



Red Hat Ansible Automation Platform

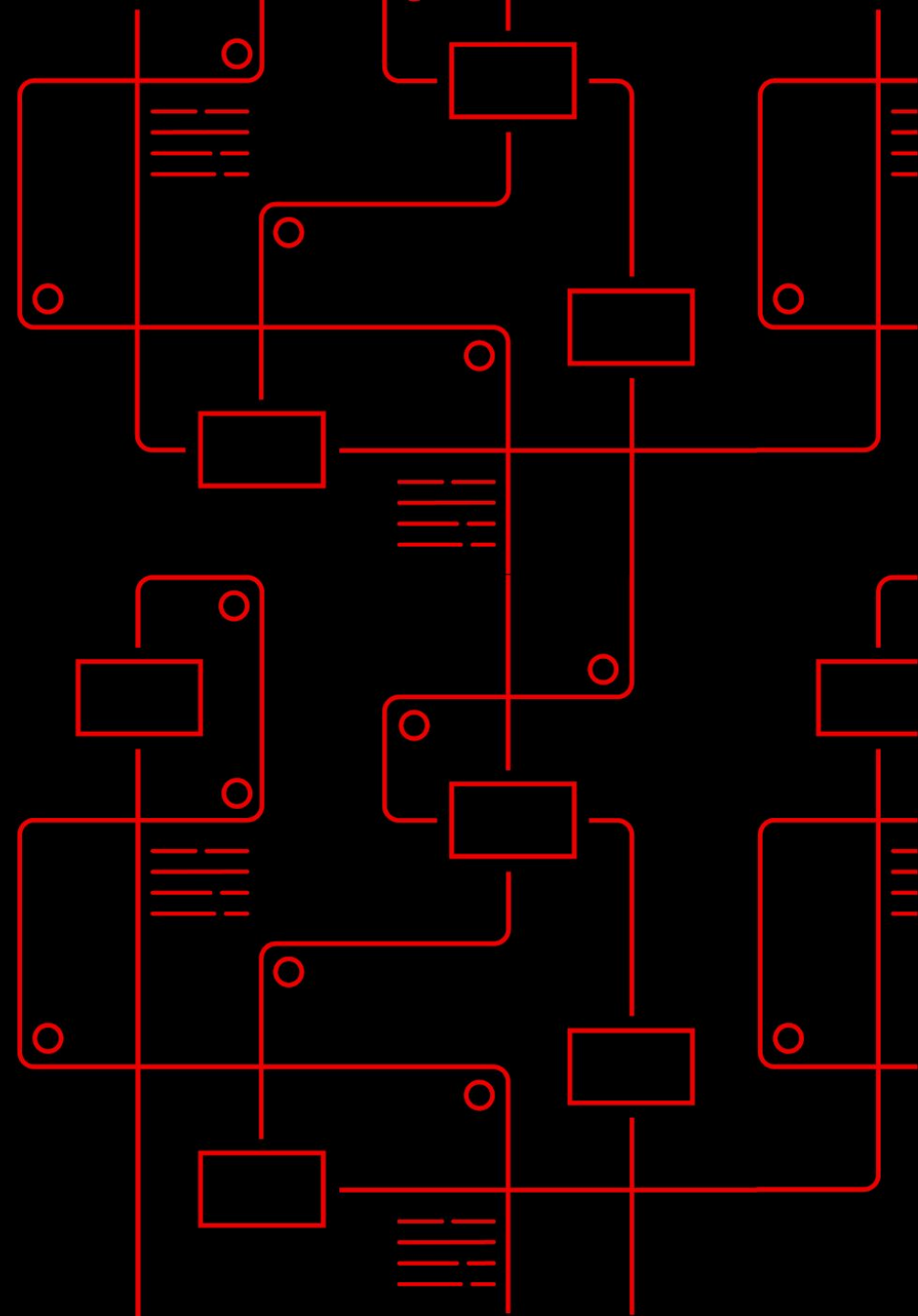
Developer Tools

Brad Krumme

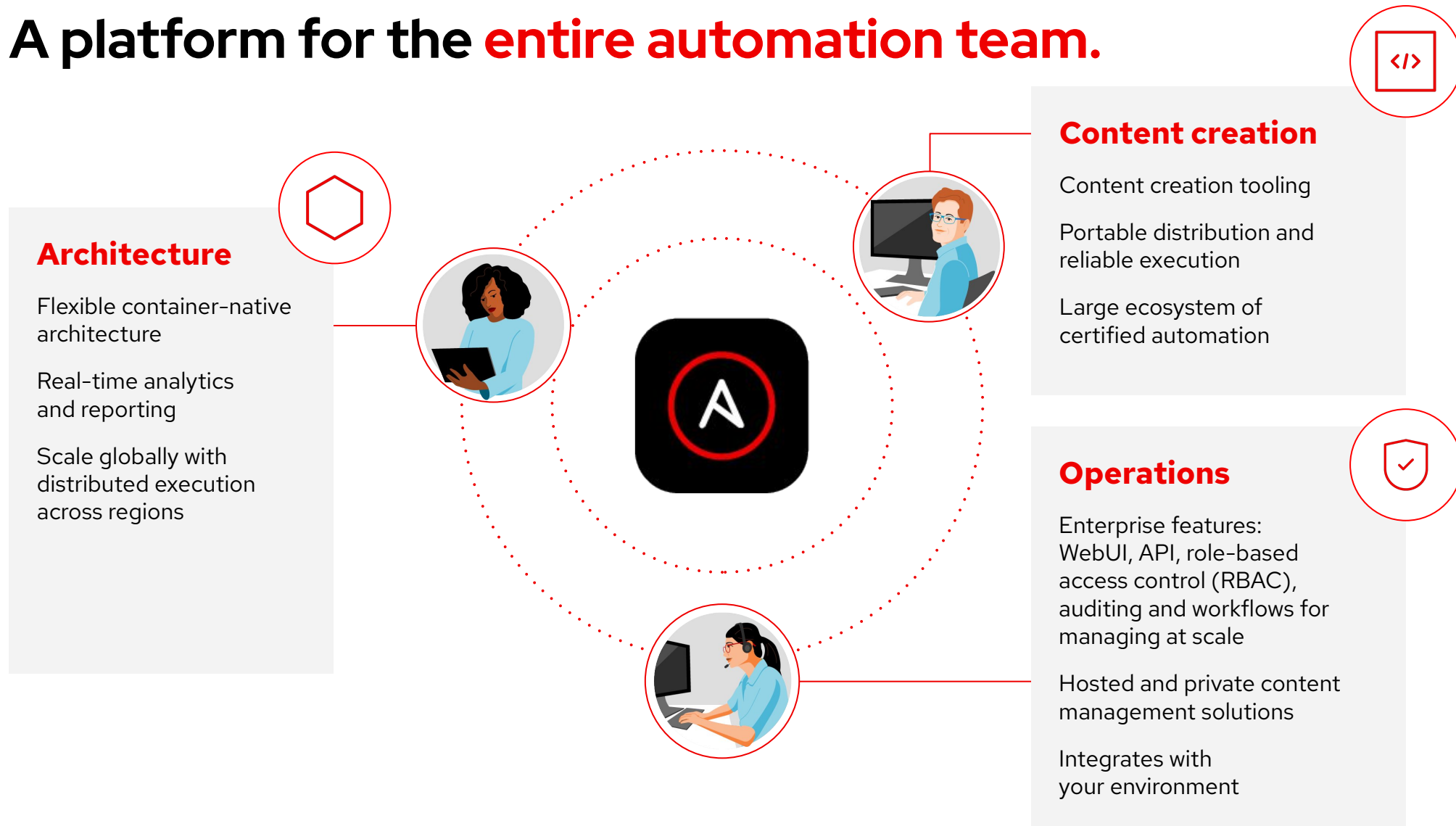
Senior Specialist Solution Architect - Automation



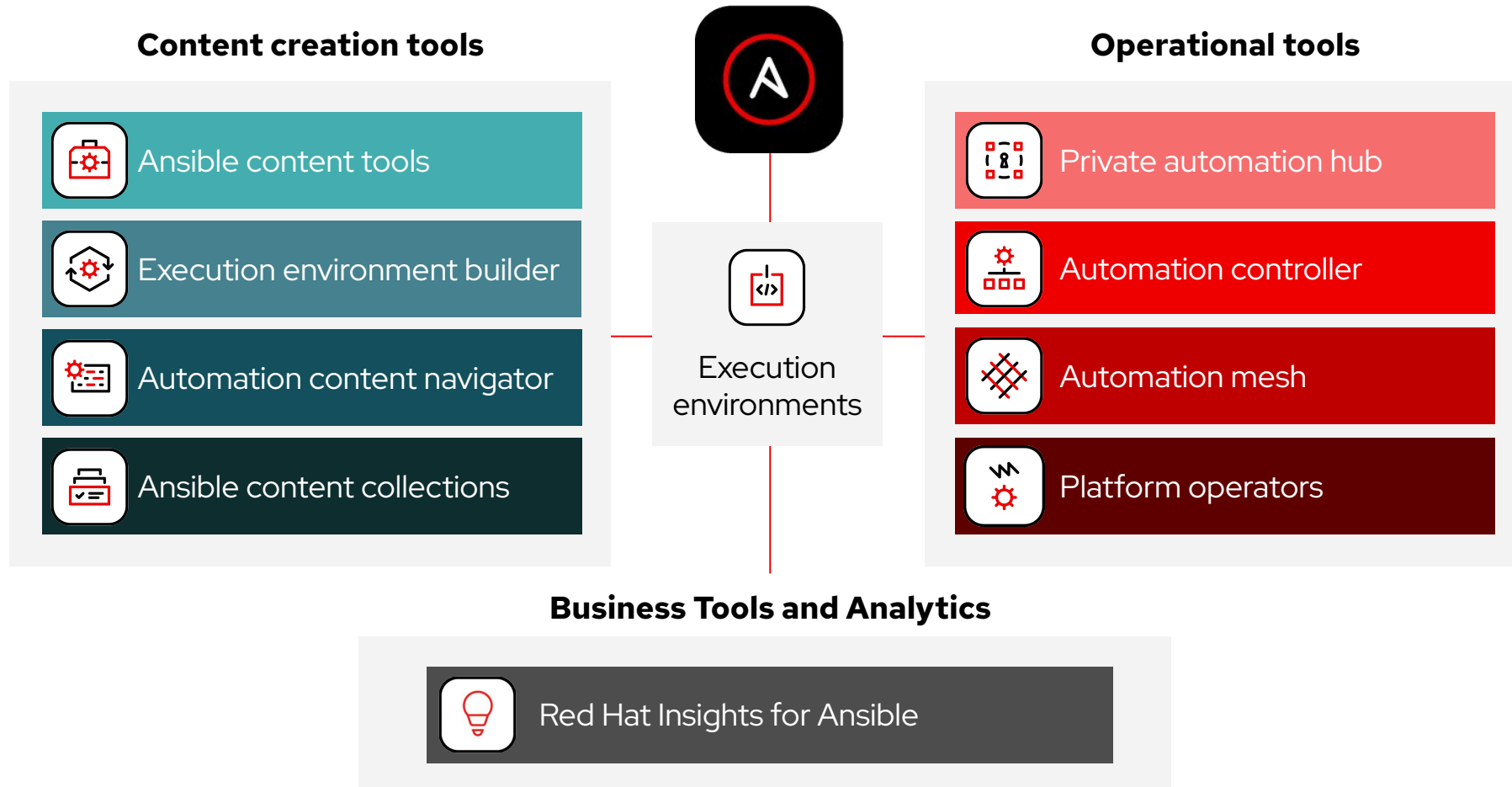
What is Red Hat[®] Ansible[®] Automation Platform?



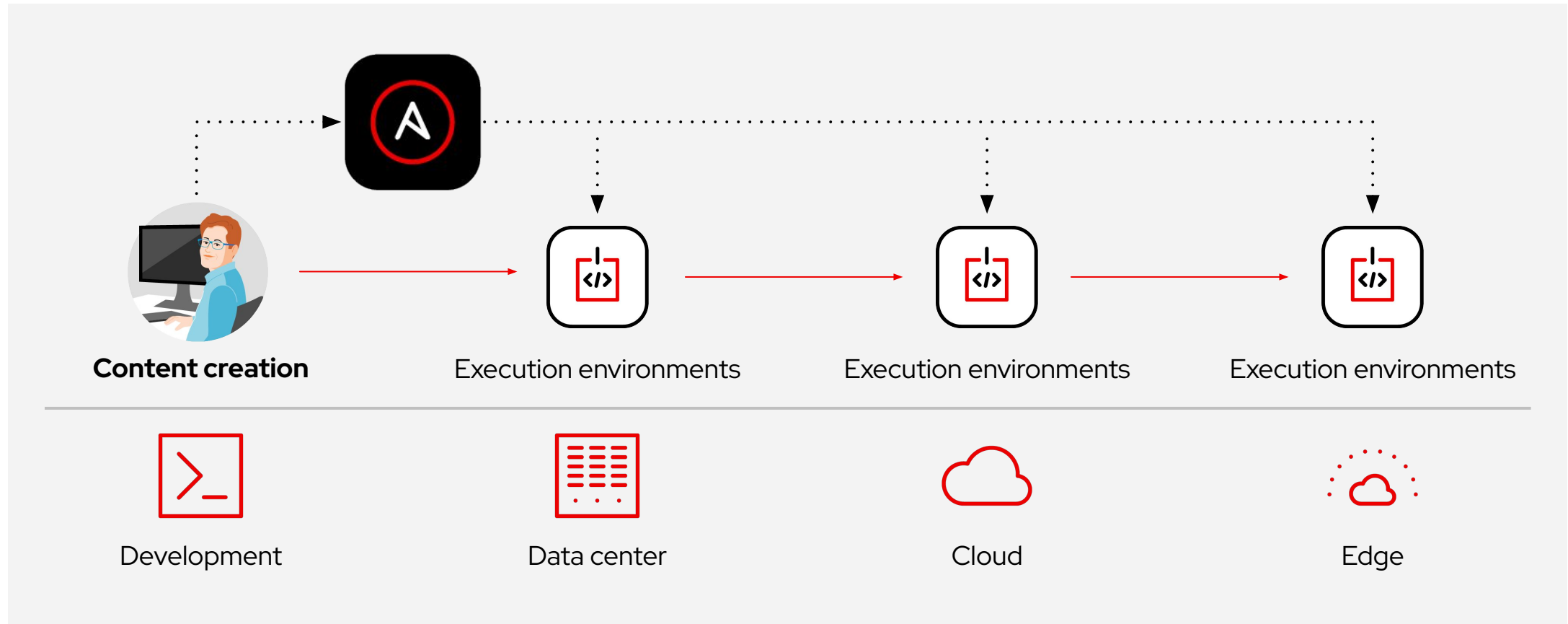
A platform for the **entire automation team.**



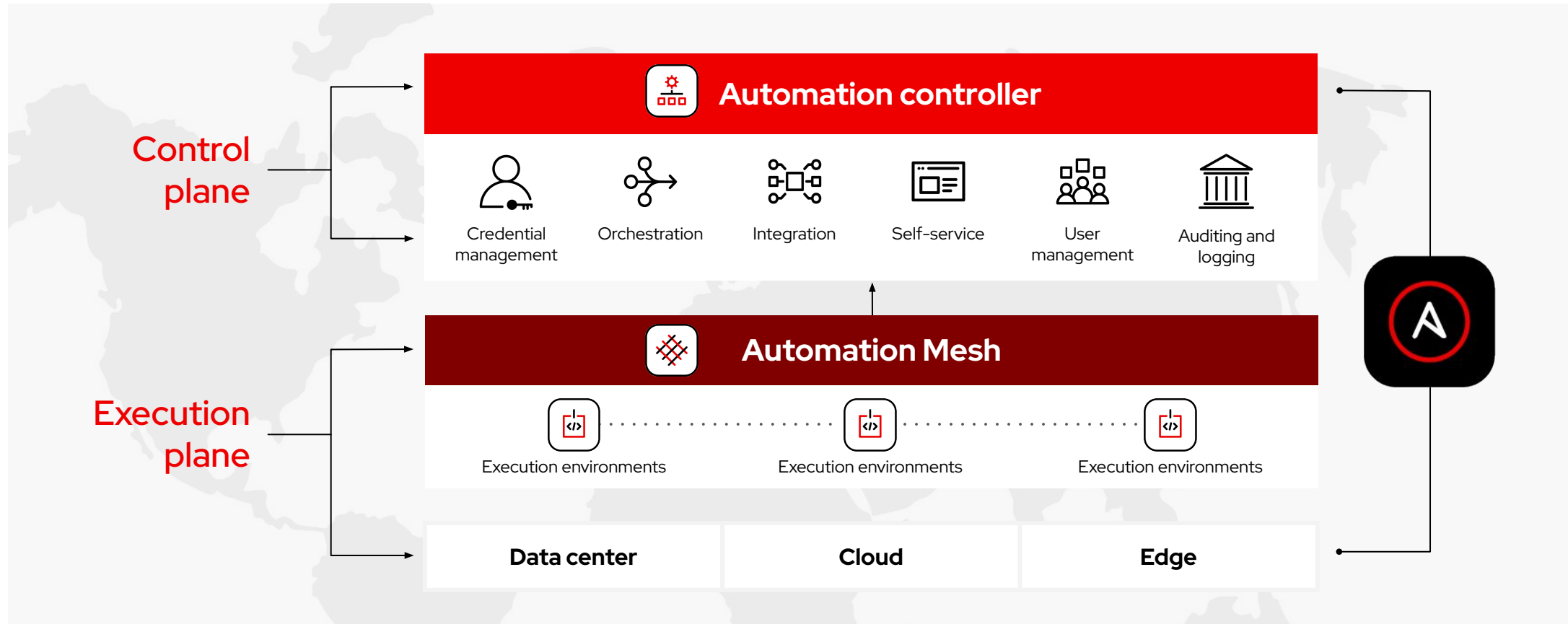
An integrated solution **for the enterprise.**



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



A distributed architecture **built for scale.**



The capabilities you need **across your IT footprint.**



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



Create

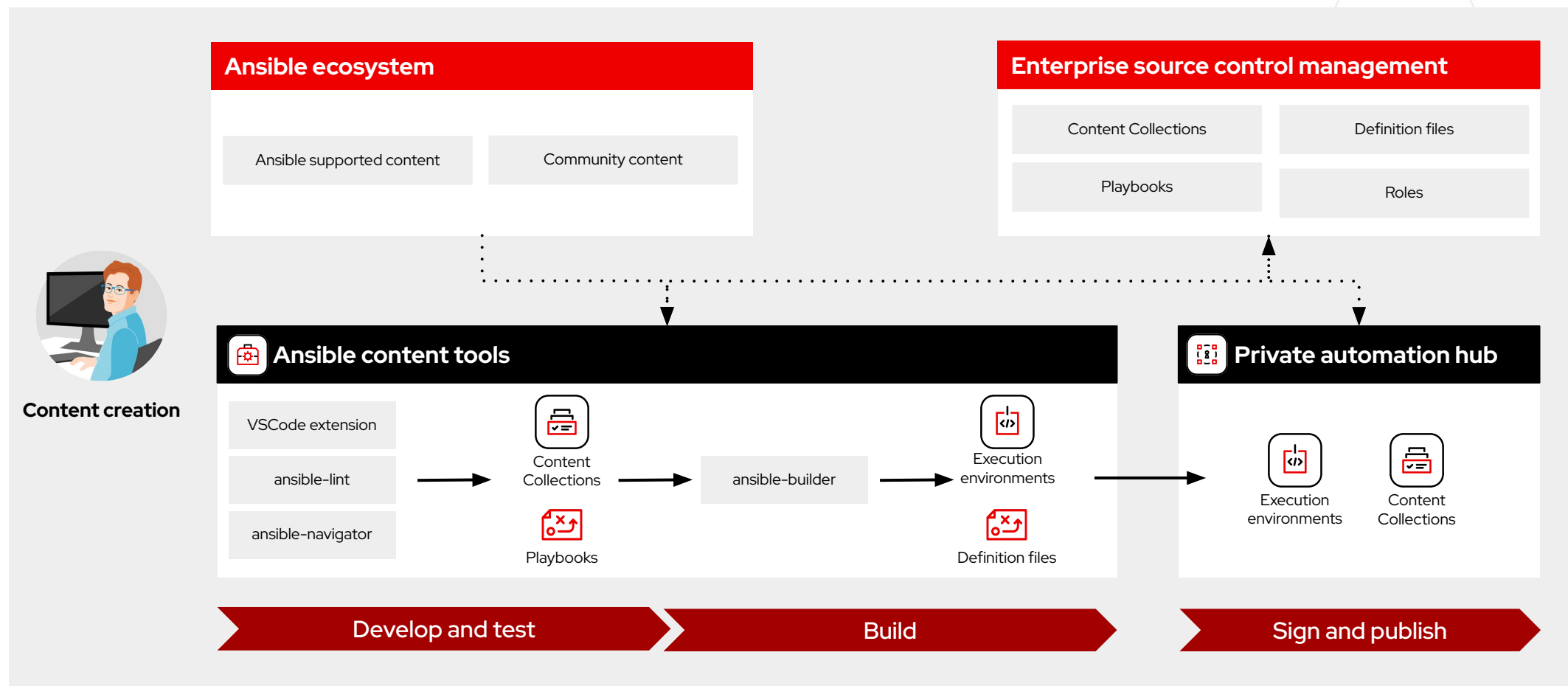
Manage

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

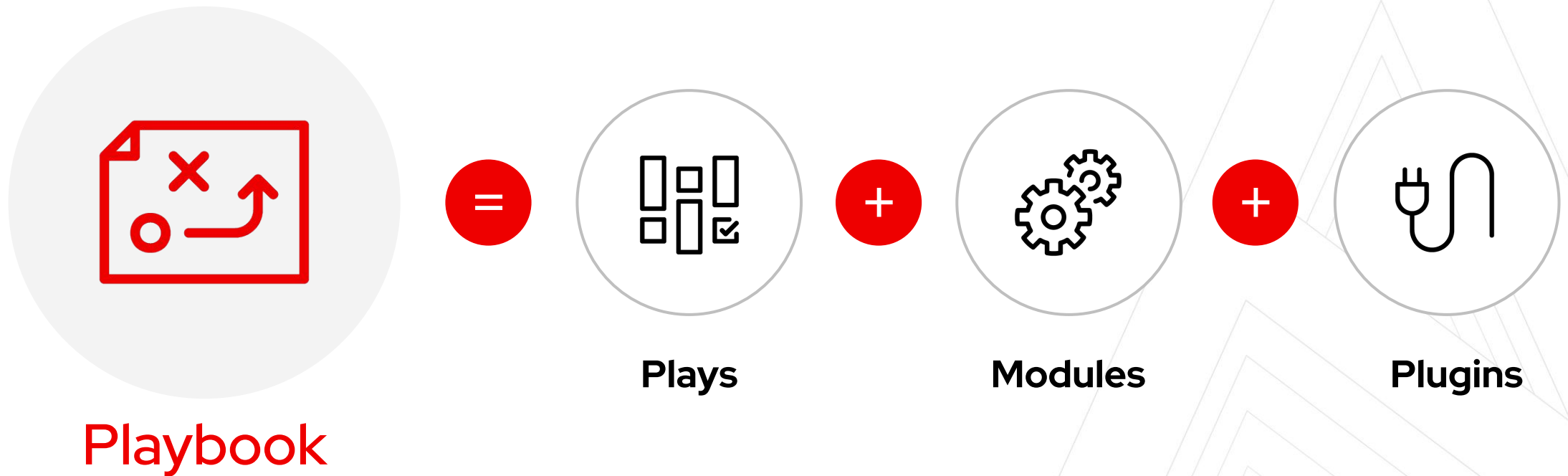
```
---
- name: Install and start apache
  hosts: web
  become: true

  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?





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

```
- 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

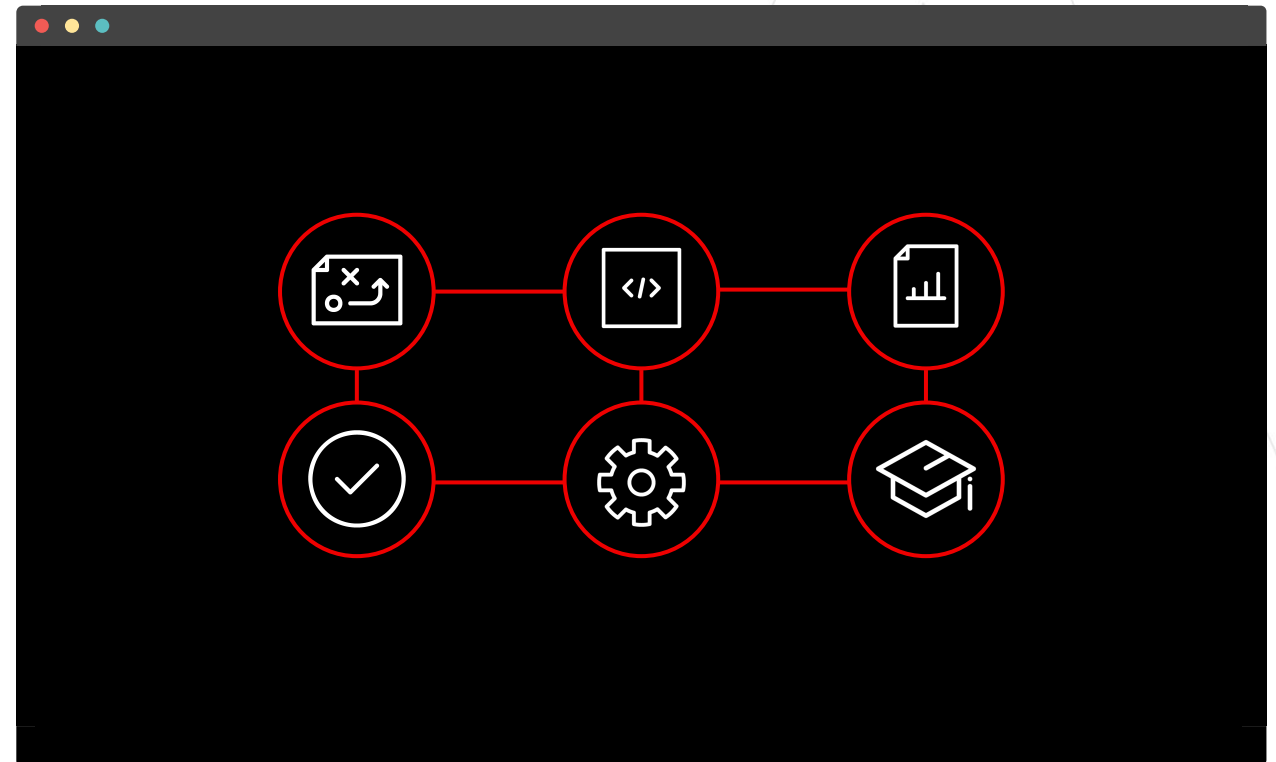
```
---  
- name: Install and start apache  
  hosts: web  
  ansible.builtin.roles:  
    - common  
    - webservers
```

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 content tools

VSCode extension

ansible-lint

ansible-navigator



Content
Collections



Playbooks

ansible-builder



Execution
environments



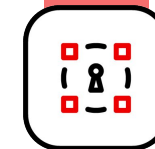
Definition files

Develop and test

Build

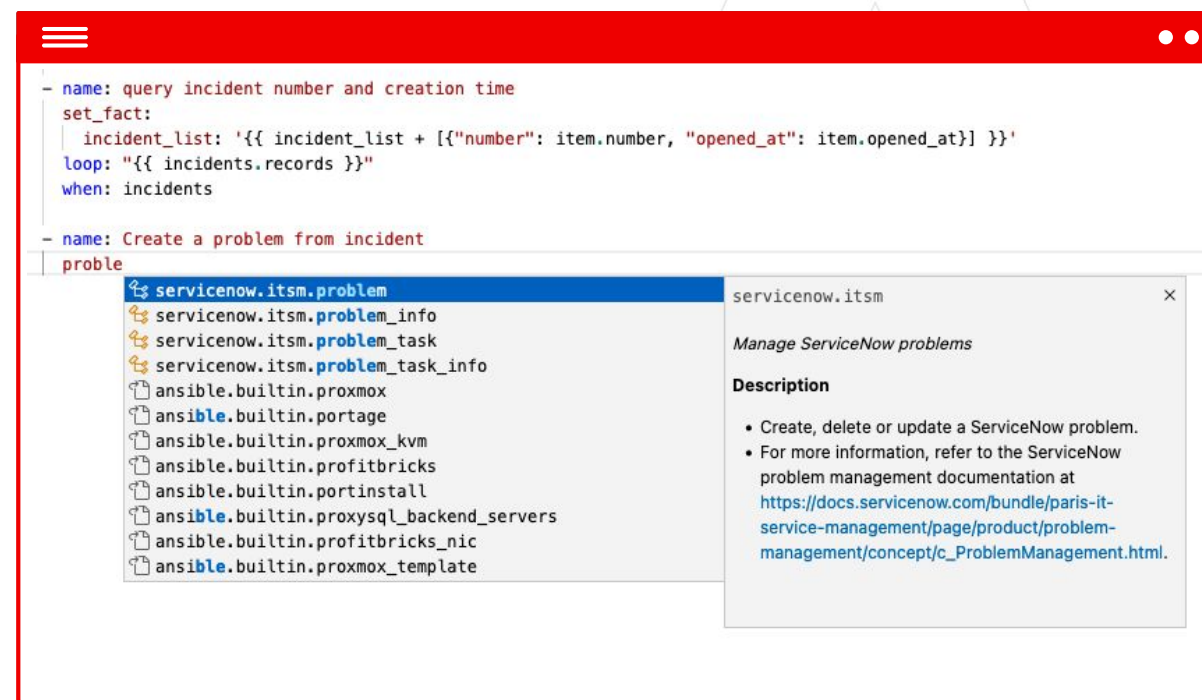
Ansible VS Code extension

Simplifying content creation



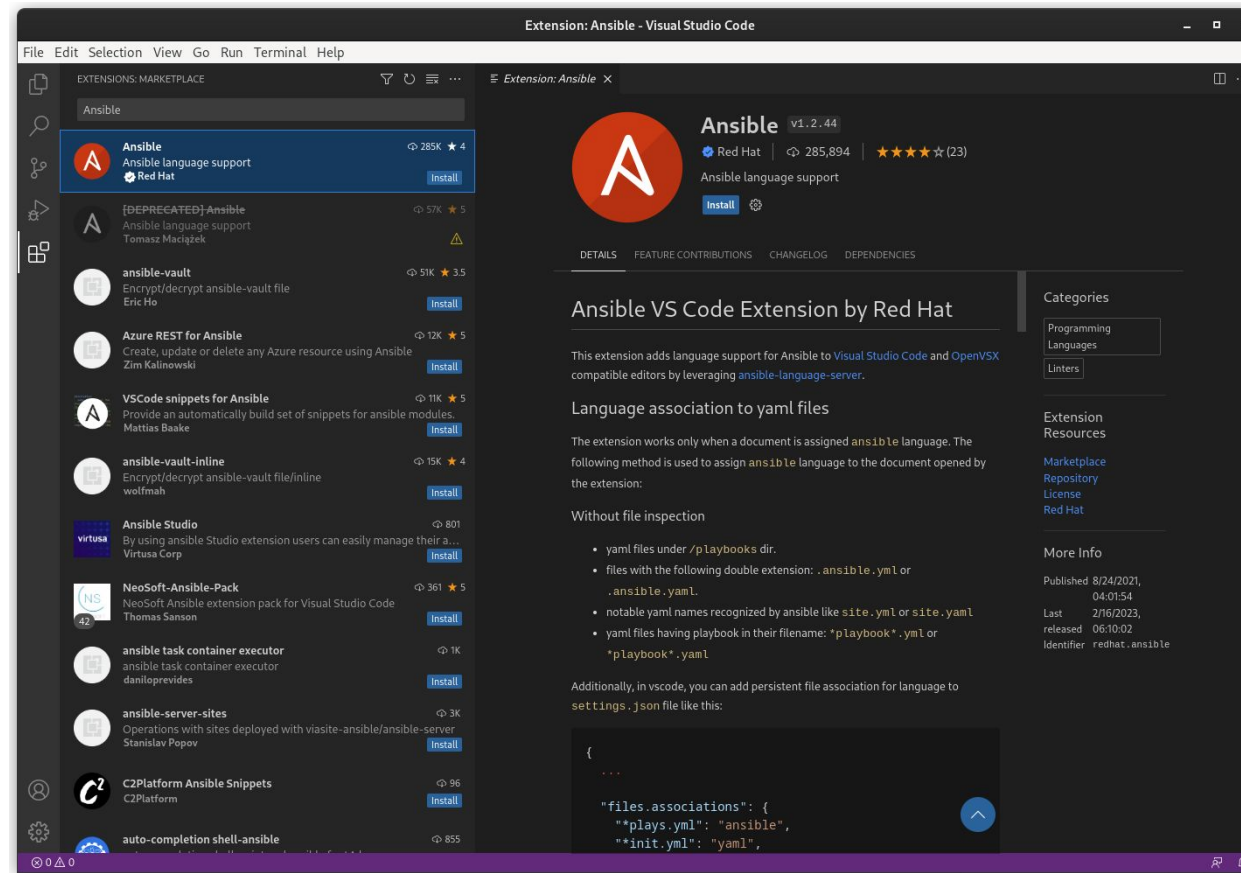
What is it?

- ▶ 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

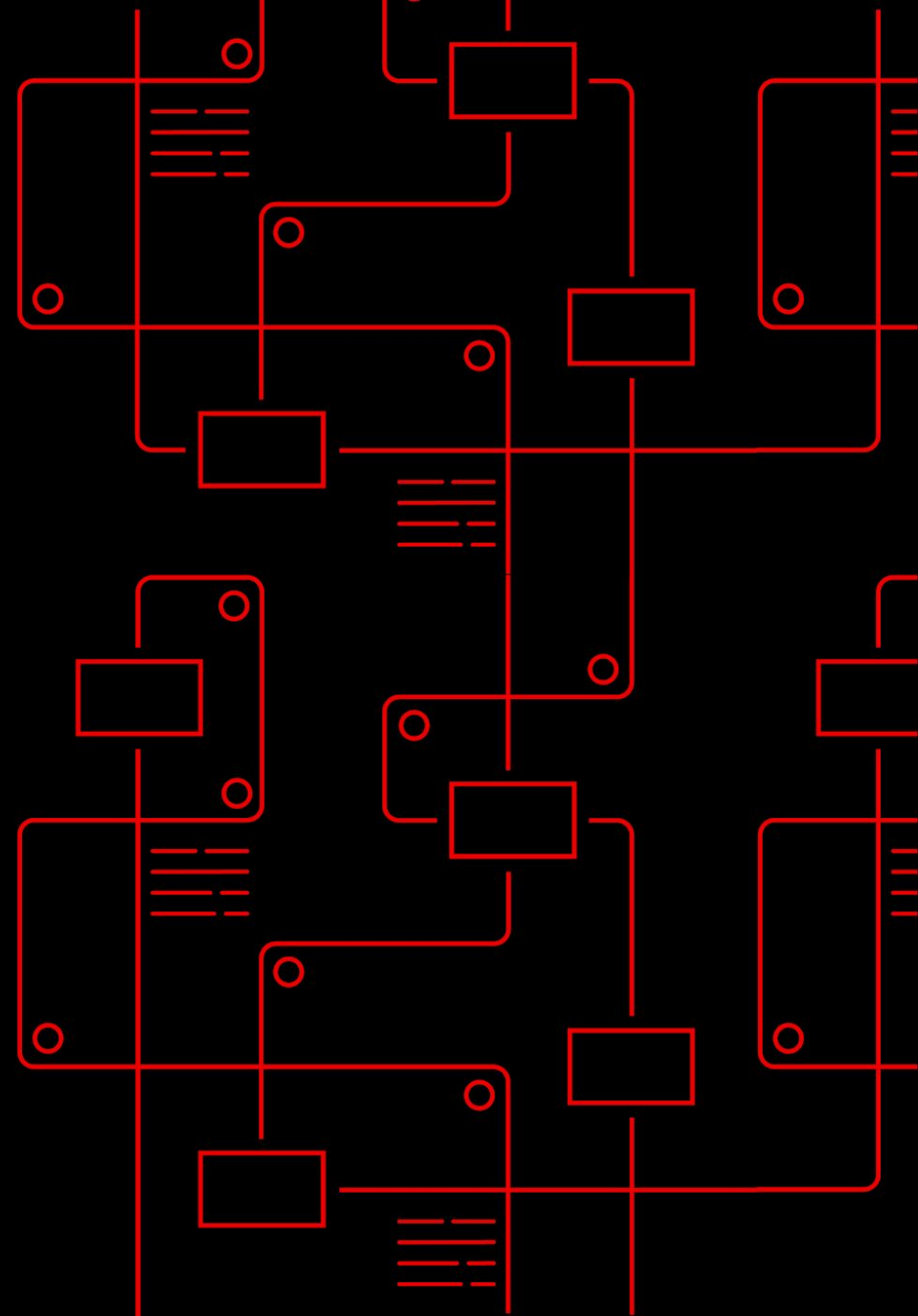


Ansible VS Code extension

Install from VS Code Extensions Menu



Ansible VS Code Demo



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 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.

```
$ ansible-lint playbook.yml
```

Ansible lint (ansible-lint).

Installation paradigms



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

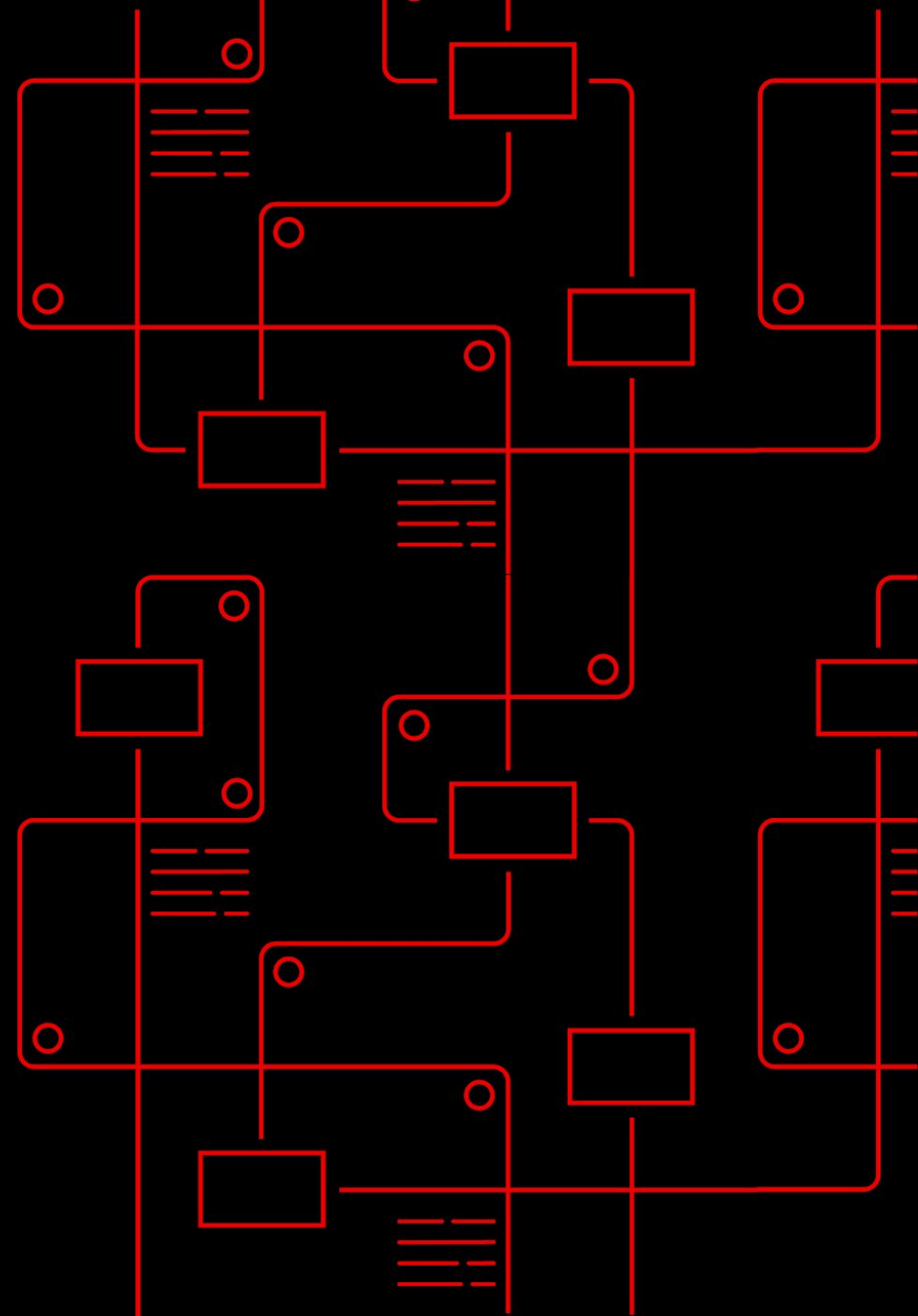
```
$ sudo dnf -y install ansible-lint
```

OR

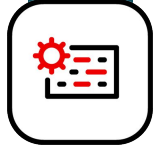
```
# For installation on other linux systems  
(Installation from upstream)
```

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


Ansible Lint Demo



Ansible content navigator (ansible-navigator).



What is it?

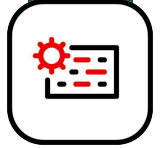
It is a command line utility and text-based user interface (TUI) for running, testing and developing Ansible automation content

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

```
$ 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)
```

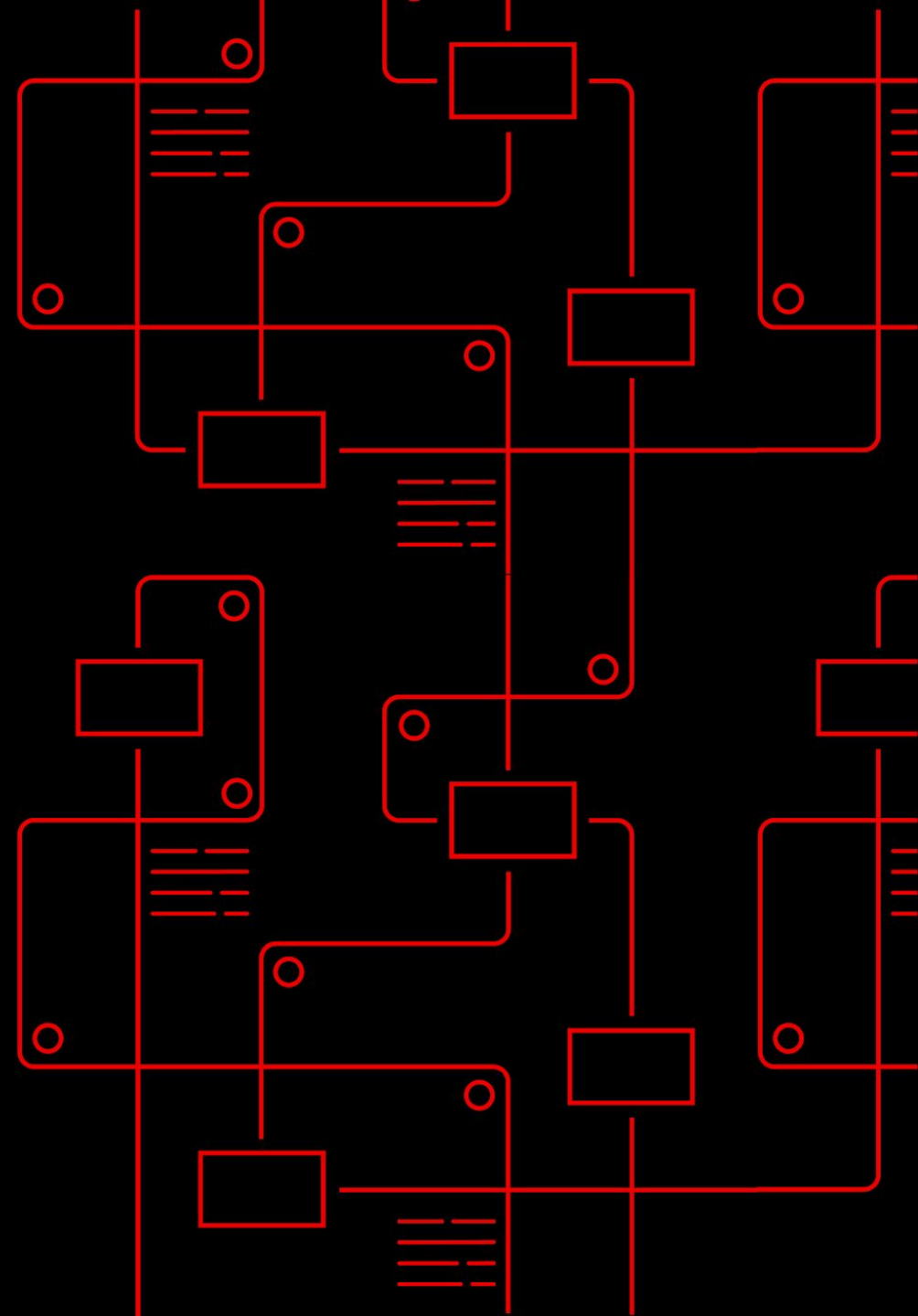
```
$ sudo dnf -y install ansible-navigator
```

OR

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

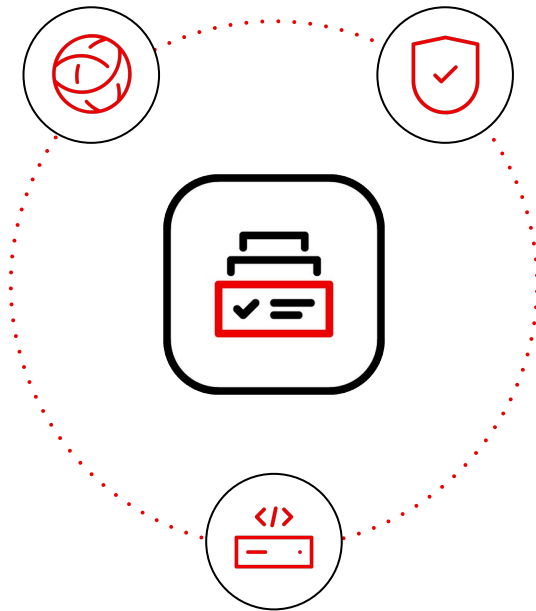
```
$ python3 -m pip install ansible-navigator
```

Ansible Navigator Demo

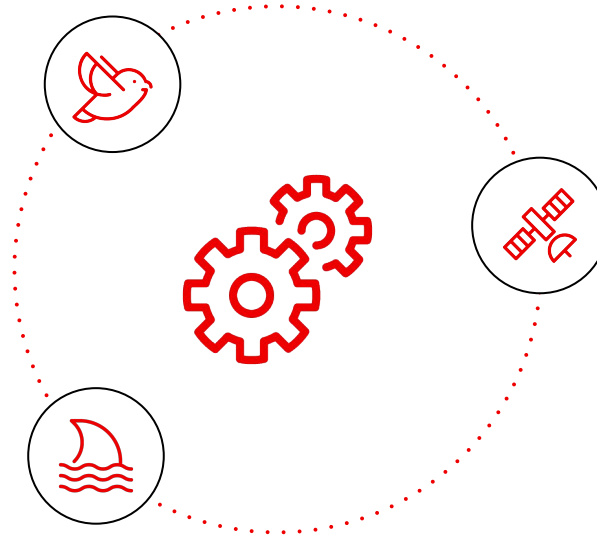


Many technologies, different life cycles.

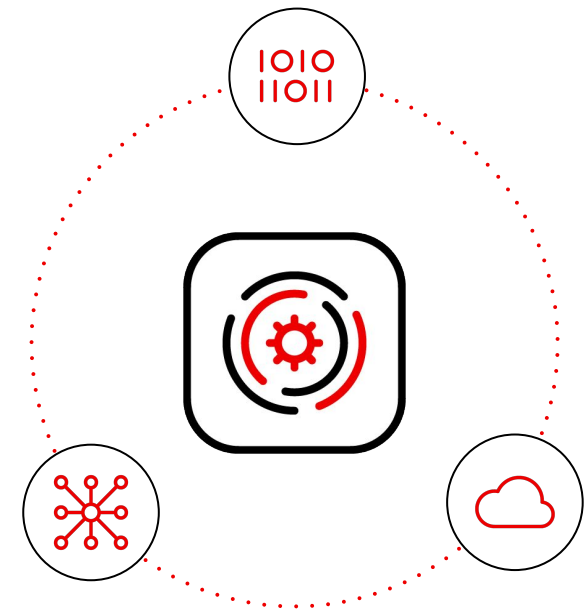
How do I keep it all aligned?



Collections

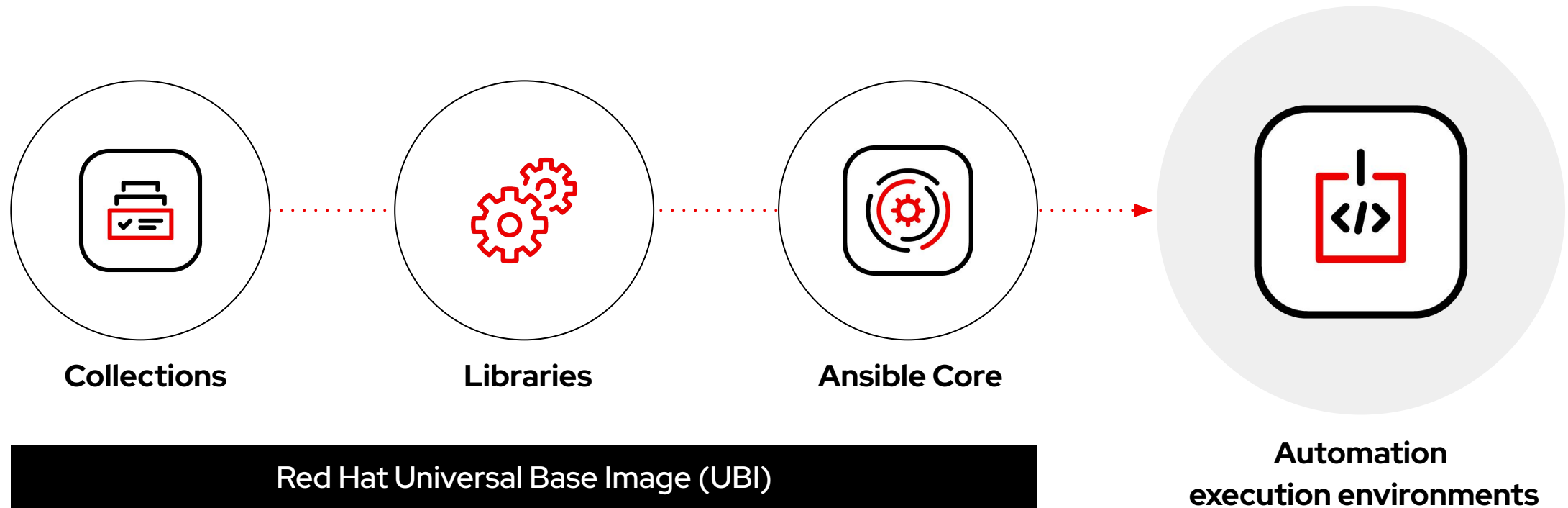


Dependencies



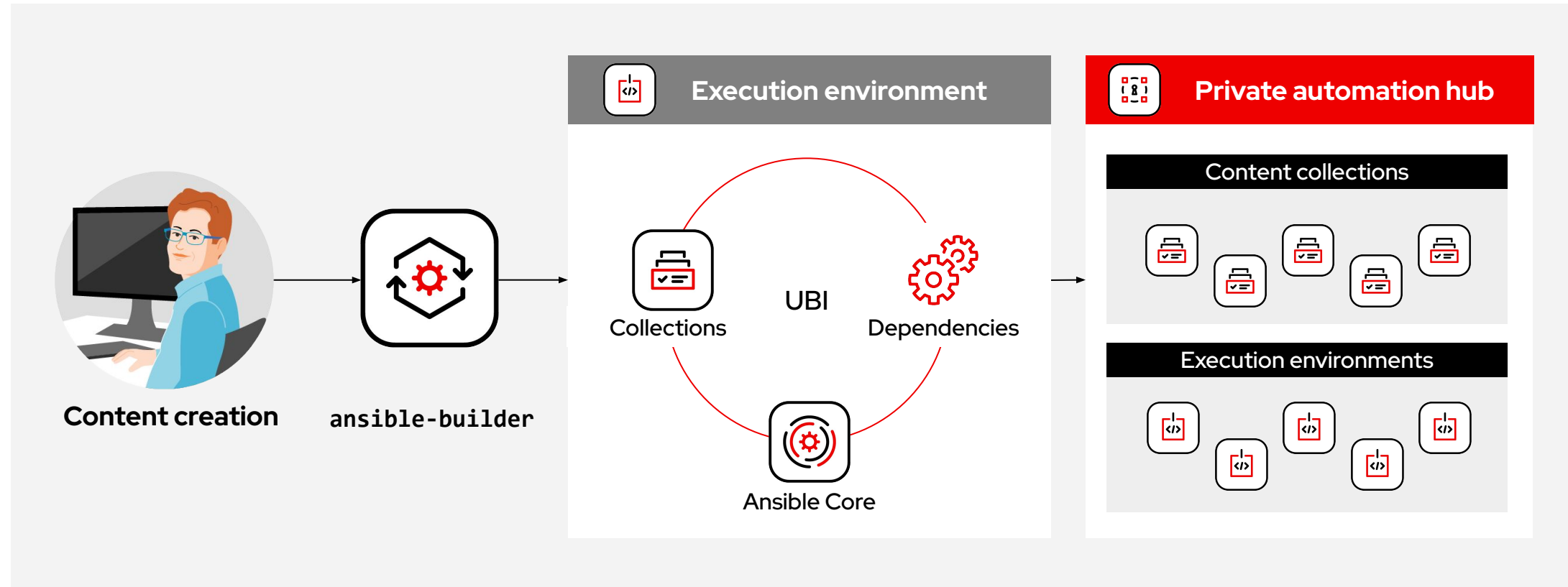
Runtime

Automation execution environments. Reuse and scale automation content.



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

```
$ ansible-builder build --tag repo/custom_ee:latest
```


Execution environment builder. **Build.**

Installation paradigms



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

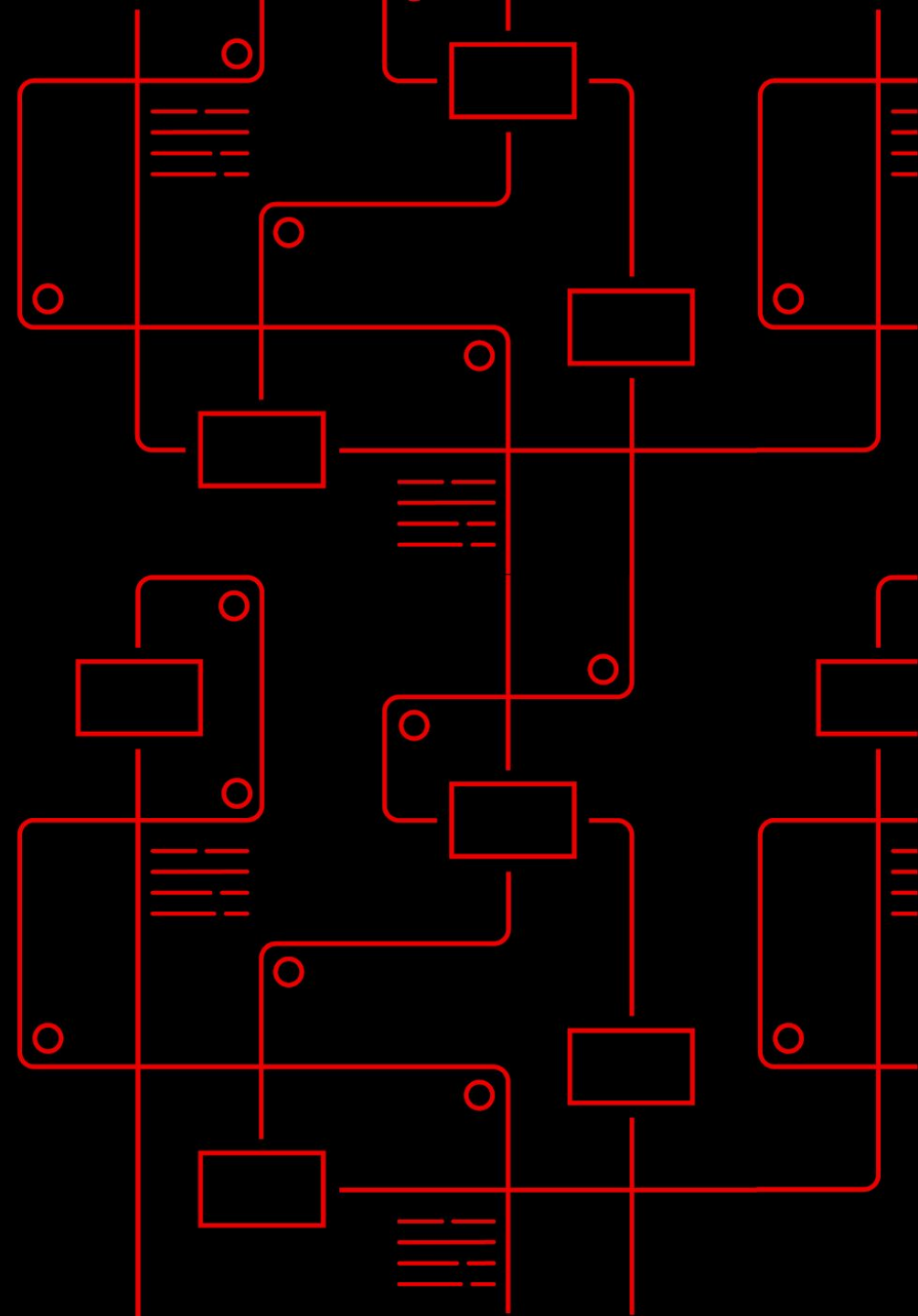
```
$ sudo dnf -y install ansible-builder
```

OR

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

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

Ansible Builder Demo



Thank you

 linkedin.com/company/red-hat

 youtube.com/user/RedHatVideos

 facebook.com/redhatinc

 twitter.com/RedHat