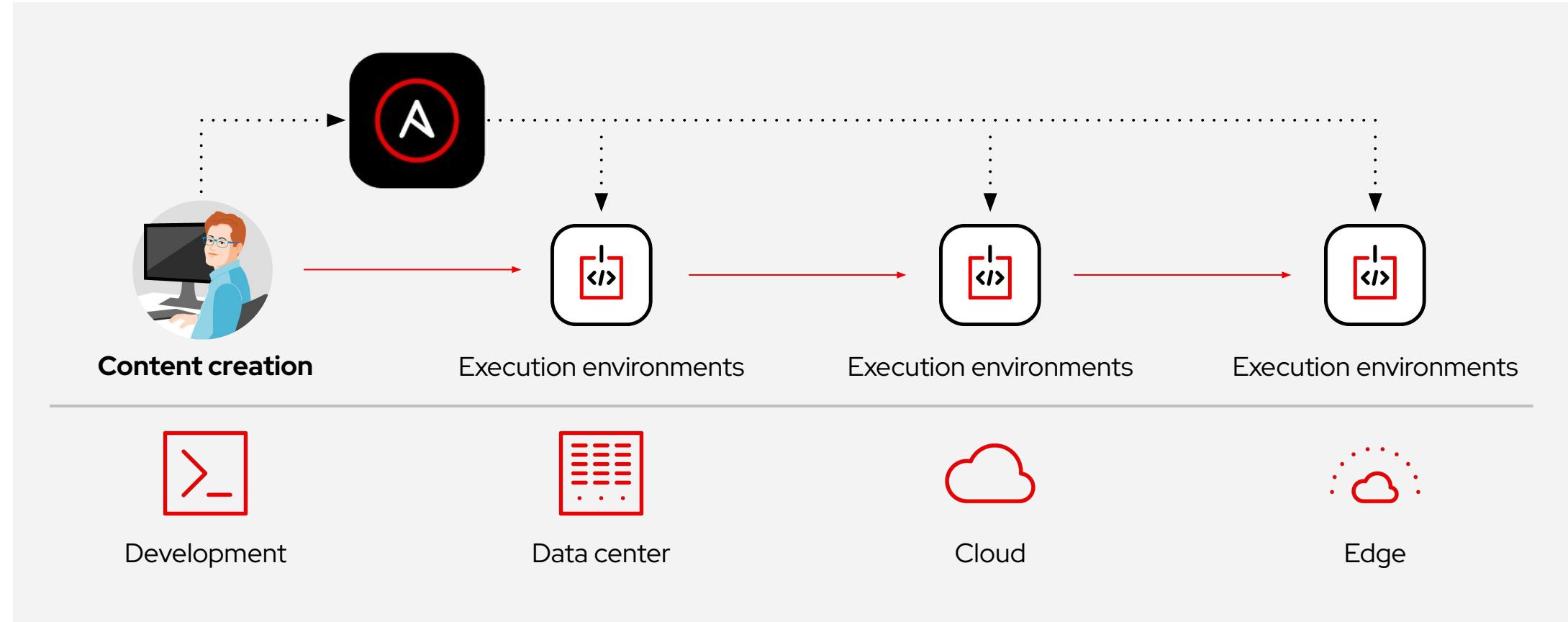
Operations

Enterprise features:
WebUI, API, role-based access control (RBAC), auditing and workflows for managing at scale
Hosted and private content management solutions
Integrates with your environment

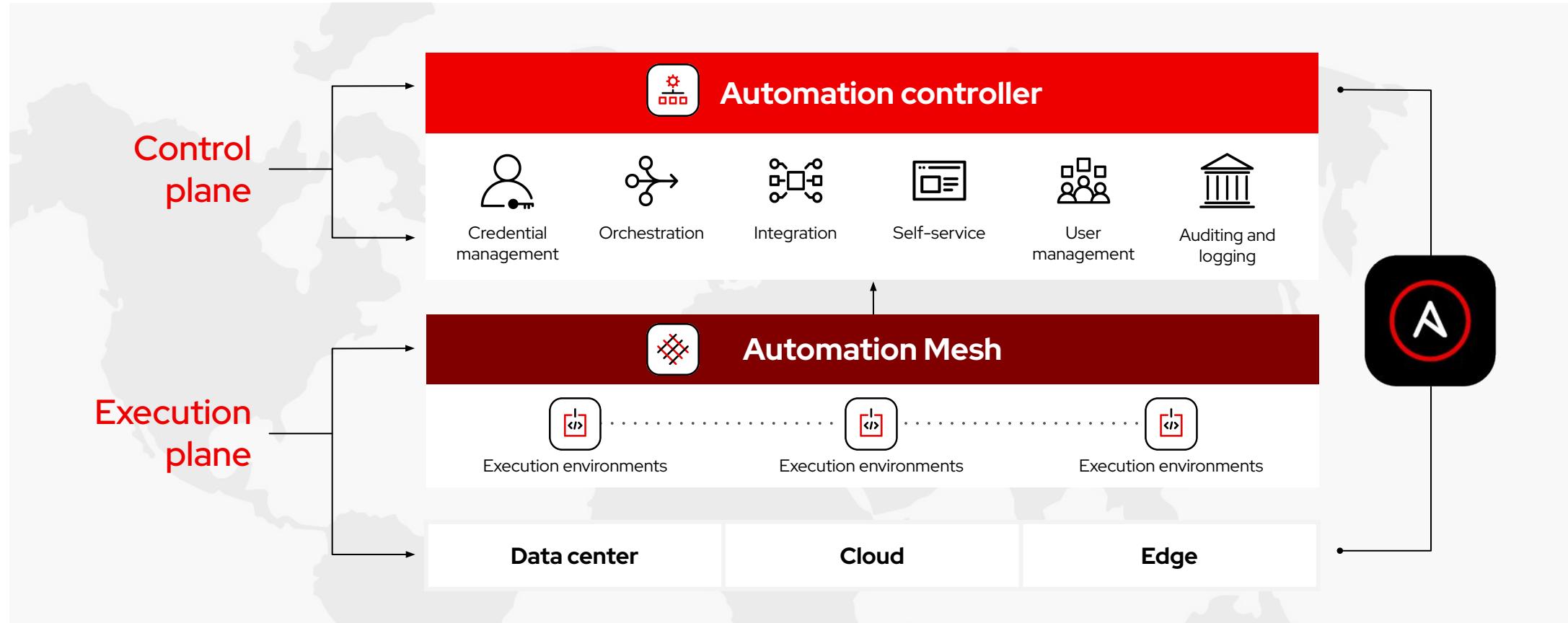
An integrated solution for the enterprise.



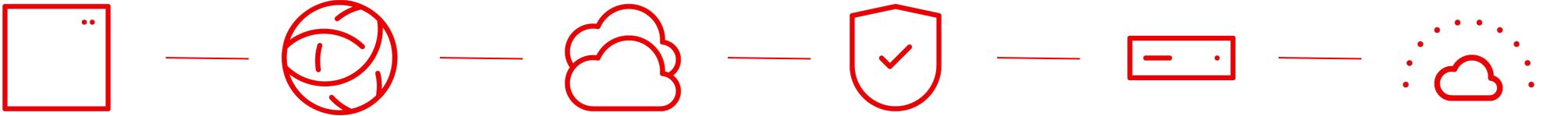
Built for consistency. **Portability** is reliability.



A distributed architecture built for scale.



The capabilities you need across your IT footprint.



Applications

- ▶ DevOps
- ▶ CI/CD
- ▶ GitOps

Network

- ▶ Configuration management
- ▶ Infrastructure awareness
- ▶ Network validation

Cloud

- ▶ Orchestration
- ▶ Operationalisation
- ▶ Governance

Security

- ▶ Investigation enrichment
- ▶ Threat hunting
- ▶ Incident response

Infrastructure

- ▶ Deployment
- ▶ Provisioning
- ▶ Management

Edge

- ▶ Extend security
- ▶ Scalability
- ▶ Interoperability

Enabling your automation team **to consistently...**



Create

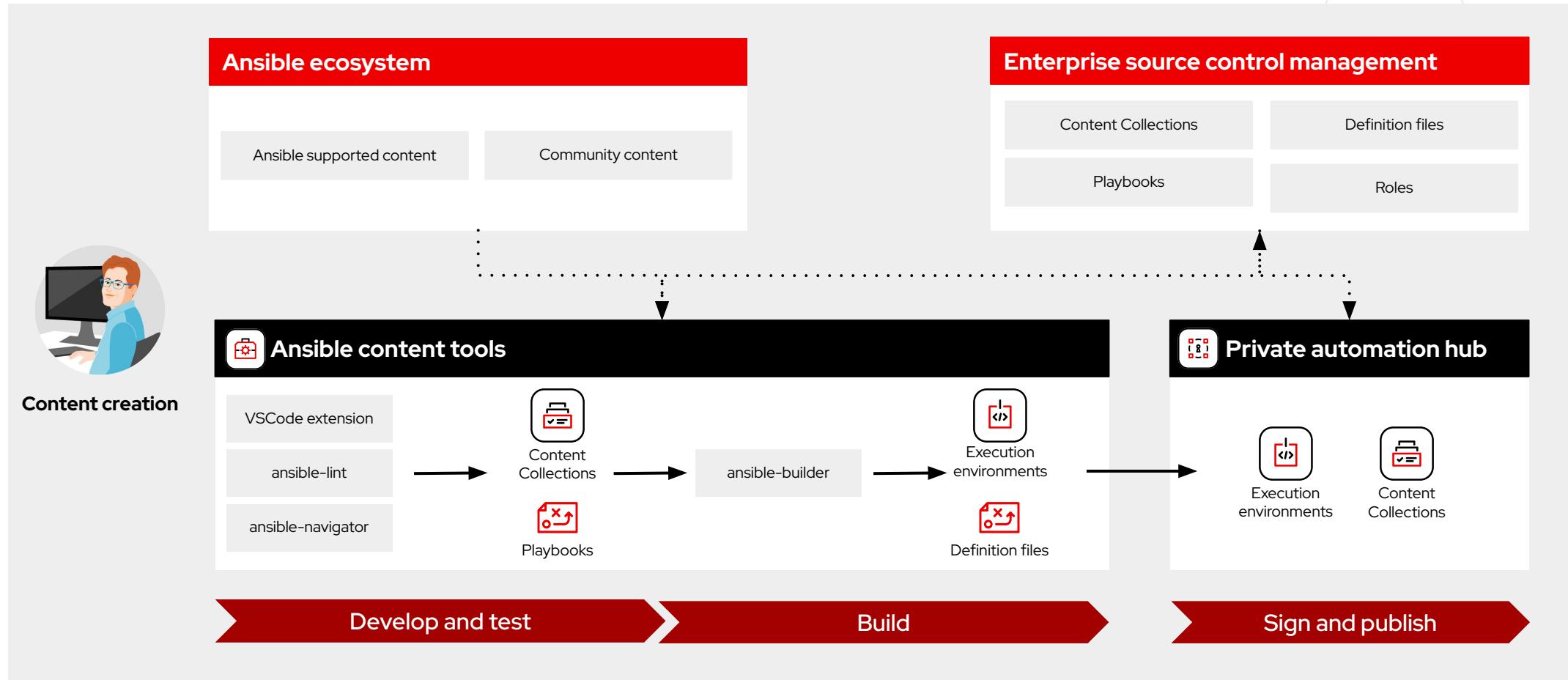
Manage

Scale

Create



The automation content life cycle. **Create.**



Ansible Core (`ansible-core`)



What is it?

- ▶ The main building block for Ansible
- ▶ Simple YAML syntax to develop Ansible Playbooks
- ▶ Provides CLI tools to develop, test and run playbooks
- ▶ Pluggable architecture that allows extensions through Content Collections

```
---  
- name: Shutdown VM guest  
hosts: localhost  
gather_facts: false  
tasks:  
  - name: Turn off specified VM guest  
    vmware.vmware_rest.vcenter_vm_guest_power:  
      state: shutdown  
      vm: 1021343  
      vcenter_hostname: vcenter.demoredhat.com  
      vcenter_username: admin  
      vcenter_password: tedlasso
```

Ansible playbooks

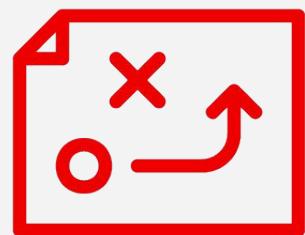
```
---
- name: Install and start apache
  hosts: web
  become: true

  tasks:
    - name: Ensure the httpd package is installed
      ansible.builtin.yum:
        name: httpd
        state: present

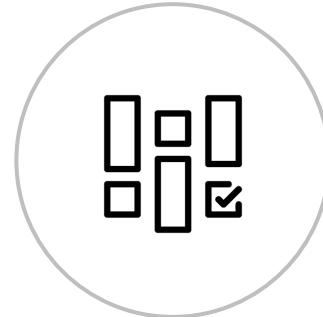
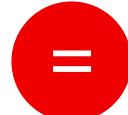
    - name: Create the index.html file
      ansible.builtin.template:
        src: files/index.html
        dest: /var/www/html/

    - name: Start the httpd service if needed
      ansible.builtin.service:
        name: httpd
        state: started
```

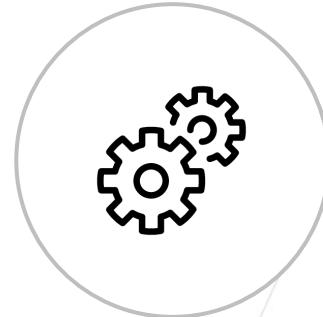
What makes up an Ansible playbook?



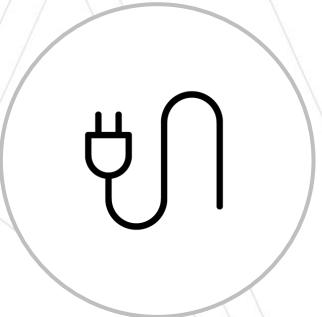
Playbook



Plays



Modules



Plugins

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

```
...  
- name: Ensure the httpd package is installed  
  hosts: web  
  become: true
```

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 created in Python, or Powershell for Windows setups, but can be developed in any language

A dark-themed terminal window showing a single line of Ansible YAML code. The code defines a task to create an index.html file using the 'template' module.

```
- name: Create the index.html file
  ansible.builtin.template:
    src: files/index.html
    dest: /var/www/html/log
```



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: true
```

Example filter plugins:

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

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



Ansible Roles. Reusable automation actions.

What are they?

- ▶ Group 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

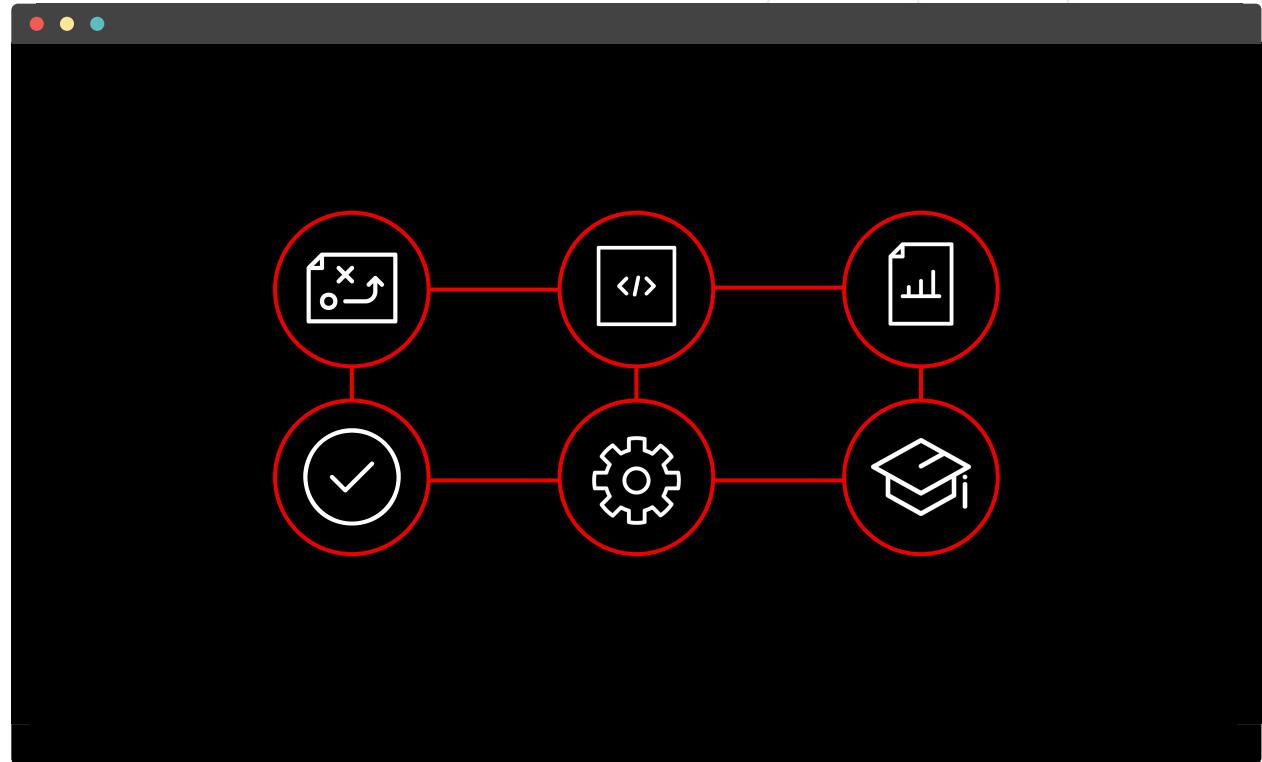
```
...  
- name: Install and start apache  
hosts: web  
ansible.builtin.roles:  
  - common  
  - webservers
```

Content Collections.

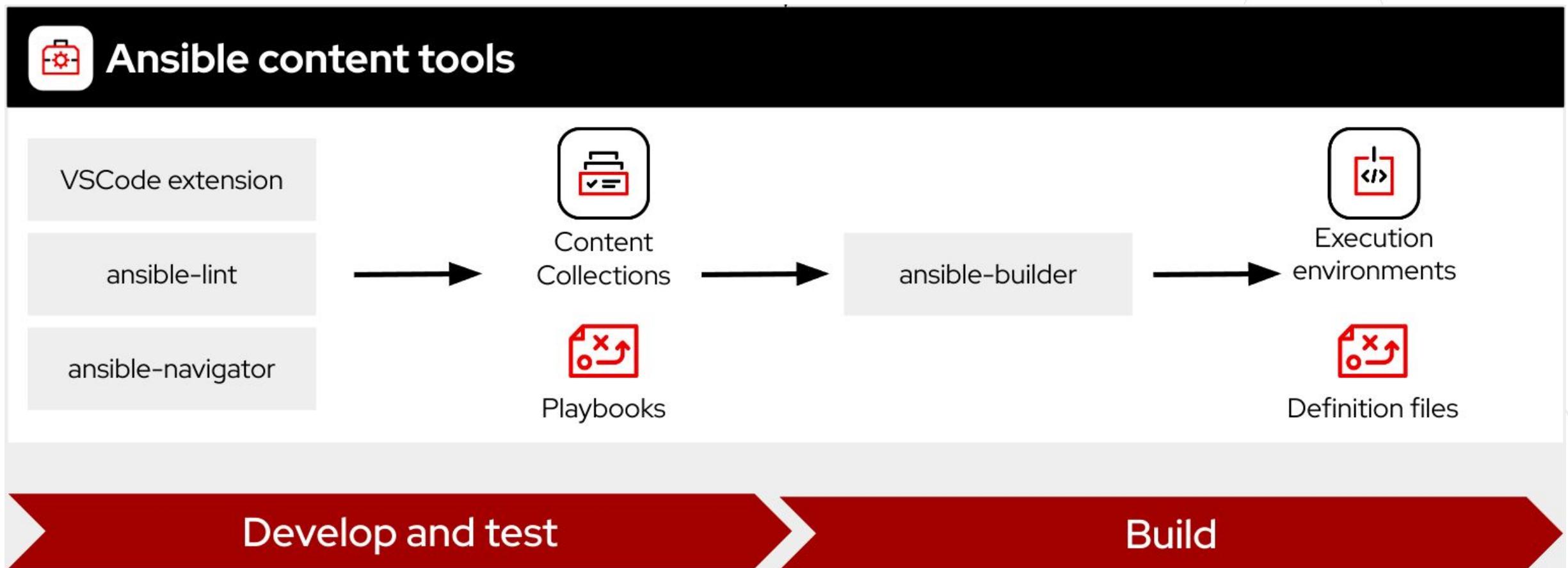
Simplified, consistent content delivery.

What are they?

- ▶ Group 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



Ansible content tools: Workflow



Ansible VS Code extension

Simplifying content creation



What is it?

- ▶ Syntax highlighting of keywords such as module names
- ▶ Live validation of your code while you type
- ▶ Integration with ansible-lint*
- ▶ Autocompletion on play, block or task contents etc
- ▶ Documentation references as you code

The screenshot shows a code editor window with the following snippet:

```
- name: query incident number and creation time
  set_fact:
    incident_list: '{{ incident_list + [{"number": item.number, "opened_at": item.opened_at}] }}'
  loop: "{{ incidents.records }}"
  when: incidents

- name: Create a problem from incident
  problem
```

A tooltip is displayed over the word "problem", listing available completions:

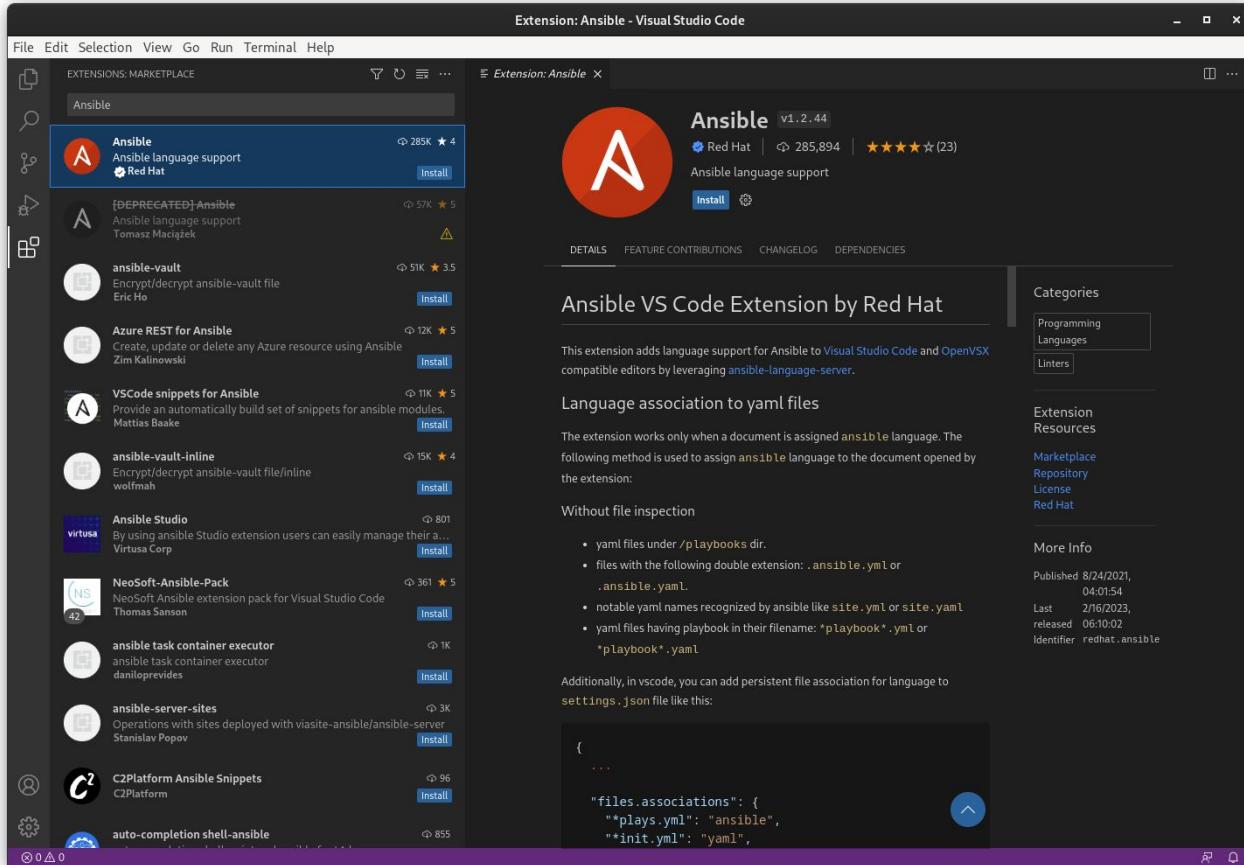
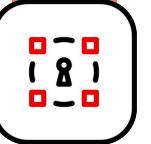
- servicenow.itsm.problem
- servicenow.itsm.problem_info
- servicenow.itsm.problem_task
- servicenow.itsm.problem_task_info
- ansible.builtin.proxmox
- ansible.builtin.portage
- ansible.builtin.proxmox_kvm
- ansible.builtin.profitbricks
- ansible.builtin.portinstall
- ansible.builtin.proxysql_backend_servers
- ansible.builtin.profitbricks_nic
- ansible.builtin.proxmox_template

The tooltip also includes a "Description" section with the following text:

- Create, delete or update a ServiceNow problem.
- For more information, refer to the ServiceNow problem management documentation at https://docs.servicenow.com/bundle/paris-it-service-management/page/product/problem-management/concept/c_ProblemManagement.html.

Ansible VS Code extension

Install from VS Code Extensions Menu



Ansible lint (ansible-lint)

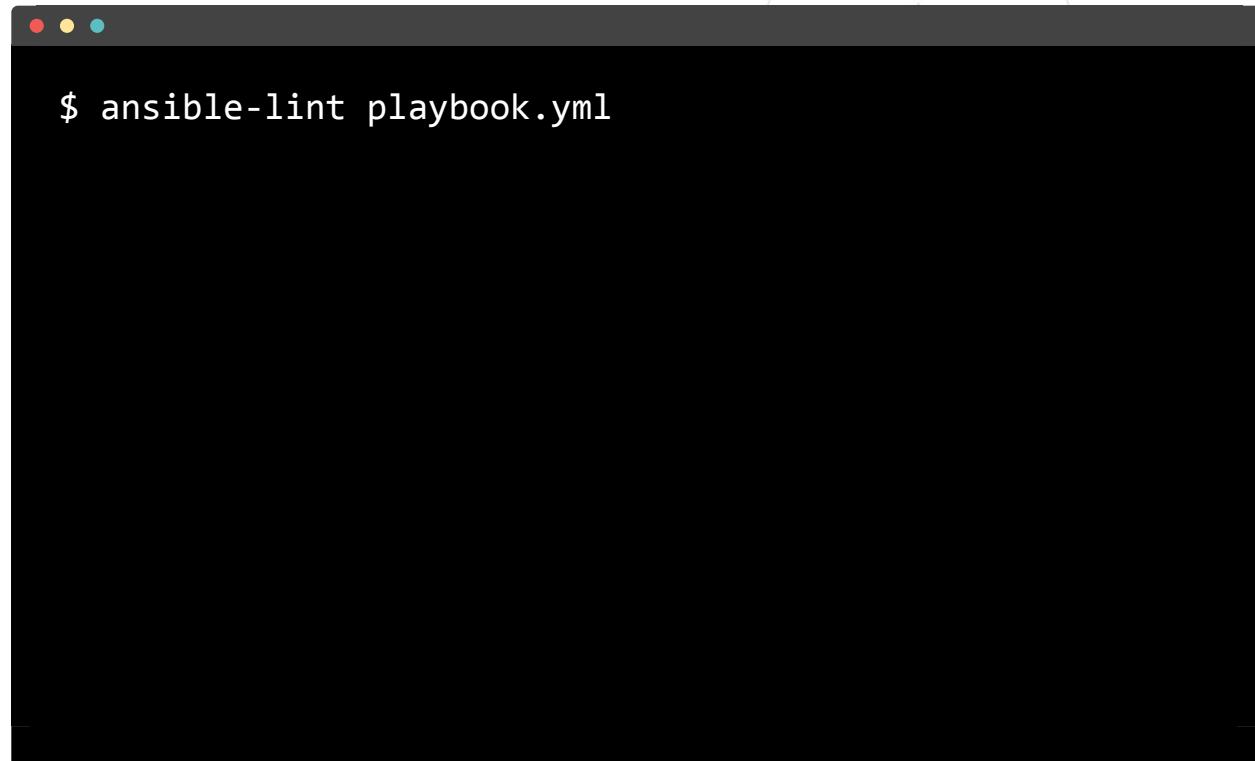
Fully supported (AAP 2.3)



What is it?

Command-line tool for linting playbooks, roles and collections aimed towards any Ansible users.

- ▶ Promote best practices and patterns.
- ▶ Develop consistent code across teams and scale using an opinionated strategy.
- ▶ Integrate into larger development workflows and CI tools.
- ▶ Helps upgrade playbooks to later Ansible Core versions.



A screenshot of a terminal window with a dark background. At the top, there are three small colored dots (red, yellow, blue). Below them, the command '\$ ansible-lint playbook.yml' is typed in white text. The rest of the terminal window is blank, showing a black background.

Ansible lint (ansible-lint).

Installation paradigms



```
# For installation on Red Hat Enterprise Linux  
(Requires Ansible Automation Platform Subscription)
```

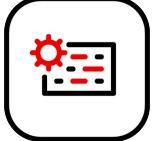
```
$ sudo dnf -y install ansible-lint
```

OR

```
# For installation on other linux systems  
(Installation from upstream)
```

```
$ python3 -m pip install ansible-lint
```

Ansible content navigator (ansible-navigator).



What is it?

It is a command line utility and text-based user interface (TUI) for running, testing and developing Ansible automation content

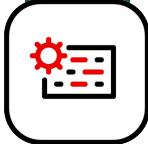
- ▶ Review EEs
- ▶ Develop collections
- ▶ Develop playbooks
- ▶ Troubleshoot problems

A screenshot of a terminal window with a black background and light gray text. The window has three colored dots (red, yellow, green) in the top-left corner. The text inside the terminal is:

```
$ ansible-navigator run playbook.yml -i inventory.ini
```

Ansible content navigator (ansible-navigator).

Installation paradigms



```
# For installation on Red Hat Enterprise Linux  
(Requires Ansible Automation Platform Subscription)
```

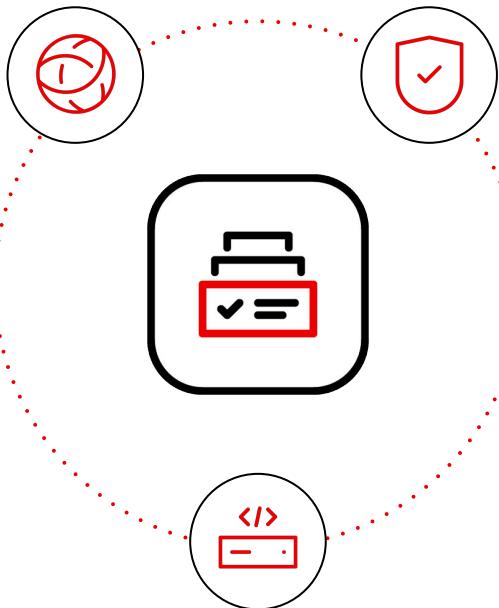
```
$ sudo dnf -y install ansible-navigator
```

OR

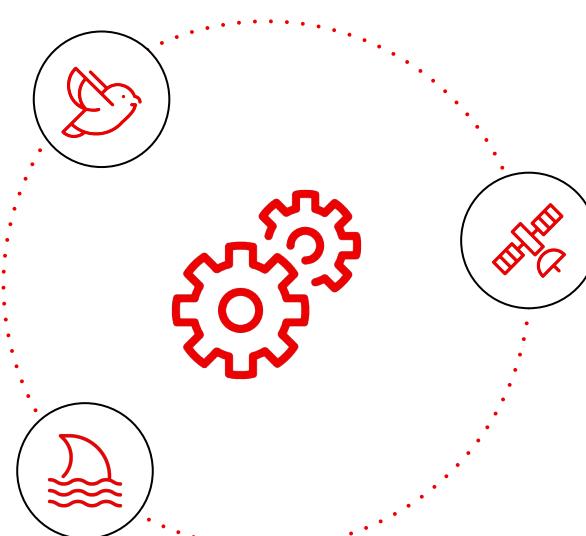
```
# For installation on other linux systems  
(Installation from upstream)
```

```
$ python3 -m pip install ansible-navigator
```

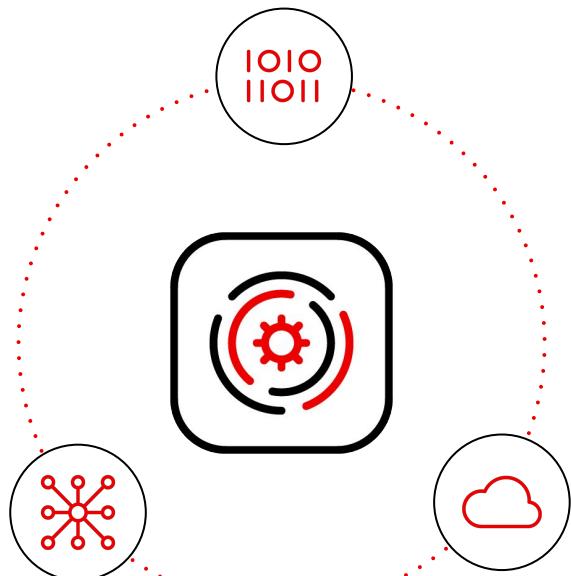
Many technologies, different life cycles. How do I keep it all aligned?



Collections

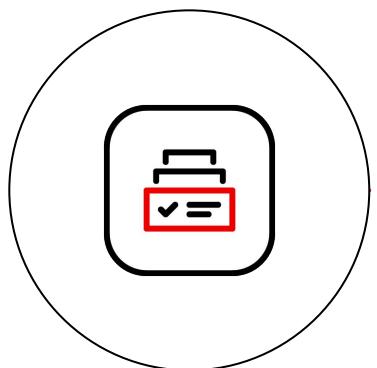


Dependencies

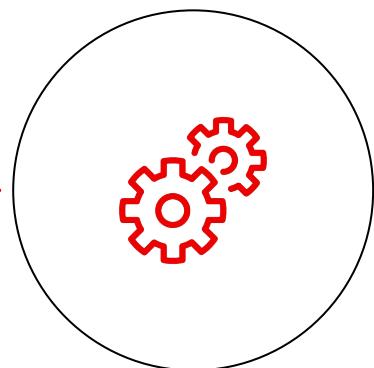


Runtime

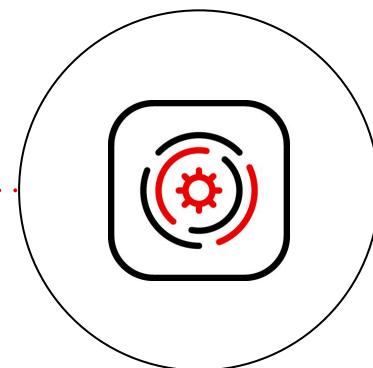
Automation execution environments. Reuse and scale automation content.



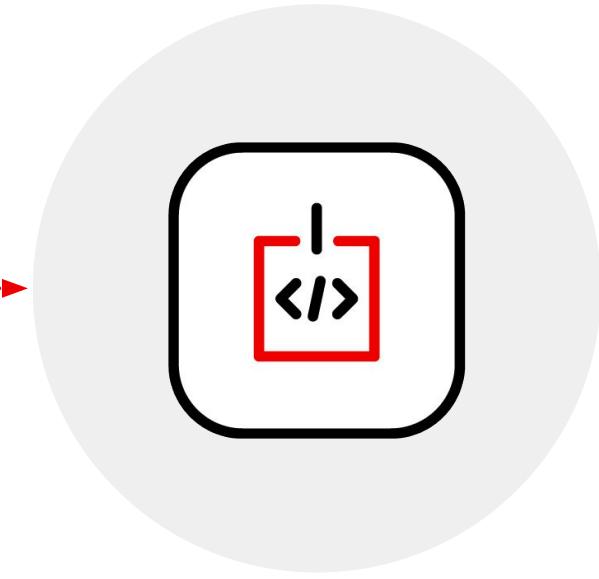
Collections



Libraries



Ansible Core

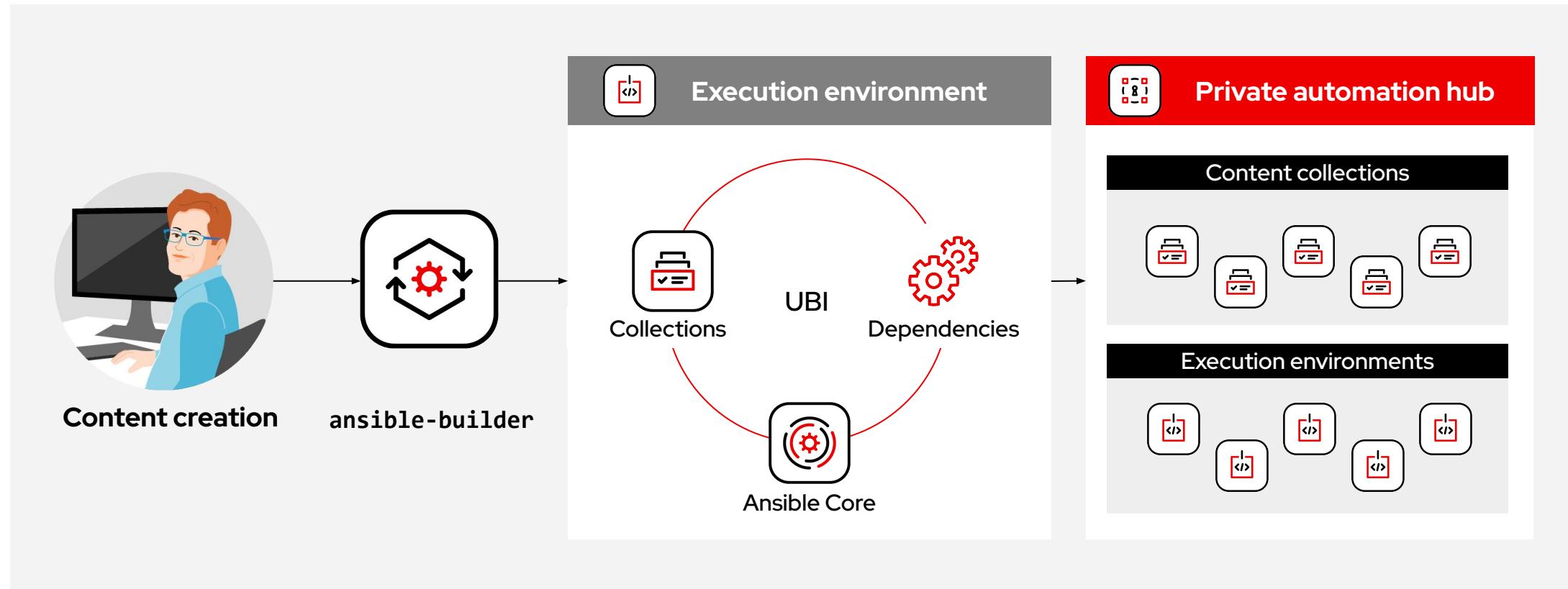


Automation
execution environments

Red Hat Universal Base Image (UBI)

Execution environment development.

Build, collaborate, sign, publish.





Execution environment builder. **Build.**

ansible-builder

What is it?

- ▶ Easily build custom execution environments with the exact Ansible content needed
- ▶ Manage, track and version execution environments
- ▶ Share execution environment build artifacts with other teams

A black terminal window with a dark gray header bar. In the header bar, there are three small colored dots (red, yellow, blue). Below the header, the text '\$ ansible-builder build --tag repo/custom_ee:latest' is displayed in white. The rest of the terminal window is black, indicating a blank or unprinted area.



Execution environment builder. **Build.**

Installation paradigms

```
# For installation on Red Hat Enterprise Linux  
(Requires Ansible Automation Platform Subscription)
```

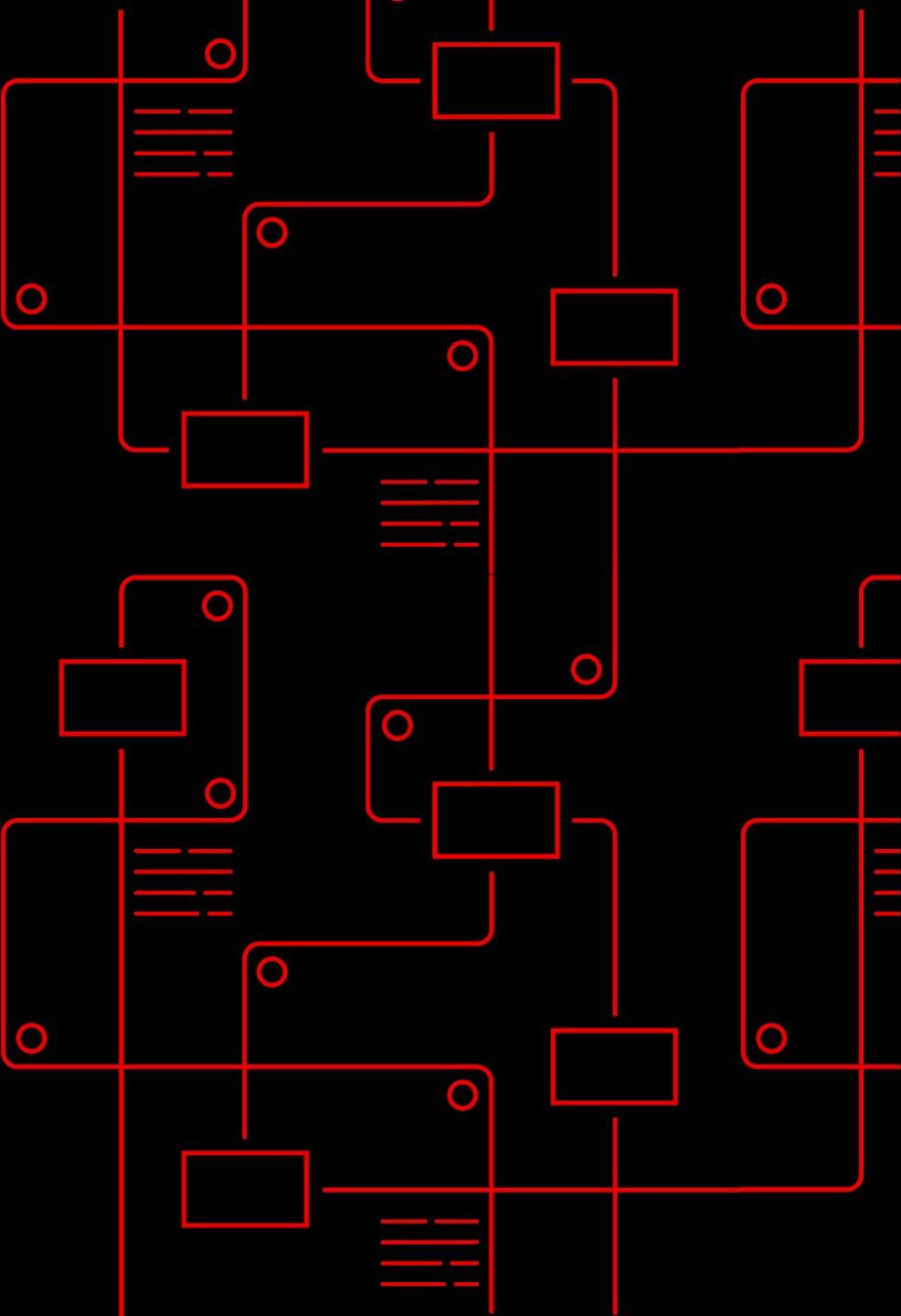
```
$ sudo dnf -y install ansible-builder
```

OR

```
# For installation on other linux systems  
(Installation from upstream)
```

```
$ python3 -m pip install ansible-builder
```

Demo Time





Ansible Lightspeed

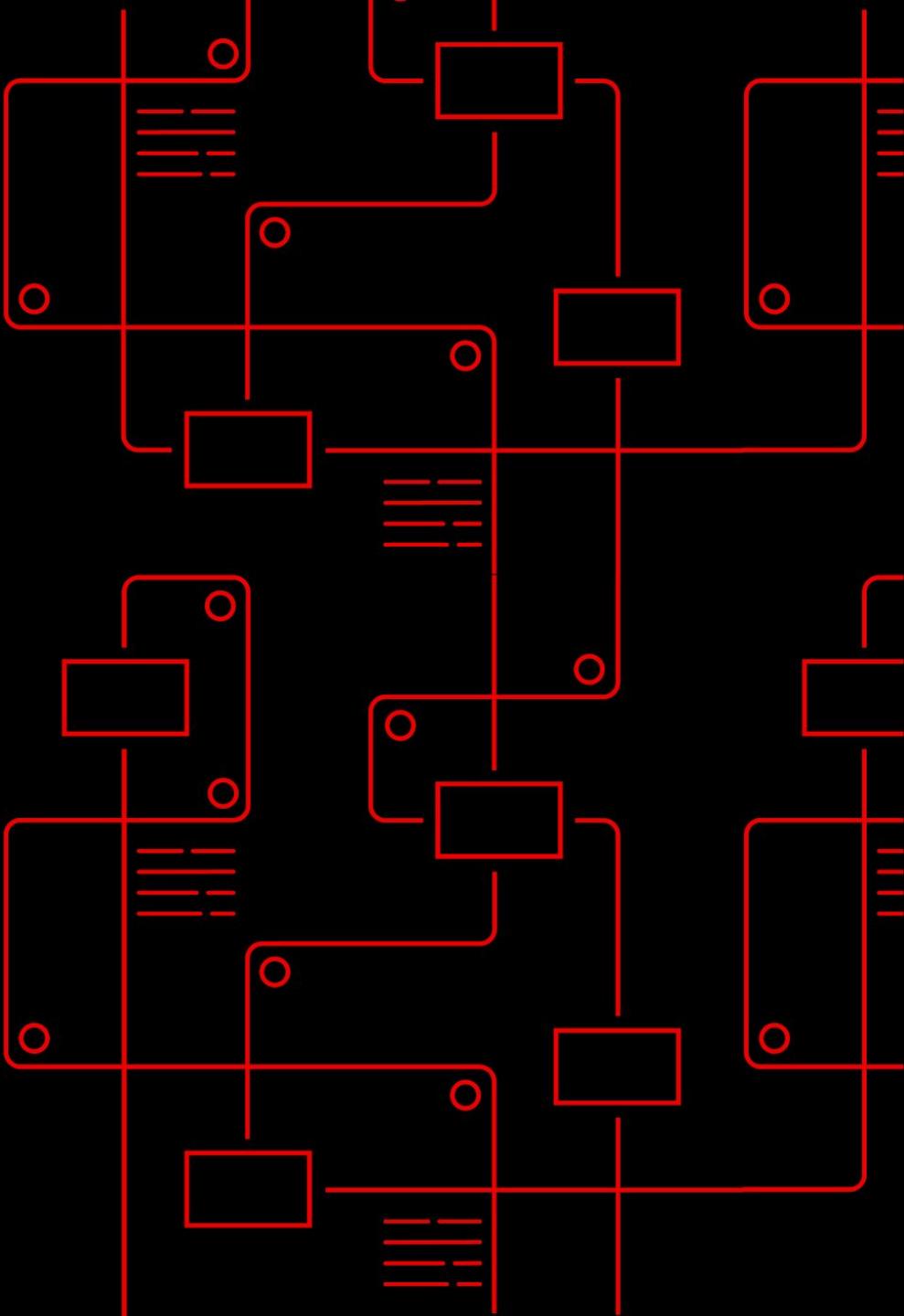
With IBM Watson Code Assistant



The experience



with IBM Watson Code Assistant



It starts with Ansible Playbooks

Ansible Lightspeed will eventually impact the Ansible experience in a **number of profound ways**.

But at the start, the experience will be focused on the very foundation of it all: **Ansible Playbooks**.

```
---
- name: Apache server installed
  hosts: web
  become: true
  tasks:
    - name: latest Apache version installed
      yum:
        name: httpd
        state: latest

    - name: Apache enabled and running
      service:
        name: httpd
        enabled: true
        state: started

    - name: copy index.html
      copy:
        src: web.html
        dest: /var/www/html/index.html
```

Ansible Lightspeed with IBM Watson Code Assistant

Ansible Lightspeed with IBM Watson Code

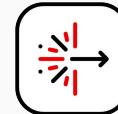
Assistant is a generative AI service accessed via the Ansible VSCode extension, allowing users to accept and run recommended code directly in their code editing environment while creating Ansible Playbooks.

A *Tech Preview* for the service will be available for all Ansible users in late June, with a commercial offering to follow this fall.

The **IBM Watson Code Assistant** integration is infused with IBM's Ansible foundation model. This foundation model combines Ansible Galaxy data and Red Hat subject matter expertise to deliver highly relevant code automation recommendations that adhere to Ansible best practices.

IBM Watson Code Assistant is built on the **Red Hat OpenShift Data Science** platform.

Ansible Lightspeed with IBM Watson Code Assistant



IBM Watson Code Assistant



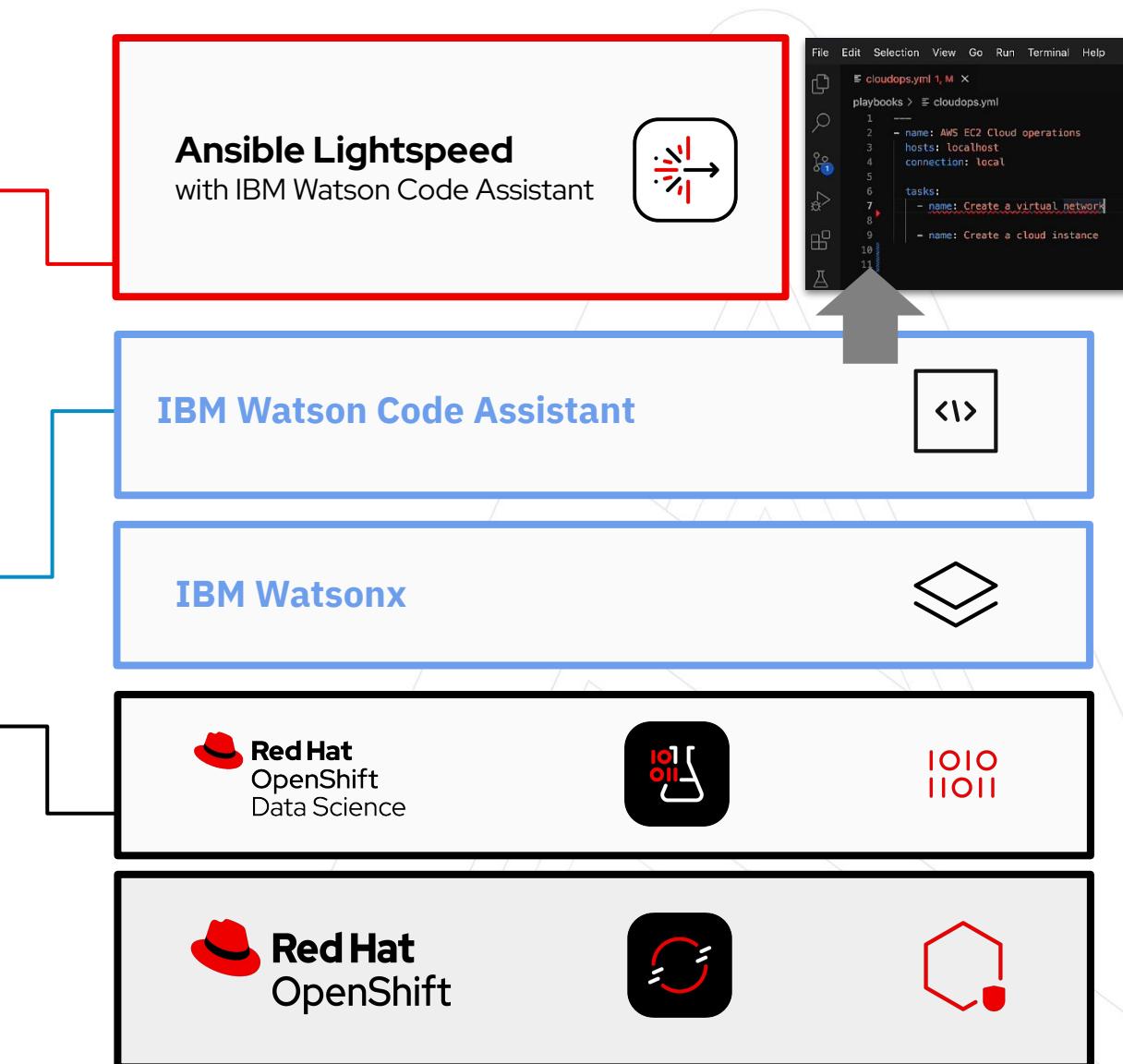
IBM Watsonx



Red Hat
OpenShift
Data Science



Red Hat
OpenShift



The Ansible Lightspeed experience

Enhancing Playbook creation

```
File Edit Selection View Go Run Terminal Help  
cludops.yml 1, M ×  
playbooks > cludops.yml  
1 ---  
2 - name: AWS EC2 Cloud operations  
3 hosts: localhost  
4 connection: local  
5  
6 tasks:  
7   - name: Create a virtual network  
8   - name: Create a cloud instance  
9  
10  
11  
12
```

1. Ansible Lightspeed with IBM Watson Code Assistant is **accessible via VSCode extension**

2. Type in a task directly into the VSCode editor. Ansible Lightspeed takes over.

```
File Edit Selection View Go Run Terminal Help  
cludops.yml 1, M ×  
playbooks > cludops.yml  
1 ---  
2 - name: AWS EC2 Cloud operations  
3 hosts: localhost  
4 connection: local  
5  
6 tasks:  
7   - name: Create a virtual network  
8     ec2_vpc_net:  
9       name: "{{ vpc_name }}"  
10      cidr_block: "{{ vpc_cidr }}"  
11      region: "{{ region }}"  
12      state: present  
13      aws_access_key: "{{ aws_access_key }}"  
14      aws_secret_key: "{{ aws_secret_key }}"  
15      register: vpc  
16
```

Accept or Ignore or Modify suggestion?
Source: Ansible (Extension) Accept Ignore Modify

3. Ansible Lightspeed **will make a code recommendation** for the developer to consider

4. User has option to **Accept, Ignore, or Modify** recommended code snippet

```
File Edit Selection View Go Run Terminal Help  
cludops.yml M ×  
playbooks > cludops.yml  
1 ---  
2 - name: AWS EC2 Cloud operations  
3 hosts: localhost  
4 connection: local  
5  
6 tasks:  
7   - name: Create a virtual network  
8     ec2_vpc_net:  
9       name: "{{ vpc_name }}"  
10      cidr_block: "{{ vpc_cidr }}"  
11      region: "{{ region }}"  
12      state: present  
13      aws_access_key: "{{ aws_access_key }}"  
14      aws_secret_key: "{{ aws_secret_key }}"  
15      register: vpc  
16
```

Please provide feedback
Press 'Enter' to confirm your input or 'Escape' to cancel

5. If "accepted," playbook is automatically populated and user can move on to the next task

6. User prompted to provide feedback; this is important for **helping to train the model**.

Key differentiator: Content source matching

The screenshot shows a code editor window with a dark theme. The file being edited is `test.yml`, which contains the following Ansible playbook code:

```
1 ---  
2 - name: Create a VM with nginx for a website  
3 hosts: localhost  
4 become: false  
5 tasks:  
6 - name: Install nginx on rhel  
7 when: ansible_os_family == "RedHat"  
8 ansible.builtin.package:  
9   name: nginx  
10  state: present
```

The status bar at the bottom indicates the file is 146 bytes long and has 252 characters. It also shows the current position is Line 10, Column 23, with 2 spaces, using UTF-8 encoding, and is an Ansible file.

Below the code editor, the terminal pane displays the output of the `ANSIBLE: LIGHTSPEED TRAINING MATCHES` command for the role `puwanut.ansible_nginx`. The output includes:

- URL: https://galaxy.ansible.com/puwanut/ansible_nginx
- Path: tasks/nginx_install.yml
- Data Source: Ansible Galaxy roles
- License: MIT
- Ansible type: Playbook
- Score: 1.9092153

Yellow arrows point from the text boxes to the corresponding information in the terminal output. The text boxes are labeled as follows:

- Specific URL**: Points to the URL in the terminal output.
- Data source description**: Points to the Data Source in the terminal output.
- Open source license info**: Points to the License in the terminal output.
- Code score**: Points to the Score in the terminal output.

In the bottom right corner, there is a Red Hat logo and the text "Red Hat Ansible Automation Platform".

Ansible Lightspeed with IBM Watson Code Assistant: Custom Models

```
- name: Add user to z/OS system
hosts: all
gather_facts: false
environment: "{{ environment_vars }}"
tasks:
  - name: Generate random password
```

1

```
- name: Add user to z/OS system
hosts: all
gather_facts: false
environment: "{{ environment_vars }}"
tasks:
  - name: Generate random password
    ansible.builtin.set_fact:
      password: "{{ lookup('password', '/dev/null length=15 chars=ascii_letters') }}"
```

2

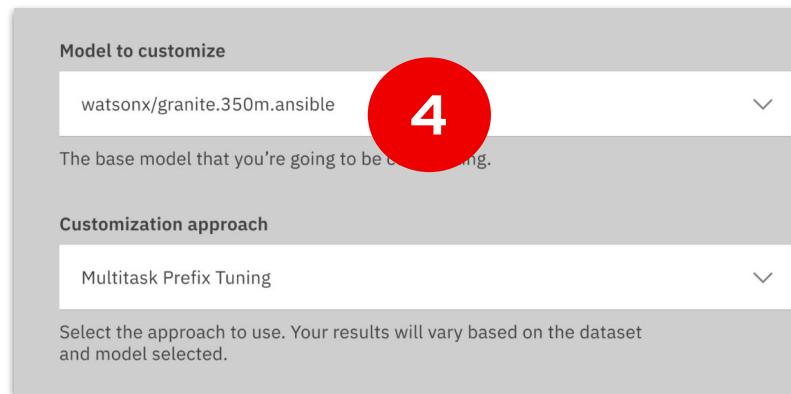
```
- name: Add new user
ansible.builtin.user:
  name: "{{ name }}"
  shell: /bin/bash
  password: "{{ password | password_hash('sha512') }}"
  update_password: "{{ update_password }}"
- name: Permit new user access to resources
  ansible.posix.authorized_key:
    user: "{{ user }}"
    key: "{{ lookup('file', item) }}"
    loop: "{{ copy_user_public_keys | flatten(levels=1) }}
```

3

1. Task description written in natural language in VSCode editor

2. Hit enter, get code recommendation for the task (from base foundation model)

3. Developer fills in unique variables; watsonx foundation model delivers predictive suggestions



4

List of URLs

<https://github.ibm.com/ansible-collections-cio>
<https://github.ibm.com/ansible-collections-cirrus>
<https://github.ibm.com/ansible-projects-cio>

Enter a list of URLs to GitHub organizations, delimited by new lines.

5

4. Watson Code Assistant customizable with private data sets (IBM CIO team data in this example)

5. Tune base model with private data, which is then pulled in and processed

```
- name: Add user to z/OS system
hosts: all
gather_facts: false
environment: "{{ environment_vars }}"
tasks:
  - name: Generate random password
    ansible.builtin.set_fact:
      password: "{{ '' | generate_password }}"
      no_log: true
  - name: Add new user
    ibm_zos_core.zos_tso_command:
      commands:
        - ADDUSER({{ user }} upper )
        - DFLTRGR({{ default_group }} upper )
        - AUTHORITY({{ default_group_authority }} upper )
        - OWNER({{ owner }} upper )
        - NAME('{{ name }}')
        - PASSWORD('{{ password }} upper ')
        - PHRASE('{{ passphrase }}')
        - SECLEVEL({{ security_label }} upper )
        - SECLEVEL({{ security_level }} upper )
        - ADSCATEGORY({{ category }} upper )
        - TSOACCTNUM({{ tso_account_number }} upper )
        - PROC({{ tso_logon_procedure }} upper )
        - DFP(DATAAPPL({{ dfp_data_application }}) DATACLAS({{ data_class }}) MGMYCLAS({{ management_class }}) STORCLAS({{ storage_class }}) OMVSUID({{ omvs_uid }}))
        - HOME('{{ omvs_home_directory }}')"
```

6

6. Ansible Lightspeed now shows new results specific to the custom data set, providing more prescriptive reco base

Ansible Lightspeed with IBM Watson Code Assistant

Features and capabilities found in the **Tech Preview** service



Task generation with NLP

Users can generate automation tasks to create Ansible Playbooks using natural language prompts.

Ansible Foundation Model

Includes access to IBM's Ansible "Granite" foundation model, which is trained on Ansible Galaxy content, and infused with additional Red Hat Ansible subject matter expertise.

Pre- and post- data processing

Feature that amplifies quality and relevance of automation code recommendations from Ansible specific Foundation Model. Code recommendation outputs processed to align with modern Ansible best practices.

Content source matching

The service will always attempt to match a content recommendation to Ansible Galaxy data sources, in order to show the potential provenance of recommendation content from Ansible contributors.

Ansible content developers get recognition for their potential contributions to content recommendations.

Ansible experience enhancement

Delivered through Ansible's existing VS Code extension, and works in concert with other Ansible content tools (ansible-navigator, ansible-builder, ansible-lint)

Ansible Lightspeed will transform the Playbook creation process

- > Generate a playbook or role from a natural language request

Content Generation

- > Find me a playbook or role similar to what I'm writing

Content Discovery

- > Review my playbook and help make it better

Content Optimization

- > Tell me what this playbook is doing - and its impact

Content Explanation

Features on roadmap for upcoming commercial offering

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