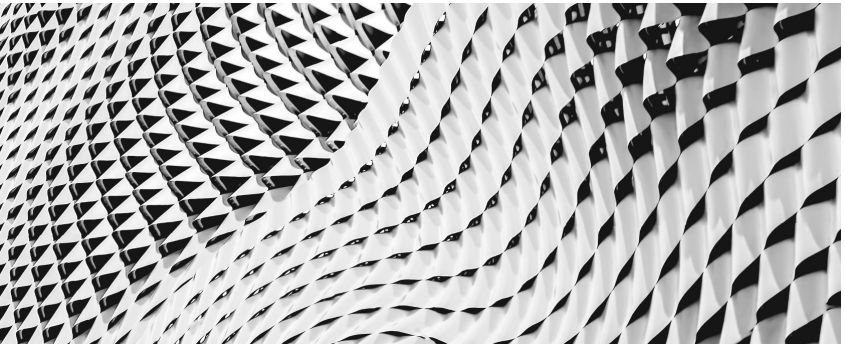
Best practices to deploy Ansible Execution Environments

Alex Dworjan
Associate Principal Specialist Solution Architect
Red Hat Ansible
www.youtube.com/@alexdworjan

What we'll discuss today

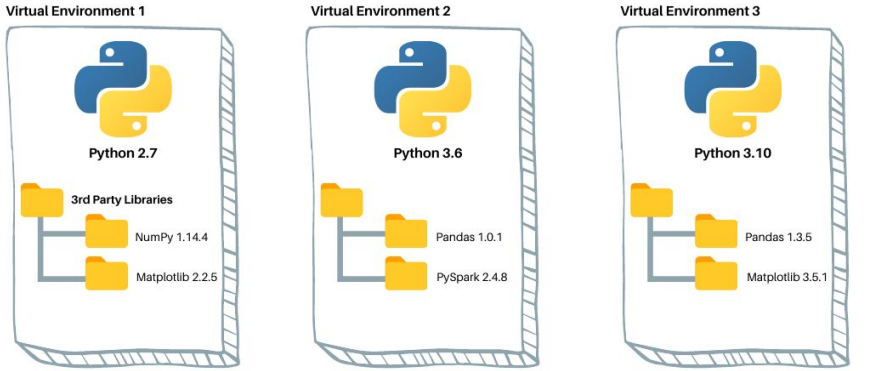
- What are execution environments and why do we need them.
- How to build a custom execution environment with Ansible
- Some guidelines

What are Execution Environments



Python Virtual environments

What did we have before?



dataquest.io

There we specify the Python interpreter and that points to the folder with the third-party libraries installed.

Why are they important?

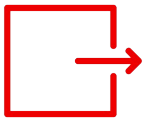
They create an isolated context for project's dependencies.

Limitations of Python Virtual Environments

They don't work for the enterprise

Tooling

Python Virtual Environments are not part of the Red Hat Ansible Automation Platform solution. These are Python constructs meant for Python developers.



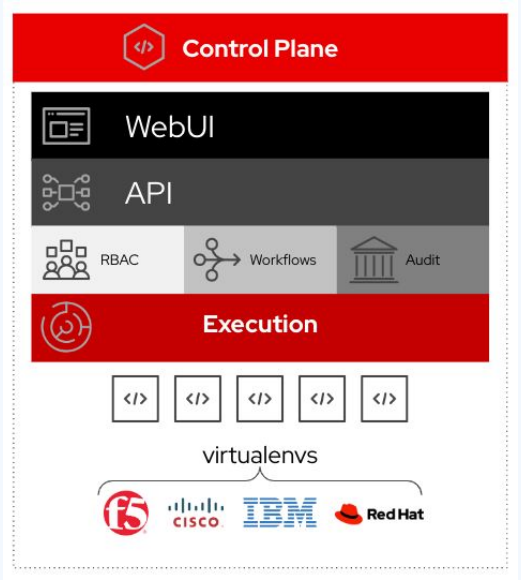
Portability

Python Virtual Environments are unique to a single system and hard to replicate on another system. Its difficult for one person to share their virtual environment with another person to replicate their automation.

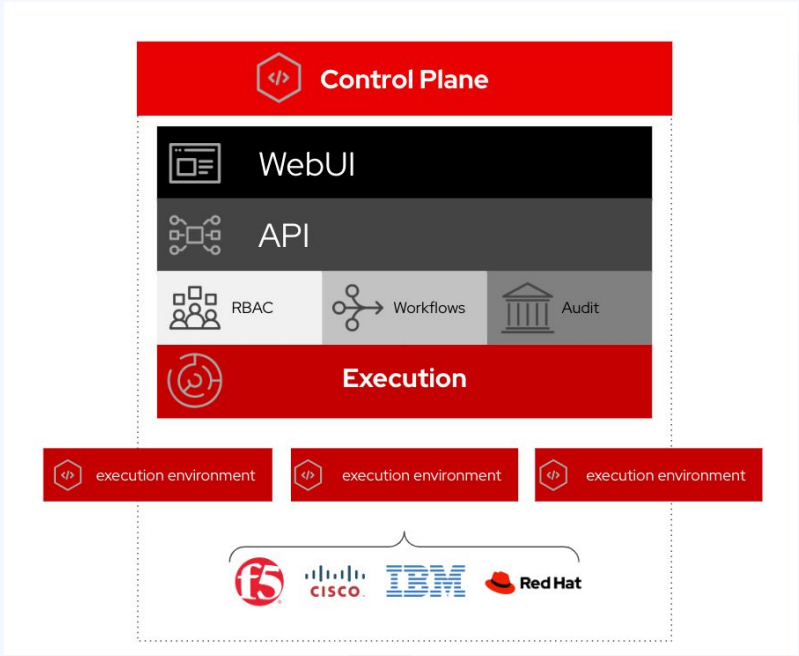
Maintenance

Python Virtual Environments may have dozens of Python dependencies per Ansible Automation Project and become increasingly hard to manage and maintain overtime.

RedHat introduced EEs in AAP 2



Previous architecture



Source:
Ansible Automation Platform 2 What's new (https://docs.google.com/presentation/d/1r-Gz1v3AnoAdXJCnaiJ5xrbrZ-cR9n4FgkZuTQrXzmA)

Virtual vs Execution Environments

Main differences

Virtual Environments

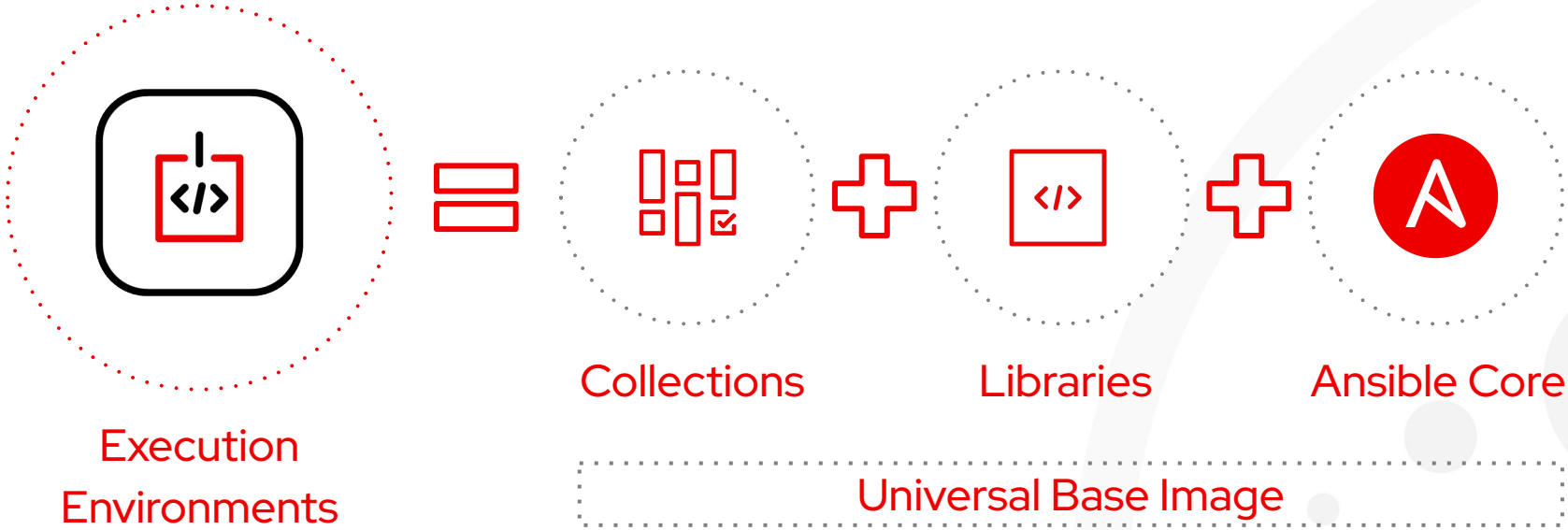
- Tightly coupled to the controller.
- Some commands run inside the venv, others run outside.
- Have to be created manually.

Execution Environments

- As a container, can be run from any configured host or cluster.
- Will be decoupled from the controller which means horizontal scaling will be automatic.
- Entirely contained.
- Tools such as Ansible Builder will make creation easier.

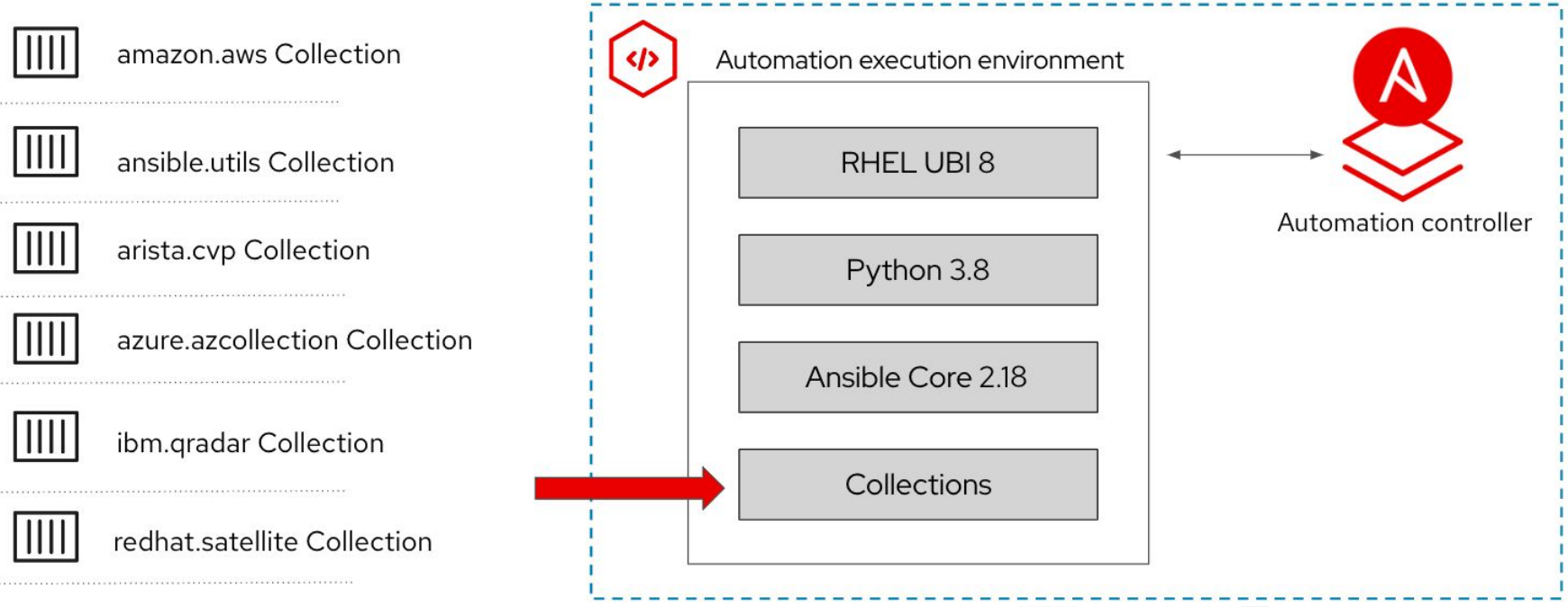
Automation Execution Environments

An automation execution environment is a container image used to execute Ansible playbooks and roles. It provides a defined, consistent, portable environment for executing automation.



Source:
Ansible Automation Platform 2 What's new (https://docs.google.com/presentation/d/1r-Gz1v3AnoAdXJCnaiJ5xrbrZ-cR9n4FqkZuTQrXzmA)
https://www.ansible.com/products/execution-environments

What's in an automation execution environment?



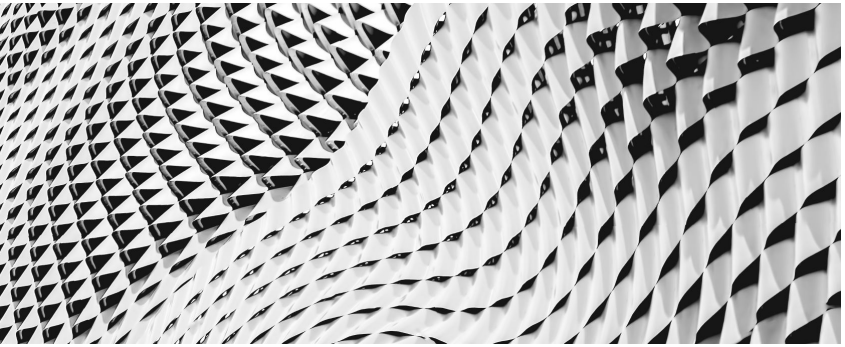
Source:
Ansible Automation Platform 2 What's new (https://docs.google.com/presentation/d/1r-Gz1v3AnoAdXJCnaiJ5xrbrZ-cR9n4FgkZuTQrXzmA)

Execution Environments available

With AAP 2.5

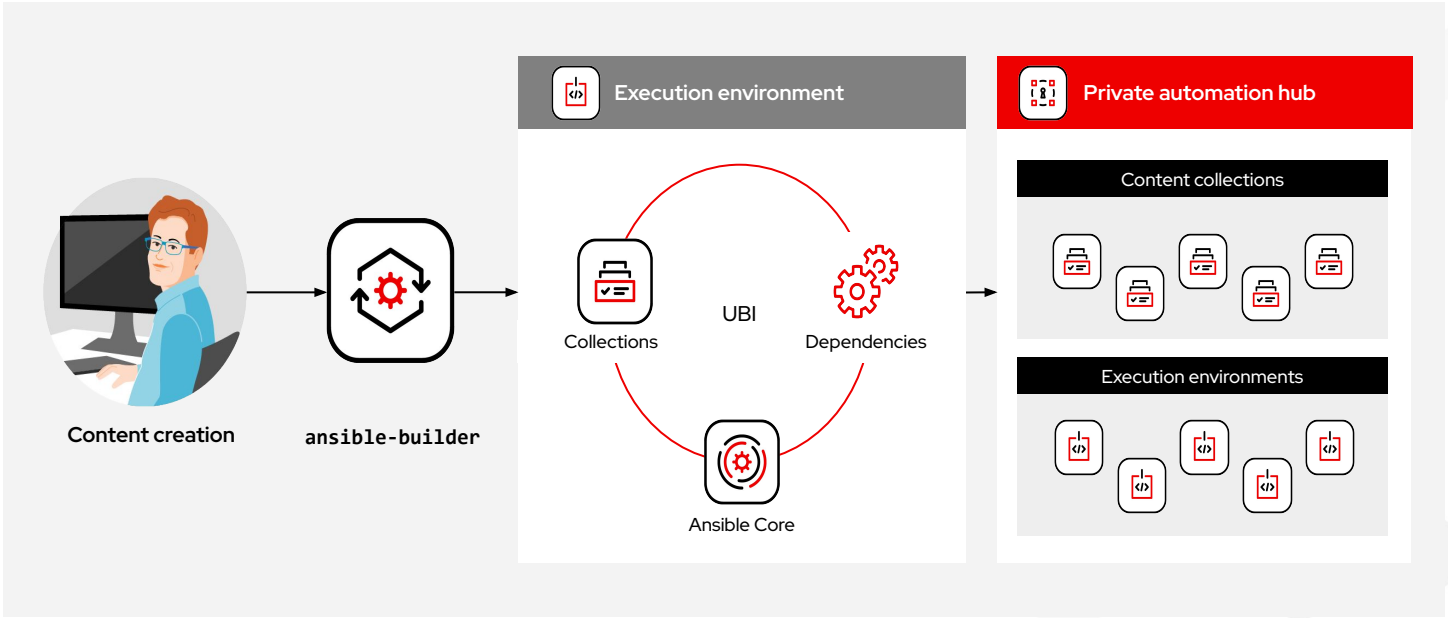
- **`ee-minimal-rhel8`**: minimal container image with ansible-core and basic collections.
- **`ee-supported-rhel8:`** container image with ansible-core and automation content collections supported by Red Hat.

How to build an Execution Environment



Execution environment development

Build, collaborate, sign, publish



Key Sections

Execution Environment Definition File

dependencies

Dictionary value to define dependencies for Ansible Galaxy (collections or roles), Python (libraries) and operating system (packages).

options

This section is a dictionary that contains keywords/options that can affect ansible-builder runtime functionality.

additional_build_files

This section allows you to add any file to the build context directory. For example an ansible.cfg file to download collections from your private automation hub.

additional_build_steps

This section enables you to specify custom build commands for any build phase.

Definition file

YAML Syntax

```yaml
---
version: 3

images:
  base_image:
    name: registry.redhat.io/ansible-automation-platform-24/ee-minimal-rhel8:latest

dependencies:
  galaxy:
    collections:
      - ansible.netcommon

additional_build_files:
  - src: files # copy contents from files to configs dir in build context
    dest: configs

additional_build_steps:
  prepend_galaxy:
    # reference the file from build context like this
    - COPY _build/configs/ansible.cfg /etc/ansible/ansible.cfg

options:
  package_manager_path: /usr/bin/microdnf
```


Source:
https://ansible-builder.readthedocs.io/en/stable/definition/

Dependencies

Requirements files

requirements.yml

```
---
collections:
  - geerlingguy.java
  - kubernetes.core
```

requirements.txt

```
boto>=2.49.0
botocore>=1.12.249
pytz
python-dateutil>=2.7.0
awxkit
packaging
requests>=2.4.2
xmltodict
azure-cli-core==2.11.1
python_version >= '2.7'
collection community.vmware
google-auth
openshift>=0.6.2
requests-oauthlib
```

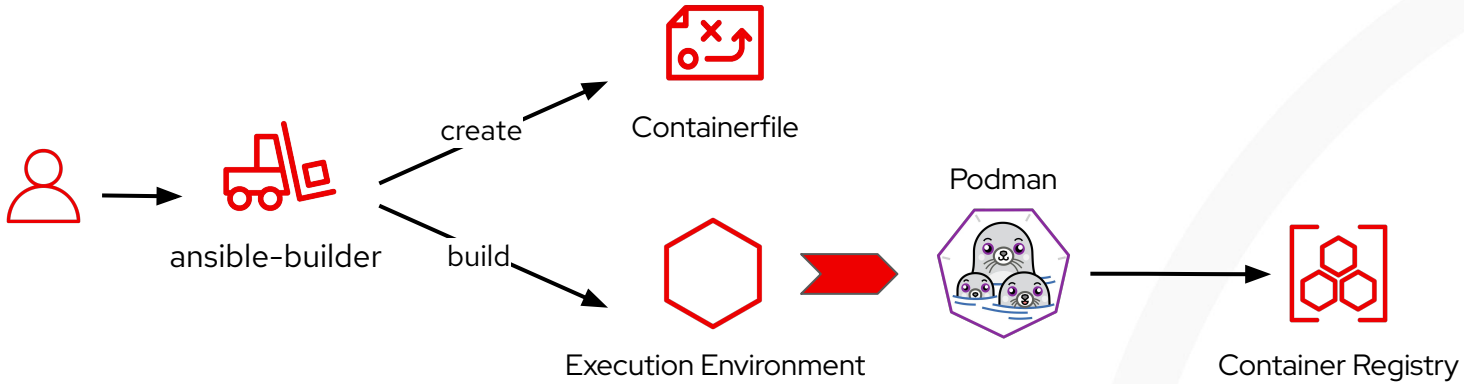
bindep.txt

```
libxml2-devel [platform:rpm]
subversion [platform:rpm]
```


Source:
https://access.redhat.com/documentation/en-us/red_hat_ansible_automation_platform/2.0-ea/html-single/ansible_builder_guide/index

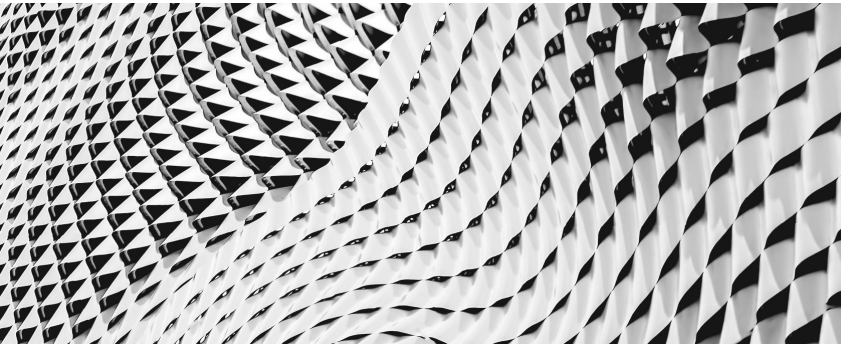
Execution Environment Building Workflow

The Builder Tool



Source:
https://podman.io/

Guidelines to build a EE



How to build a good Execution Environment?

Apply the same approach as in software development, such as devops, even if it's a very simple project.

Build as many times as needed until all errors are fixed. Test changes and rollback in the event something needs to be addressed.

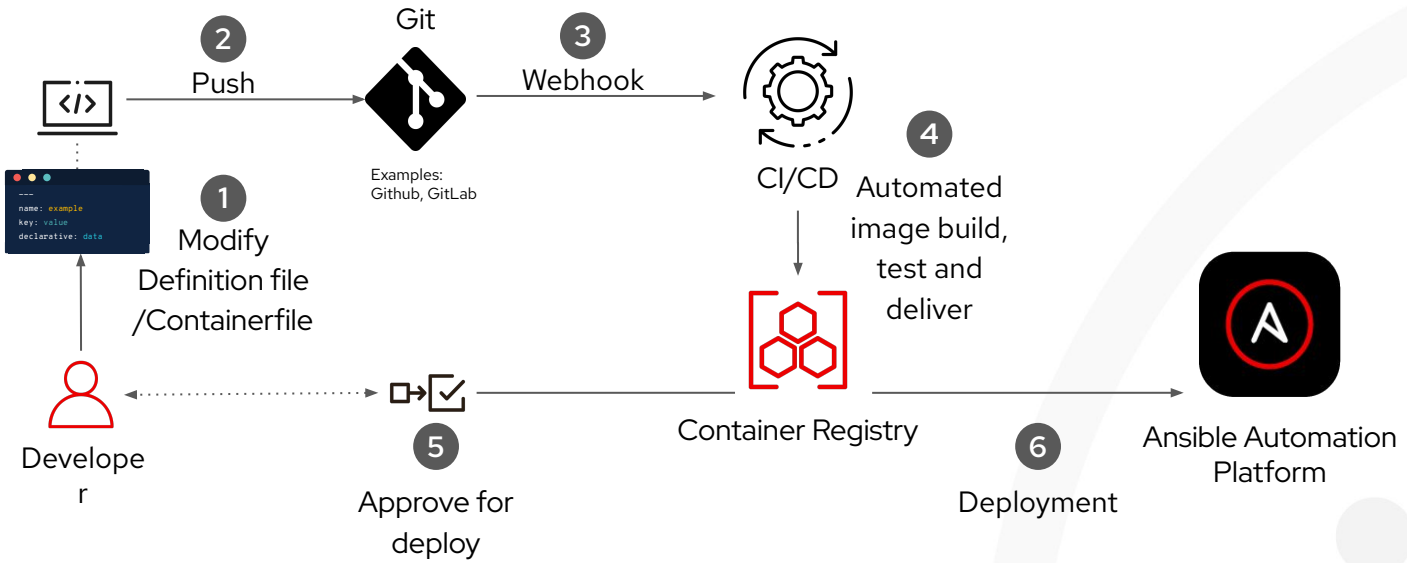
4 good habits for building EEs

1. Restrict images to only what is needed.
2. Update EEs regularly.
3. Do not create monoliths.
4. Keep track of versions and why you included pieces.

Source: https://www.redhat.com/architect/ansible-execution-environment-tips

Use Continuous Integration

When building the Execution Environment



EE as Code

What's the benefit?

- Version control history of every EE built kept in Source Control
- Provide access to modify EE definition without access to EE build host
- Provide Job Template to end-users to build new EE
  - Schedule job to rebuild EE
- Easy to build EE for new team / use-case

Requirements

- VM / Pipeline that includes podman / docker and ansible-builder
- Existing Execution environment with
  - infra.ee_utilities
  - infra.ah_configuration
  - ansible.controller
  - containers.podman or community.docker

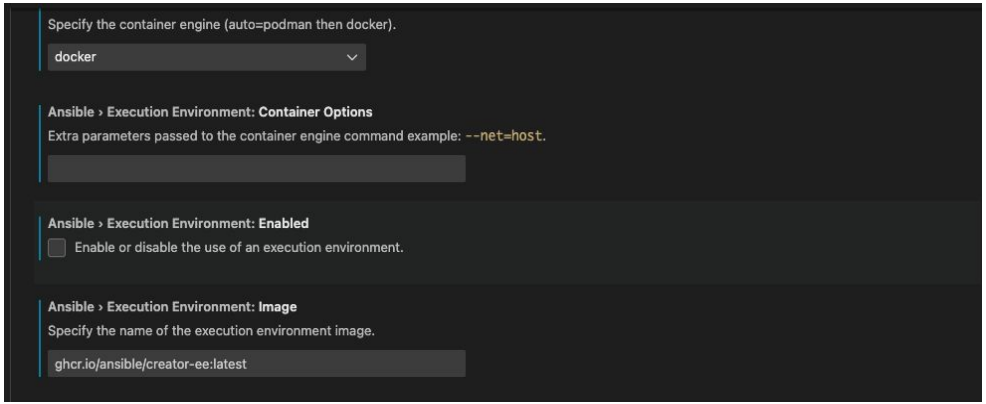
Steps to Achieve

Update Remotes requirements.yml	Update each collection remotes requirements.yml to ensure all necessary collections will be downloaded
Base EE Sync	Sync latest base EE to ensure latest ansible-core and vulnerabilities are addressed
Collection Repo Sync	Sync in all collection repos to ensure all necessary collections and versions are available
Build EE	Build the execution environment on a VM / via pipeline using the EE as code definition
Update controller	Update the Execution Environment tag in controller to ensure users utilize the correct version

Ansible VSCode extension

Test syntax locally before pushing the definition file

- Point your vscode extension to creator-ee to get the built in ansible-core and ansible-lint support irrespective of your local environment



Ansible Navigator

Test image locally before publishing

- ansible-navigator run test.yaml --eei custom-execution-environment:1.0

```
(base) asalgado@asalgado-mac ee_rsync_content % ansible-navigator run sync.yaml --eei custom-execution-environment:1.0 --pp missing -m stdout
[WARNING]: provided hosts list is empty, only localhost is available. Note that
the implicit localhost does not match 'all'

PLAY [localhost] ***********************************************************

TASK [Gathering Facts] *****************************************************
ok: [localhost]

TASK [ansible.posix.synchronize] *******************************************
fatal: [localhost]: FAILED! => {"changed": false, "cmd": "/usr/bin/rsync --delay-updates -F --compress --archive --out-format='<<CHANGED>>%i %n%L' /home/runner/
ome files/attrs were not transferred (see previous errors) (code 23) at main.c(1189) [sender=3.1.3]\n", "rc": 23}

PLAY RECAP *****************************************************************
localhost                  : ok=1    changed=0    unreachable=0    failed=1    skipped=0    rescued=0    ignored=0
Please review the log for errors.
```

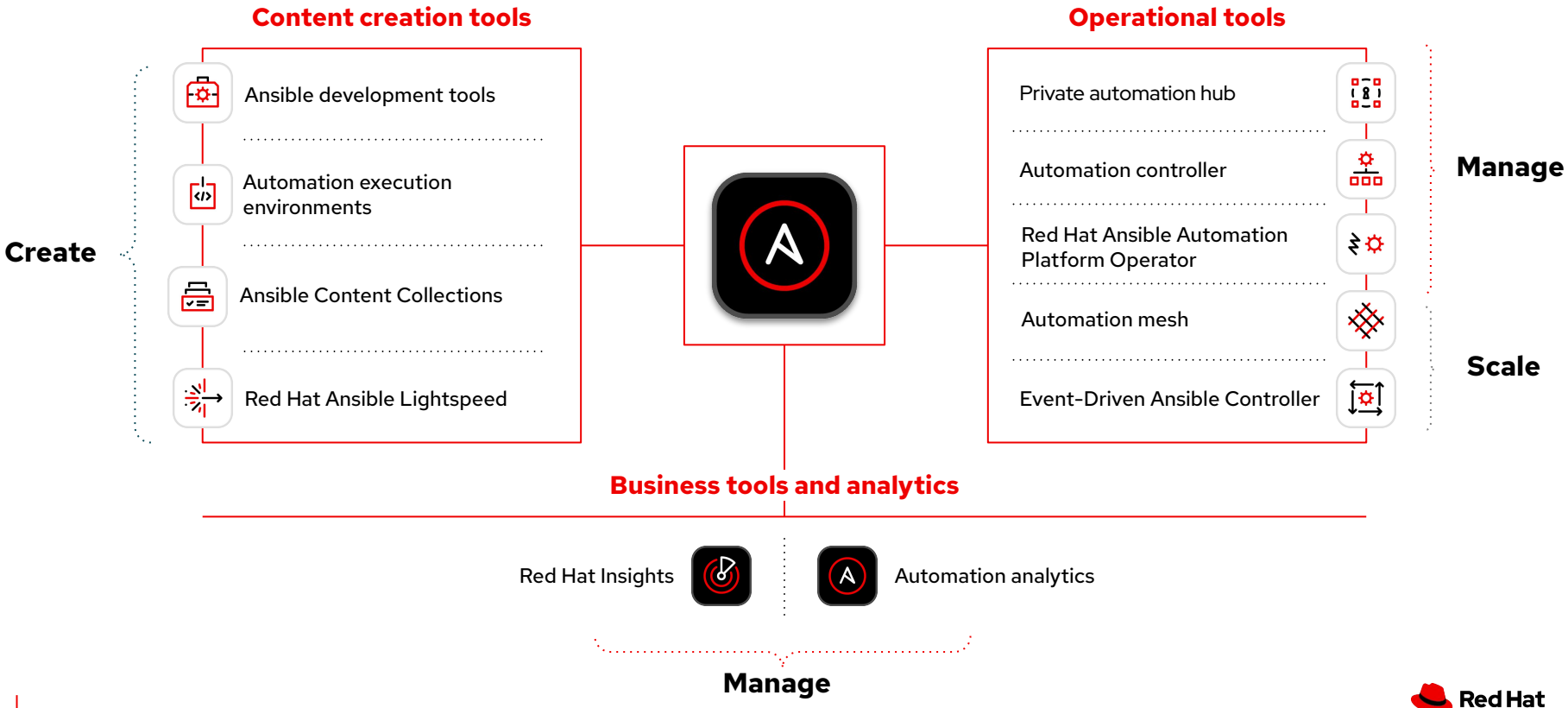
Resources

1. https://www.ansible.com/blog/whats-new-in-ansible-automation-platform-2-automation-execution-environments
2. https://www.ansible.com/blog/unlocking-efficiency-harnessing-the-capabilities-of-ansible-builder-3.0
3. https://www.ansible.com/blog/automating-execution-environment-image-builds-with-github-actions
4. https://www.ansible.com/blog/the-anatomy-of-automation-execution-environments
5. https://www.redhat.com/architect/ansible-execution-environment-tips
6. https://www.redhat.com/architect/ansible-execution-environment-automated-build
7. https://www.ansible.com/blog/digitally-signing-ansible-content-collections-using-private-automation-hub
8. https://www.ansible.com/blog/project-signing-and-verification
9. https://www.redhat.com/en/interactive-labs/ansible

Red Hat Ansible Lightspeed

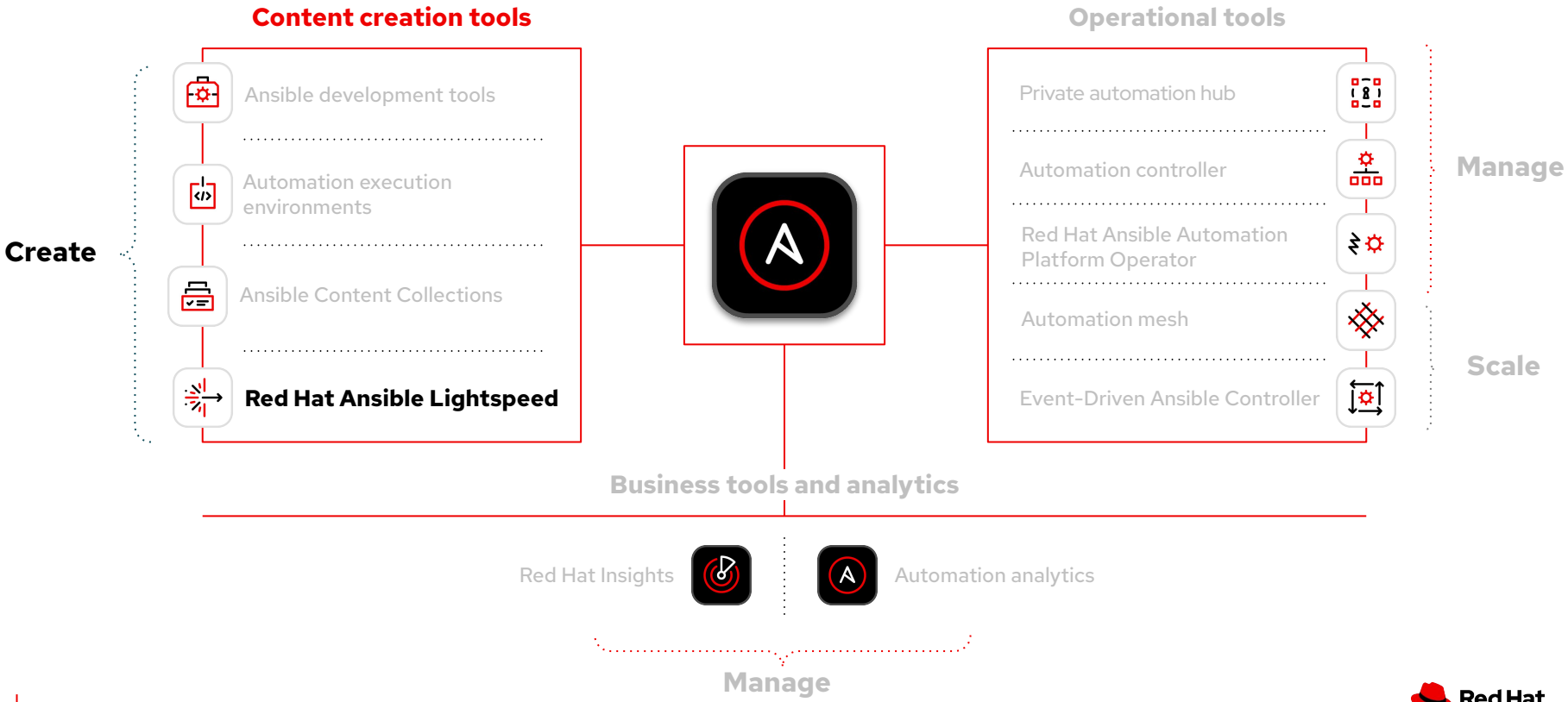
with IBM watsonx Code Assistant

Red Hat Ansible Automation Platform



Red Hat

Red Hat Ansible Automation Platform



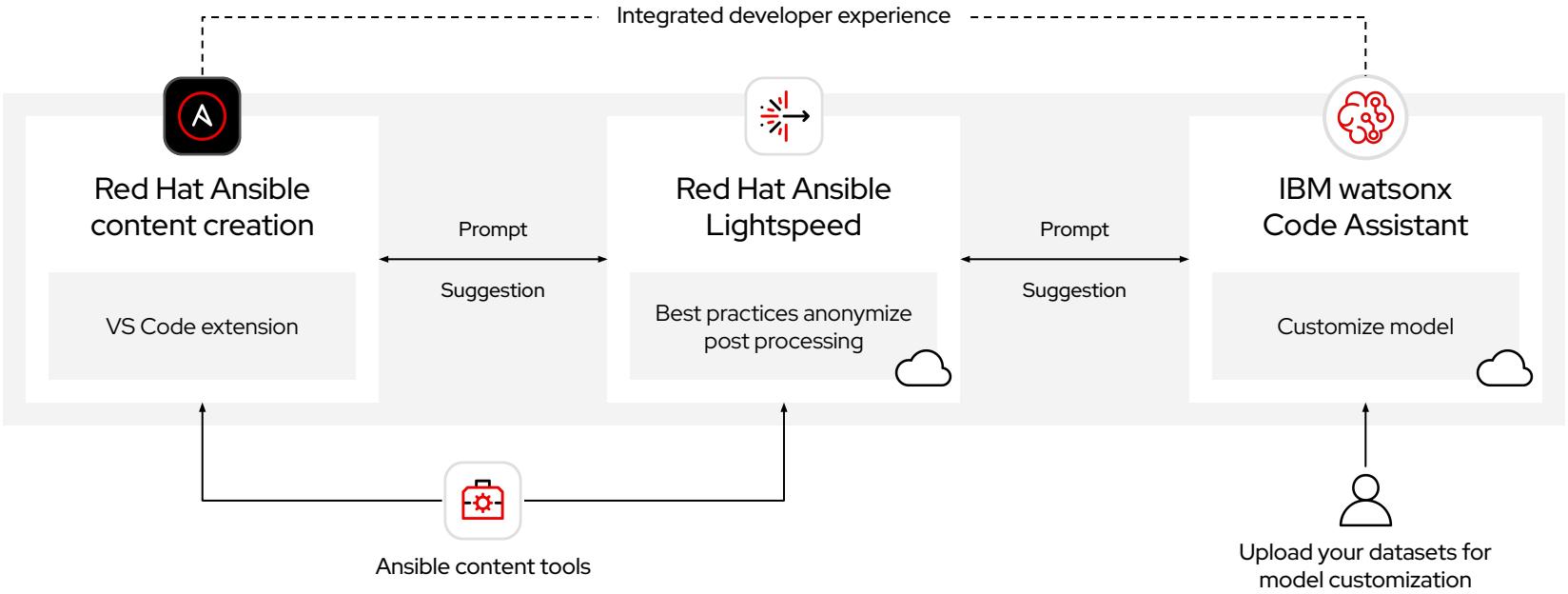
Red Hat

Red Hat Ansible Lightspeed
with IBM watsonx Code Assistant

The developer interface
Deployed natively in Visual Studio Code via the Ansible extension

The integrated service
Integration of AI services into Ansible Automation Platform with the Ansible VS Code extension

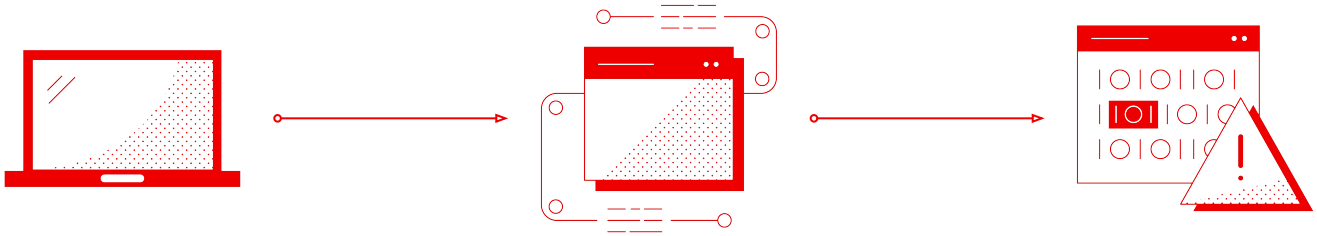
The generative AI
IBM watsonx Code Assistant is powered by the Ansible-specific watsonx.ai foundation model



Data security

- Prompt/Suggestion data is encrypted in transit and is ephemeral
- Customer Ansible playbooks and customized models are stored in customer-owned cloud object storage and are not shared with IBM, Red Hat, or any other customers

The automation code life cycle



Create

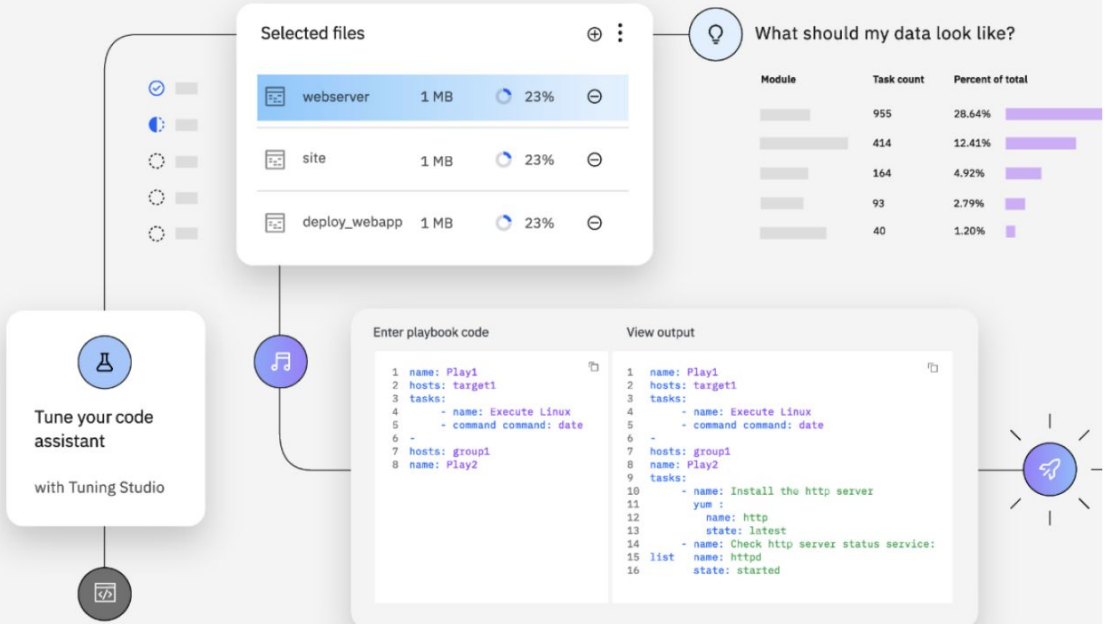
Multi-task generation
Foundational model for Ansible

Adopt

Content source matching
Post-processing capabilities

Maintain

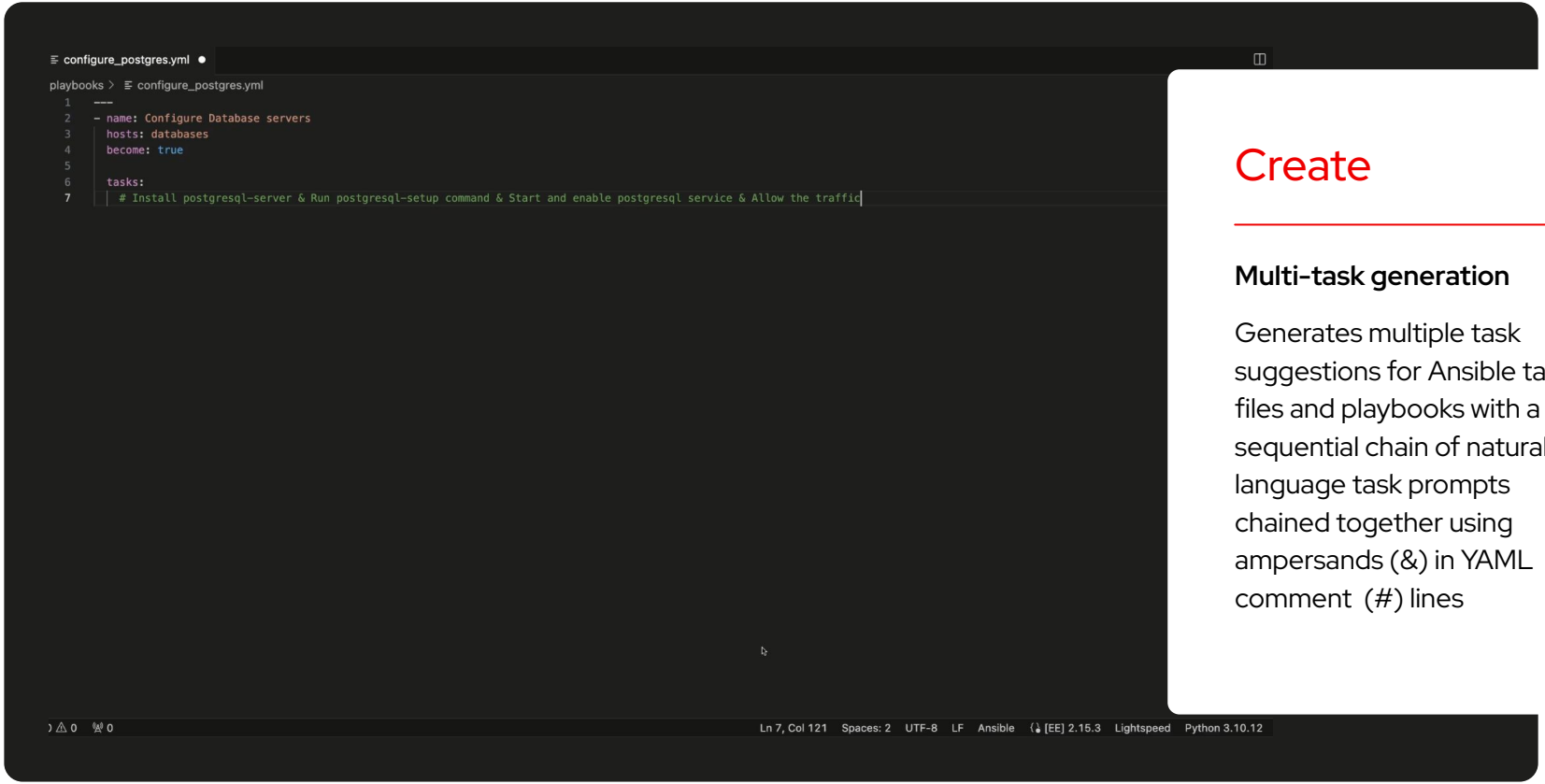
Ansible code bot
User management



Create

Foundational model for Ansible

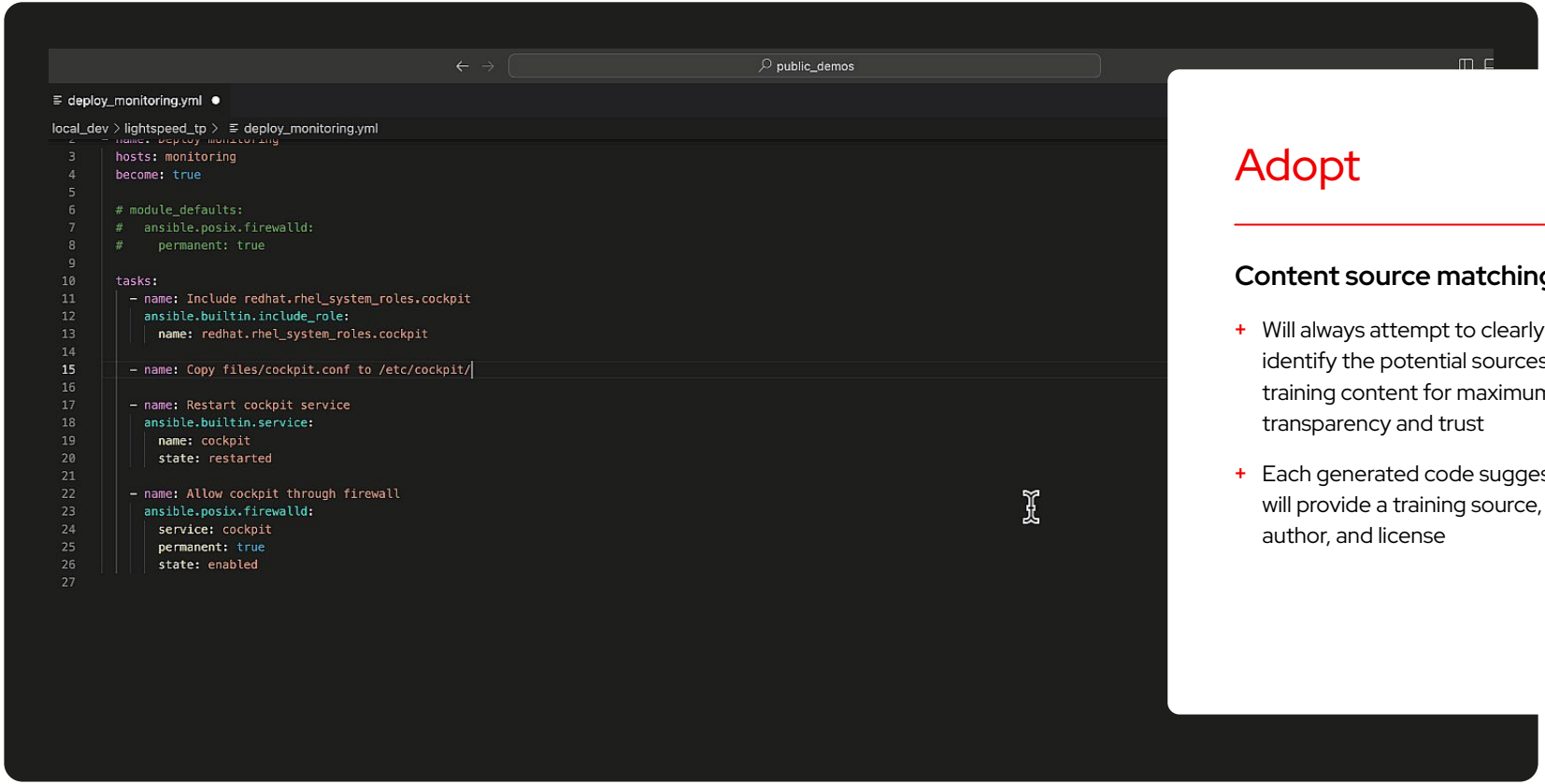
+ 3B parameter IBM watsonx Granite foundation model, trained on Ansible datasets, and infused with IBM and Red Hat automation expertise
+ Customer data is isolated to their IBM Cloud instance



Create

Multi-task generation

Generates multiple task suggestions for Ansible task files and playbooks with a sequential chain of natural language task prompts chained together using ampersands (&) in YAML comment (#) lines



Adopt

Content source matching

- Will always attempt to clearly identify the potential sources of training content for maximum transparency and trust
- Each generated code suggestion will provide a training source, author, and license

deploy_monitoring.yml

local_dev > lightspeed_tp > deploy_monitoring.yml

```yaml
---
- name: Deploy monitoring
  hosts: monitoring
  become: true

  # module_defaults:
  #   ansible.posix.firewalld:
  #     permanent: true

  tasks:
    # - name: Include redhat.rhel_system_roles.cockpit

    # - name: Copy files/cockpit.conf to /etc/cockpit/

    # - name: Restart cockpit service

    # - name: Allow cockpit through firewall
```

Adopt

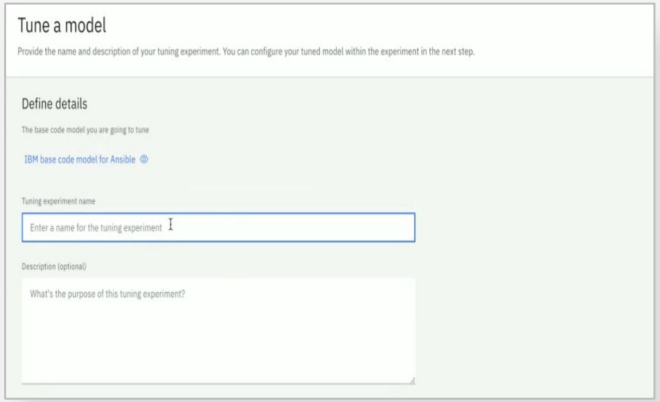
Post-processing capabilities

+ Ansible Lightspeed's post-processing capability augments model suggestions with Ansible best practices, subject matter expertise, and more
+ This processing and strong contextual awareness stems from Red Hat's unique insight into the Ansible code base, and proven experience helping customers automate at scale

Training dataset creation

```
(ansible-venv) → Ansible-OpenShift-Provisioning git:(main) ansible-content-parser https://github.com/IBM/Ansible-OpenShift-P
visioning.git /tmp/out2/ --skip-ansible-lint
INFO:pipeline:Running data pipeline
INFO:pipeline:Start scanning for 1 projects (total 269 files)
INFO:pipeline:Done
(ansible-venv) → Ansible-OpenShift-Provisioning git:(main) ls /tmp/out2
ftdata.jsonl metadata     report.txt   repository
(ansible-venv) → Ansible-OpenShift-Provisioning git:(main) ansible-content-parser . /tmp/out3/ --skip-ansible-lint
INFO:pipeline:Running data pipeline
INFO:pipeline:Start scanning for 1 projects (total 269 files)
INFO:pipeline:Done
(ansible-venv) → Ansible-OpenShift-Provisioning git:(main)
```

Model training

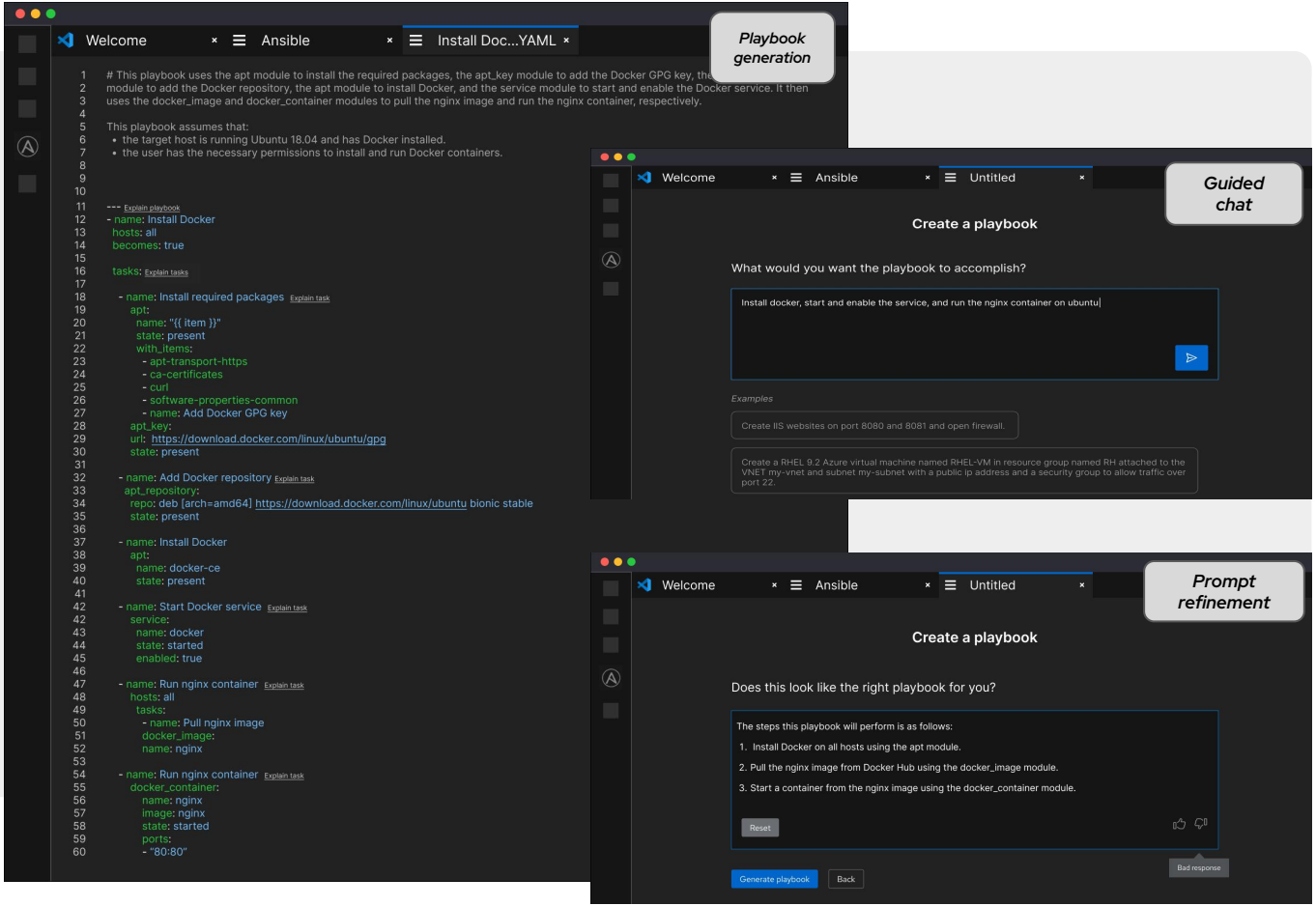


Create

Model customization / fine tuning

+ Allows Ansible Lightspeed users to leverage their existing Ansible content to train the model.
+ Code recommendations are tailored to the organizations' specific needs and automation patterns.

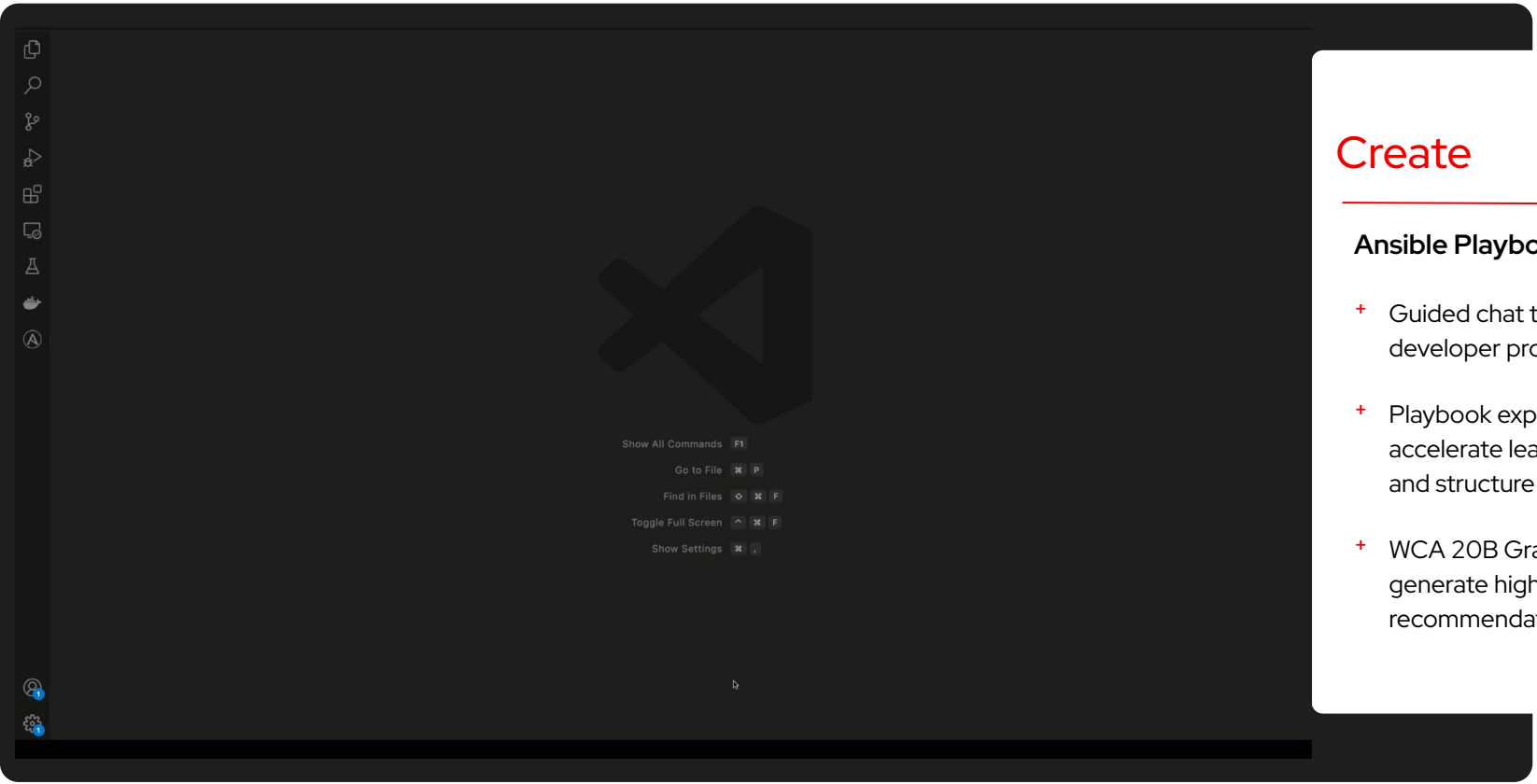
Red Hat Ansible Lightspeed



Create

Ansible Playbook generation

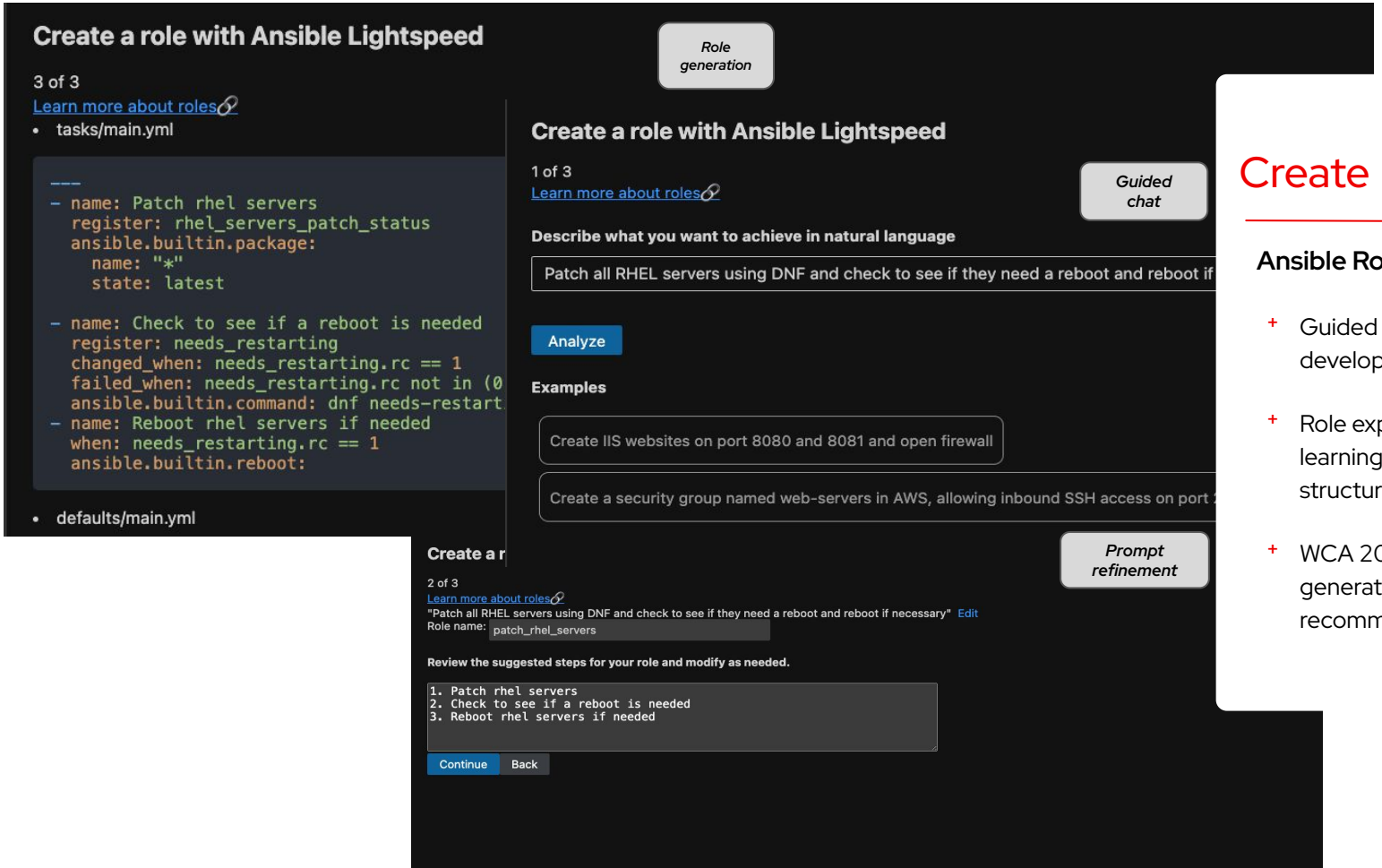
+ Guided chat to boost developer productivity
+ Playbook explanation to accelerate learning of syntax and structure
+ WCA 20B Granite model to generate high-quality recommendations



Create

Ansible Playbook generation

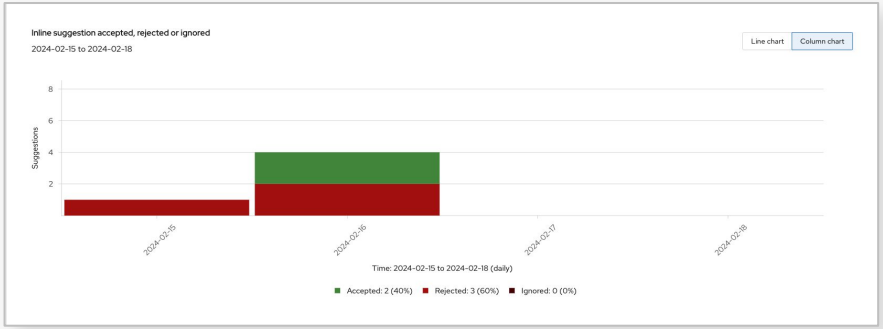
+ Guided chat to boost developer productivity
+ Playbook explanation to accelerate learning of syntax and structure
+ WCA 20B Granite model to generate high-quality recommendations

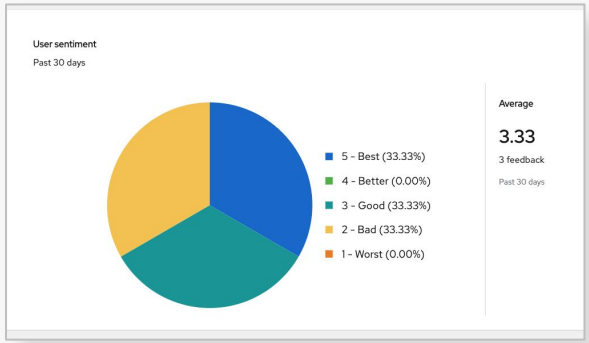


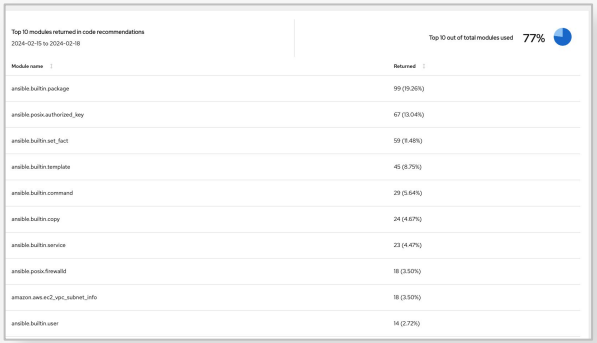
Create

Ansible Role generation

- Guided chat to boost developer productivity
- Role explanation to accelerate learning of syntax and structure
- WCA 20B Granite model to generate high-quality recommendations



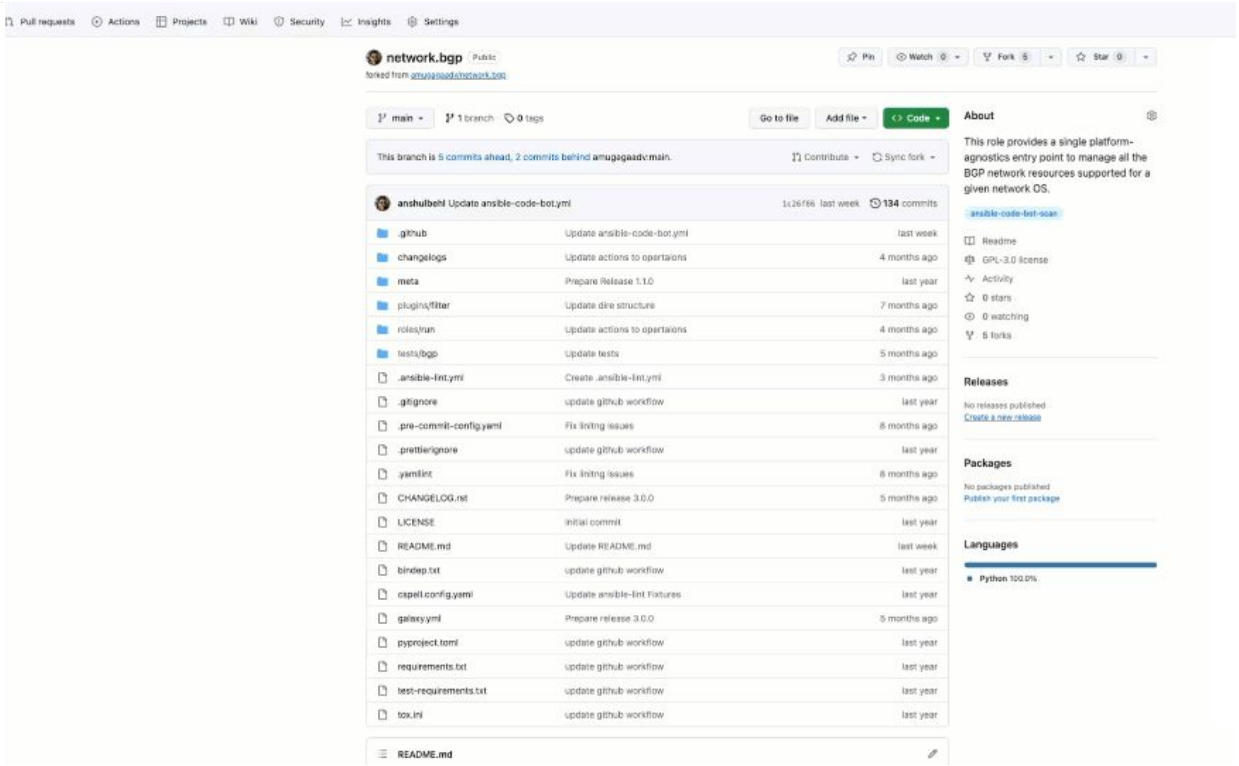




Maintain

Administrative dashboard

+ Monitoring metrics and usage reporting
+ Understand usage patterns across teams
+ Gain insights to ensure successful adoption
+ Accessed in the Red Hat Hybrid Cloud Console



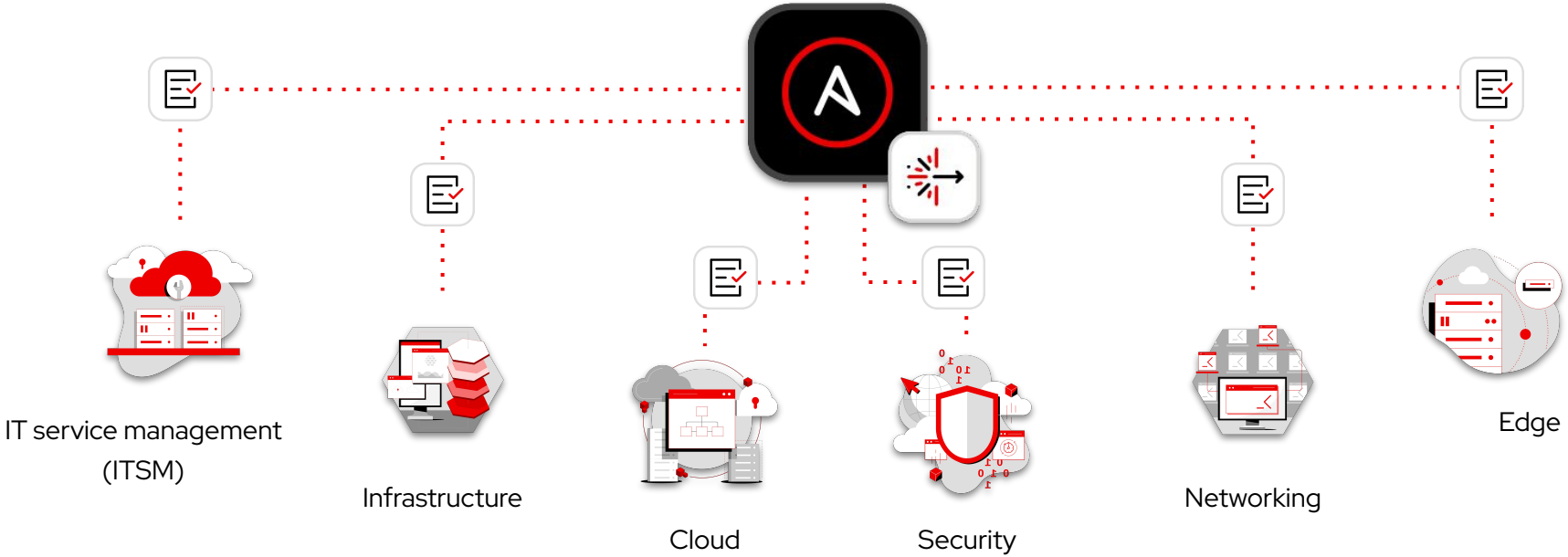
Maintain

Ansible code bot

+ Integrates natively with user-selected Git repos, scans existing Ansible content, and automatically submits pull requests with code improvement recommendations
+ Makes it easier for teams to review, test, and apply improvements to the automation code base

Accelerate automation content development across domains

More efficient, consistent creation



Red Hat

Benefits for the entire IT automation team

Regardless of skill level

AUTOMATION

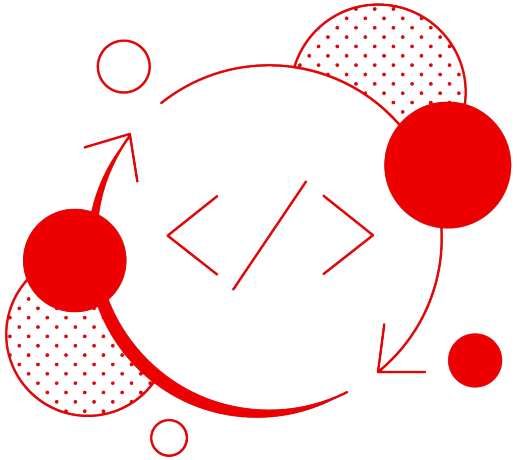
Novices

+ Reduces the Ansible and YAML learning curve
+ Helps ensure that the automation content created follows best practices
+ Increases skill development and confidence

AUTOMATION

Subject matter experts

+ Bridges the gap between automation ideas and Ansible content creation
+ Uses a data model trained across automation domains, extending expertise



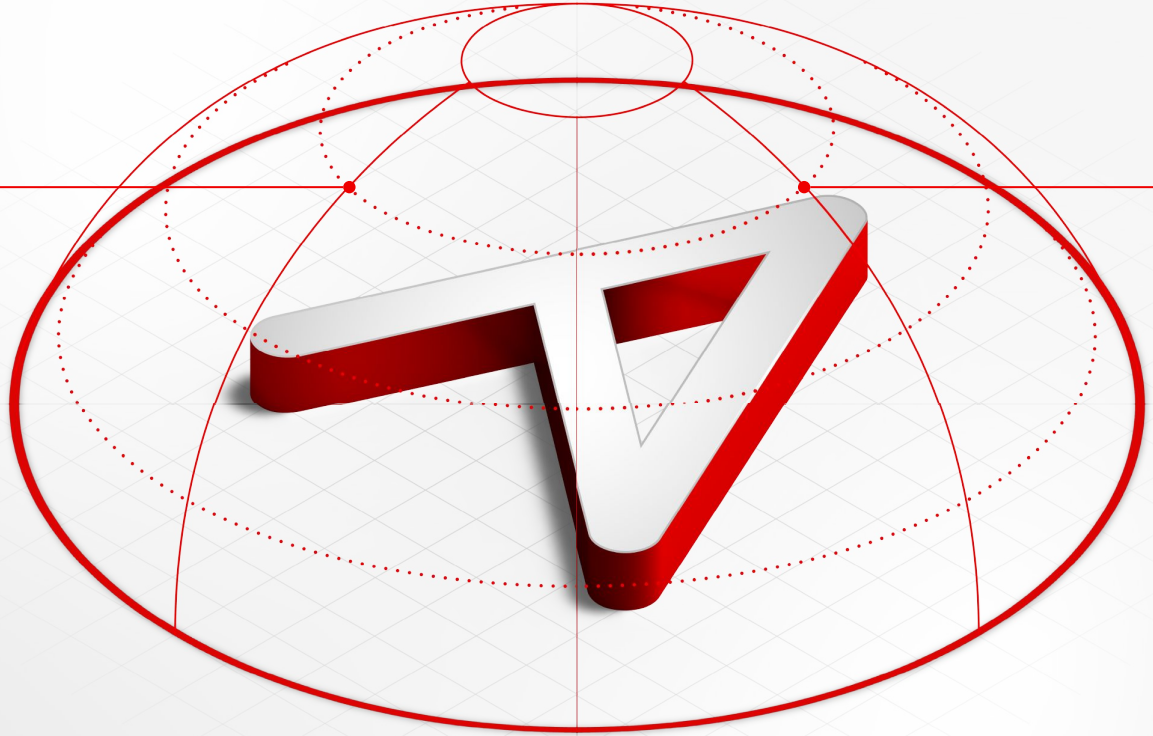
AUTOMATION

Developers

Improves productivity for Ansible Playbook creation

Integrates with existing workflows and tools for efficient adoption

Streamlines ongoing code maintenance with Ansible code bot tools



AUTOMATION

Operators

Adheres to Ansible best practices for greater confidence when running at scale

Integrates with Ansible Automation Platform for a familiar experience

Using generated AI reduced playbook development effort by over 50% in IBM CIO Organization Pilot

IBM watsonx Code Assistant for Red Hat Ansible Lightspeed technical preview generated over half of the playbook content

60%

Of IBM's Ansible Playbook content was automatically generated by watsonx Code Assistant for Red Hat Ansible Lightspeed in tech preview

10X

Expected increase in members of IBM CIO team who can help produce Ansible Playbooks going forward

"Our team of people who could produce Ansible Playbooks before the technical preview program began was around 10. Once the Enterprise Version of watsonx Code Assistant for Red Hat Ansible Lightspeed becomes Generally Available,
we can ask our entire team of 100 to product Ansible content."

Bob Epstein
Leader
IBM CIO Hybrid Cloud Platforms

Source: https://www.ibm.com/case-studies/ibm-cio-organization-wca

Transforming the way developers learn and work

How IBM Consulting achieved a 30% reduction in Ansible Playbook development effort while maintaining quality, compliance, and resiliency

Up to
45%

Improvement in initial build productivity for Ansible Playbooks

Decrease

In time it takes a new user to get up to speed and be productive in Ansible

" During the technical preview for watsonx Code Assistant for Red Hat Ansible Lightspeed, we observed initial build productivity improvements in the range of 20%–45%. As we move to GA, even more gains in productivity are expected. We also believe there's additional downstream productivity that has not been fully quantified yet. Not only are we aiming to accelerate the development phase for Ansible automations and shorten time-to-value for our clients after GA, but we intend for the quality of the content to be higher."

Gerry Leitão
Partner and Global Hybrid Cloud Automation
IBM Consulting

Source: https://www.ibm.com/case-studies/ibm-consulting-watsonx-code-assistant

Ansible Lightspeed service offering roadmap

Now

Model customization / tuning
Use watsonx Code Assistant to create custom Ansible recommendation models

Full playbook generation
Chat-based interactive interface that allows for better automation generation

On-premise deployment
For local and air-gapped environments

Playbook explanation
Understand the impact of your automation code.

Admin dashboard
Reporting and usage metrics

Simplified activation of WCA service
90-day trial

Near term

Integrations with Red Hat AI portfolio
Enable architecture(s) that utilize open-sourced models supported by Openshift AI and RHEL AI .

Add'l Ansible code bot Git repo support
GitLab

Ansible code bot pull request explanations
Understand the impact of suggested repo changes.

Long term

Chat bot for VS Code extension
Full-featured chat interface embedded into the coding experience

AAP chat bot
Integration of chat interface into Ansible automation controller

Content discovery
Find existing Ansible content instead of writing from scratch

Rulebook recommendations
Support recommendations for Event-Driven Ansible

Add'l Ansible code bot Git repo support
Azure DevOps, Bitbucket, etc.

Red Hat

Ansible Lightspeed resources

- Ansible Lightspeed homepage ▸
- Getting started video ▸
- Getting started blog ▸
- Datasheet ▸
- Self-paced interactive lab ▸

Red Hat

Red Hat Summit

AnsibleFest

Mark your calendar. Make your mark.

We're headed to Boston, Massachusetts, for Red Hat® Summit and AnsibleFest. Join us May 19–22, 2025, to connect with community, build new skills, and make your mark on open source.

Register at red.ht/summit

Red Hat

Enter code: Automate25
during registration to unlock a $200 discount per pass!

Red Hat
Ansible Automation
Platform

Thank you

Red Hat is the world's leading provider of enterprise open source software solutions. Award-winning support, training, and consulting services make
Red Hat a trusted adviser to the Fortune 500.


linkedin.com/company/red-hat


youtube.com/user/RedHatVideos


facebook.com/redhatinc


twitter.com/RedHat

Red Hat