



ANSIBLE LIGHTSPEED

SPEED UP YOUR PLAYBOOK
DEVELOPMENT

Alex Bron

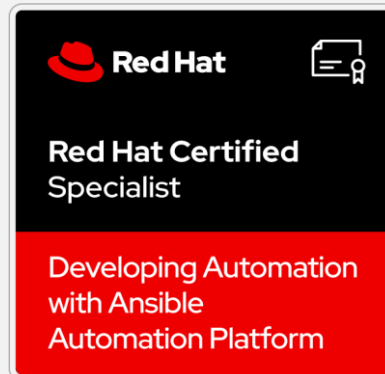
LOOK WHO'S TALKING



ALEX BRON

Red Hat Certified Architect

Technical coach Conclusion Xforce automation team
Member of the Red Hat Accelerators program



SPEED UP YOUR PLAYBOOK DEVELOPMENT WITH ANSIBLE LIGHTSPEED

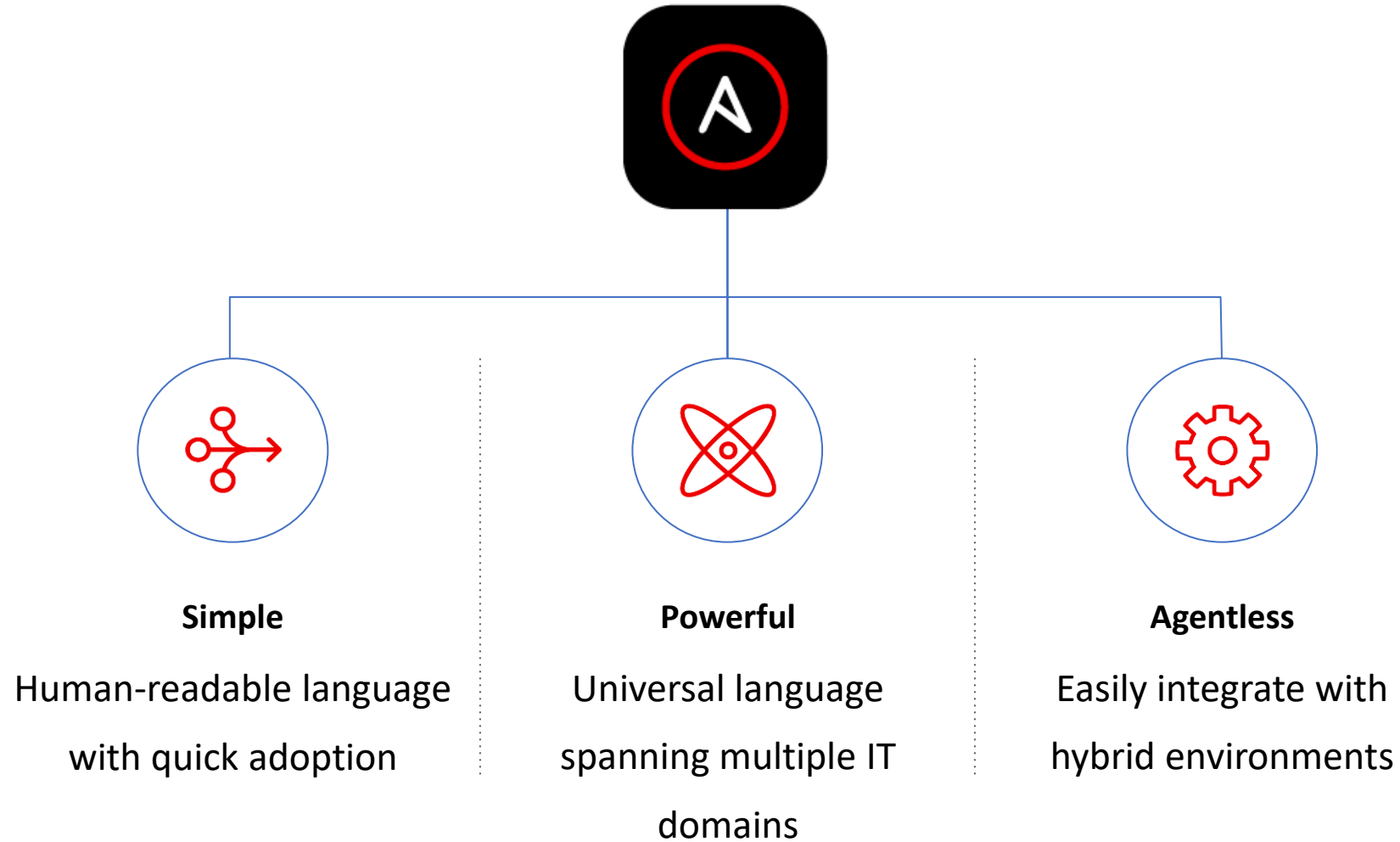
CONCLUSION
XFORCE

**Visual Studio Code
for Ansible code
creation**

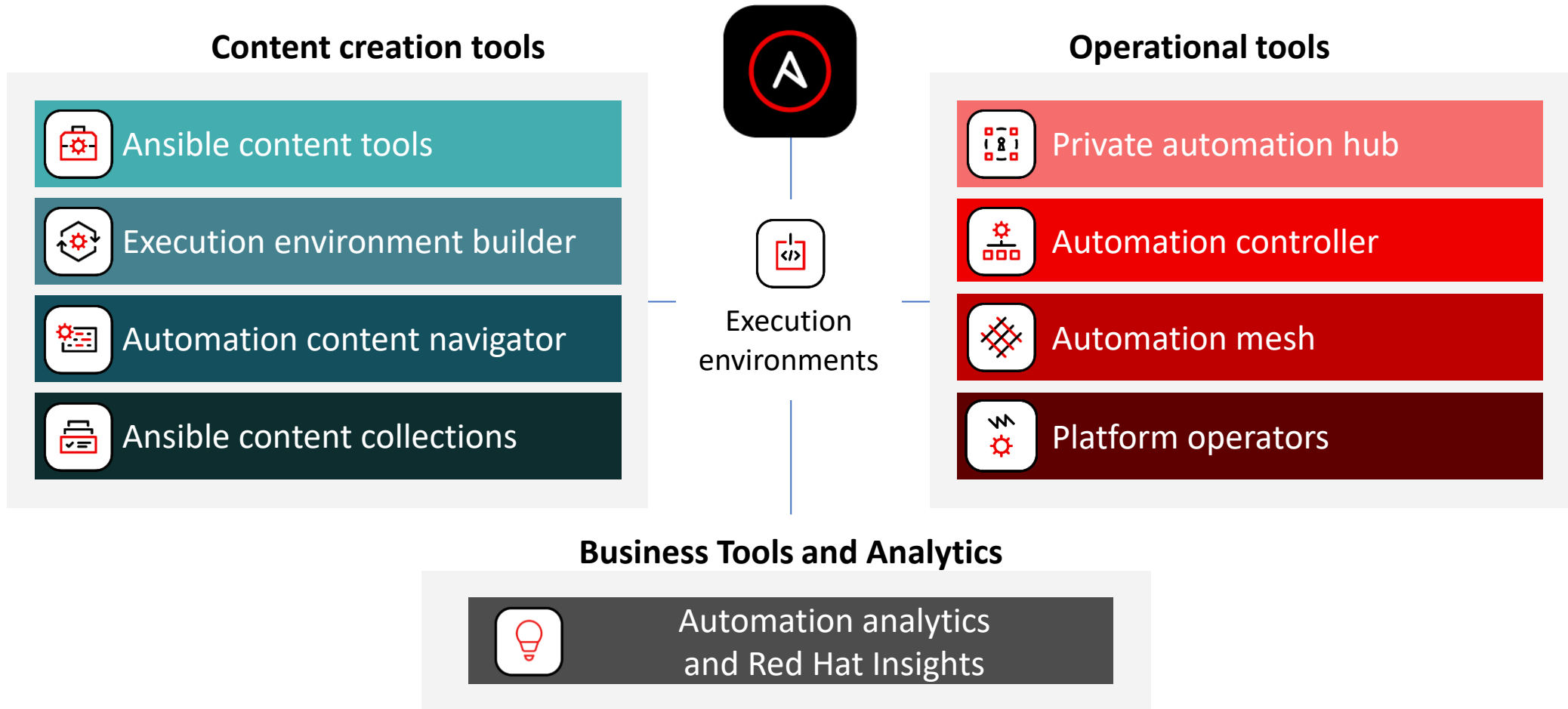
Ansible Lightspeed

Live demo

Ansible is the **de facto automation language**.



An integrated solution for the enterprise.



Vi (improved)

- ▶ Syntax highlighting (for yaml)
- ▶ Auto indentation
- ▶ Plugins for in-line encryption (vaulting)

- ▶ But: who likes vi ...

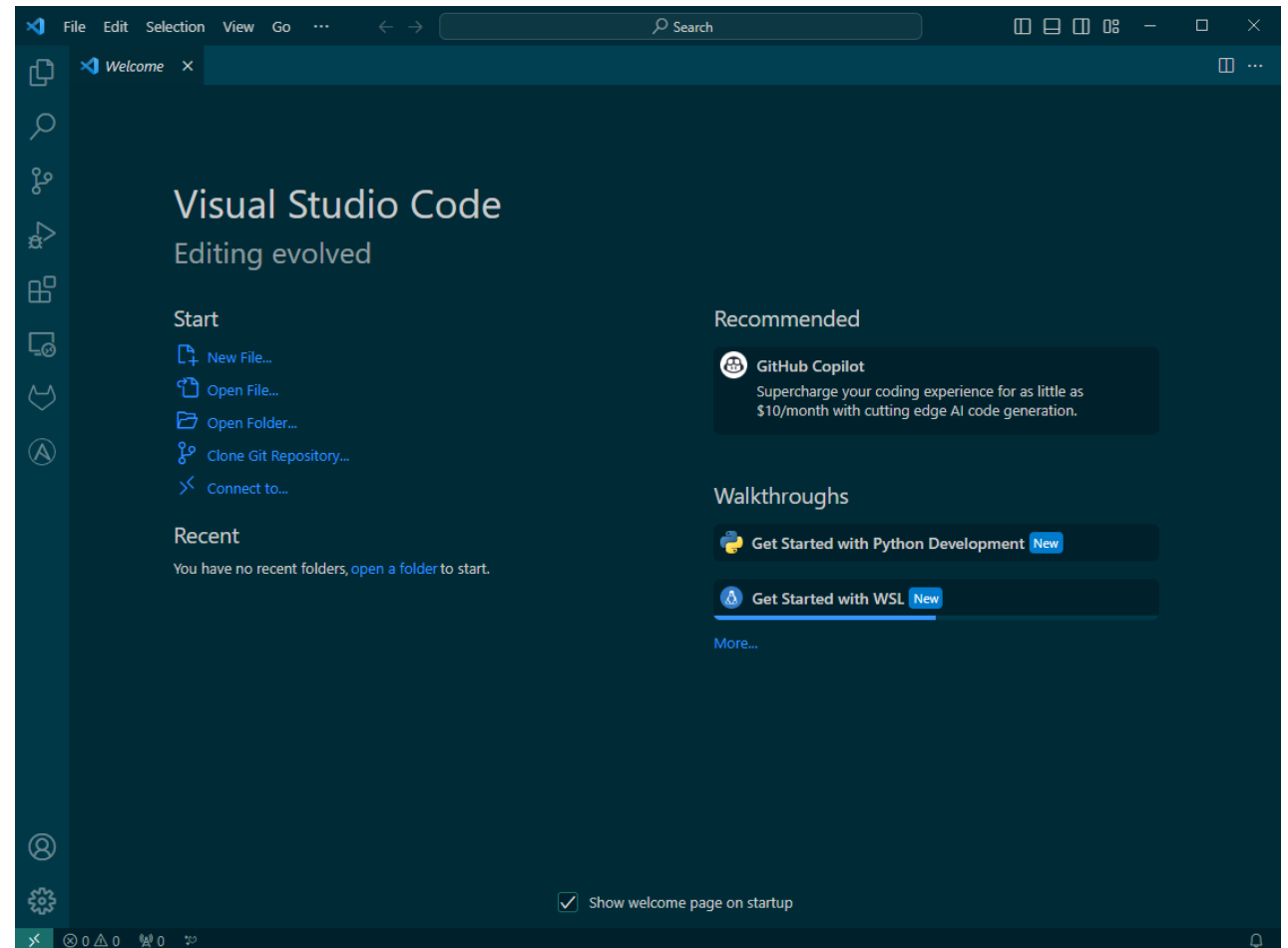
```
1  |--
2  - import_playbook: library/check_region.yml
3    tags:
4      - always
5
6  - name: Update and reboot the virtual machines
7    hosts: "{{ region | default('test') }}"
8    serial: 1
9    tags:
10     - never
11     - vm
12    tasks:
13     - name: Full block only on vms
14       block:
15
16         - name: "Perform the dnf update" # noqa package-latest
17           become: true
18           ansible.builtin.package:
19             name: "*"
20             state: latest
21             disablerepo:
22               - epel
23               - epel-modular
24
25         - name: "Check if a reboot is required"
26           become: true
27           ansible.builtin.command: needs-restarting -r
28           register: bootrequest
29           ignore_errors: true
30           failed_when: false
31           changed_when: false
32
33         - name: "Reboot when package updates are installed"
34           become: true
35           ansible.builtin.reboot:
36             msg: "Rebooting after OS package updates"
37             reboot_timeout: 3600
38             when: bootrequest.rc == 1 and not ansible_connection == "local"
39
40       when: ansible_virtualization_role == "guest"
41
"update_all.yml" 76L, 2122C
```



Visual Studio Code (vscode)

Extensions:

- ▶ Remote Development
 - ▶ Ssh
 - ▶ WSL
- ▶ Ansible
- ▶ Gitlab Workflow





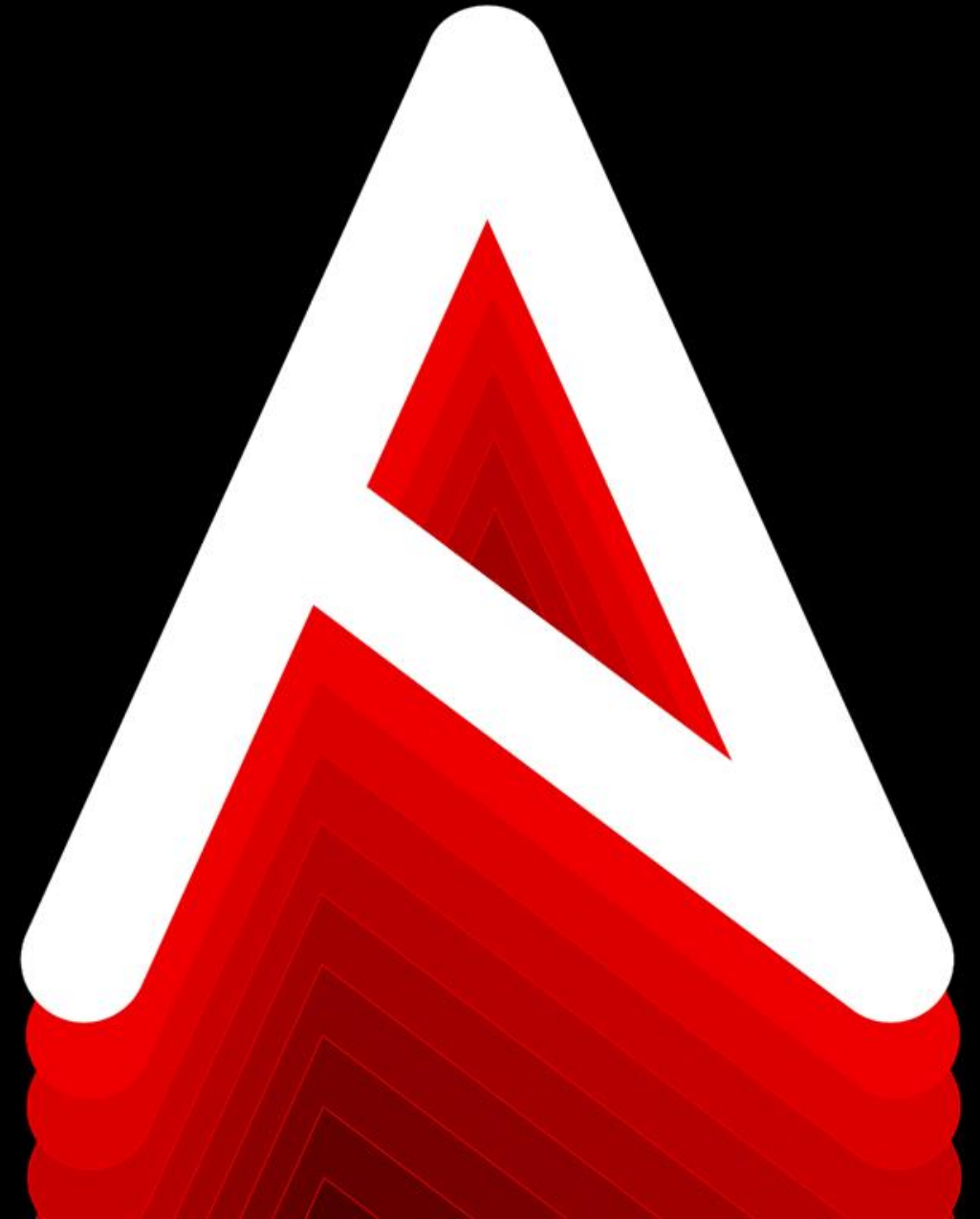
Technical Preview overview:

Ansible Lightspeed

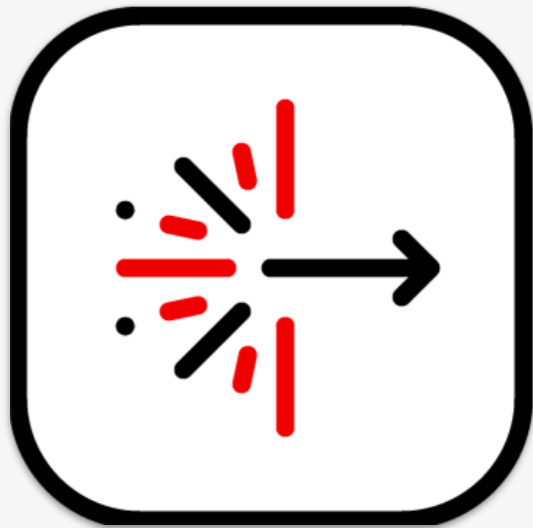
with IBM Watson Code Assistant

Ansible Business Unit

Product Marketing | Technical Marketing | Product Management



What is Ansible Lightspeed with IBM Watson Code Assistant?



Ansible Lightspeed with IBM Watson Code Assistant is a generative AI solution engineered to help individuals, teams, and organizations automate faster.

By integrating developer tooling, access to Ansible-focused foundation models, and Red Hat + IBM automation expertise, Ansible Lightspeed streamlines and enhances the Ansible content creation experience.

Primary capability in Tech Preview: **Task Generation**

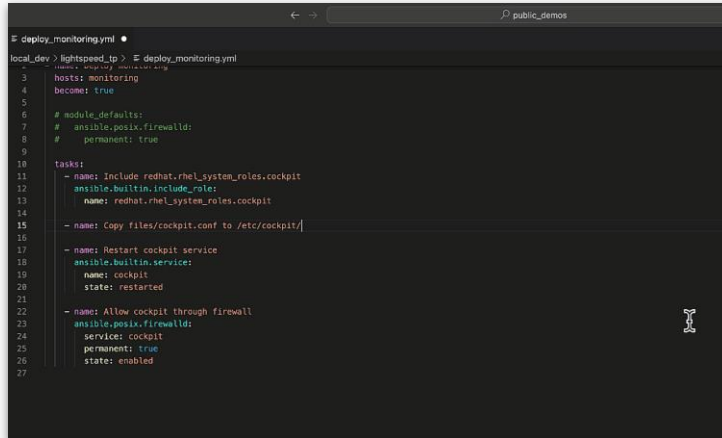
```
⌵ deploy_monitoring.yml ×
local_dev > lightspeed_tp > ⌵ deploy_monitoring.yml
1  ---
2  - name: Deploy monitoring
3    hosts: monitoring
4    become: true
5
6    # module_defaults:
7    #   ansible.posix.firewalld:
8    #     permanent: true
9
10   tasks:
11     # - name: Include redhat.rhel_system_roles.cockpit
12
13     # - name: Copy files/cockpit.conf to /etc/cockpit/
14
15     # - name: Restart cockpit service
16
17     # - name: Allow cockpit through firewall
```

Generate **tasks** for a playbook or role from natural language task description

Ansible Lightspeed **suggests**
Ansible Tasks within Ansible
Playbook and Ansible Task files

Other notable features:

Content source matching

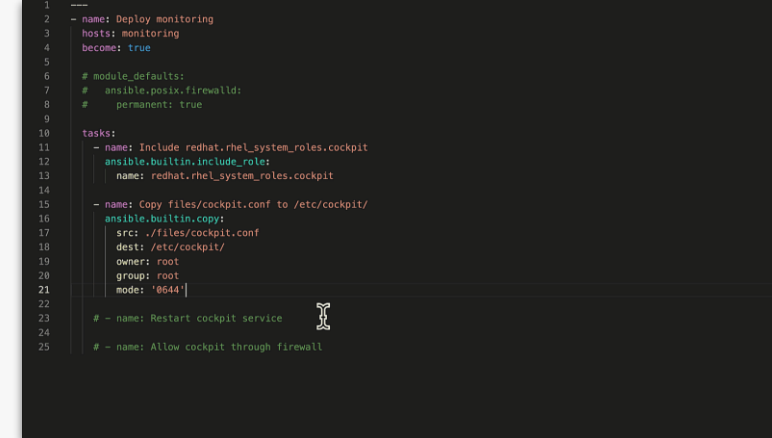


```
1 local_dev > lightspeed_tp > E deploy_monitoring.yml
2 # name: Deploy monitoring
3 hosts: monitoring
4 become: true
5
6 # module_defaults:
7 #   ansible.posix.firewall:
8 #     permanent: true
9
10 tasks:
11   - name: Include redhat.rhel_system_roles.cockpit
12     ansible.builtin.include_role:
13       name: redhat.rhel_system_roles.cockpit
14
15   - name: Copy files/cockpit.conf to /etc/cockpit/
16
17   - name: Restart cockpit service
18     ansible.builtin.service:
19       name: cockpit
20       state: restarted
21
22   - name: Allow cockpit through firewall
23     ansible.posix.firewall:
24       service: cockpit
25       permanent: true
26       state: enabled
27
```

Ansible Lightspeed addresses generative AI transparency issues with the **content source matching** feature.

Each suggestion will include **potential source, its author and license**. Our goal with this is to ensure that users, contributors, and organizations are able to employ a **generative AI solution they trust**.

Post processing

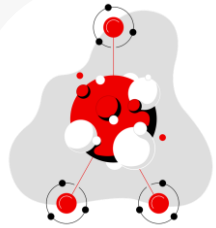


```
1 # name: Deploy monitoring
2 - name: Deploy monitoring
3   hosts: monitoring
4   become: true
5
6   # module_defaults:
7   #   ansible.posix.firewall:
8   #     permanent: true
9
10   tasks:
11     - name: Include redhat.rhel_system_roles.cockpit
12       ansible.builtin.include_role:
13         name: redhat.rhel_system_roles.cockpit
14
15     - name: Copy files/cockpit.conf to /etc/cockpit/
16       ansible.builtin.copy:
17         src: ./files/cockpit.conf
18         dest: /etc/cockpit/
19         owners: root
20         group: root
21         mode: '0644'
22
23     # - name: Restart cockpit service
24
25     # - name: Allow cockpit through firewall
```

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 our ability to leverage the vast Ansible ecosystem, along with Ansible's flexibility as a tool.

Key differentiators



Transparency, collaboration, and choice

- ▶ **All content** being used is covered by **approved open source licenses**.
- ▶ **Content source matching** and transparency will always be a key priority.
- ▶ **Contributors will have a choice** as to whether or not their work is used to fine-tune the model.



A singular focus on automation

- ▶ **Purpose-built** to accelerate your Ansible adoption.
- ▶ **Uniquely positioned** to offer an **enterprise-grade** AI solution for Ansible.
- ▶ **Laser-focussed** on delivering **high-quality, contextual suggestions**.



Meeting developers where they work

- ▶ **Natively integrated** into Ansible Visual Studio Code extension.
- ▶ **Augments and enhances** developer experience. Non-disruptive.
- ▶ **Works collectively with Ansible Content tools**, such as Ansible Lint

3 things to know about the Ansible Lightspeed Tech Preview

- 1** It is now available for **ALL Ansible users, for free**. It does not require an AAP subscription. All an Ansible user will need is a GitHub ID and the Ansible VSCode extension.
- 2** This is a limited service, with a **one primary capability: task generation**. The full featured Ansible Lightspeed with IBM Watson Code Assistant product offering is in development, and will be available later this year.
- 3** This is a **self-supported service**. Users should refer to the [Ansible Lightspeed technical preview docs on GitHub](#) for assistance.

Tech preview snapshot



Ansible Lightspeed with IBM Watson Code Assistant: Tech Preview

Access	VS Code extension
Capabilities	Task generation
Data sets	Ansible-specific “Training” data set comprised of Ansible Galaxy content + other open sources of data, and infused with additional Red Hat and IBM automation subject matter expertise.
Vendor	Red Hat
Pricing	Free
Availability	June 2023
For more info	redhat.com/ansible-lightspeed



TIME FOR A DEMO

THANK YOU!

Alex Bron

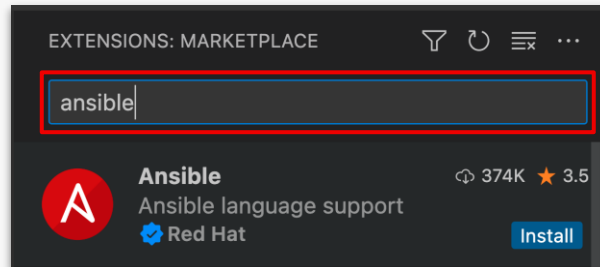


Getting connected via the Ansible VS Code extension

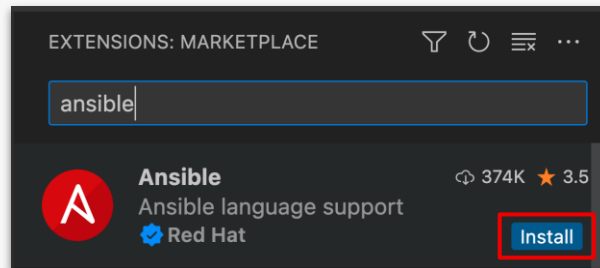
1

Install the Ansible VS Code extension.

First, install the Ansible VS Code extension by searching for “ansible” in the VS Code Marketplace.



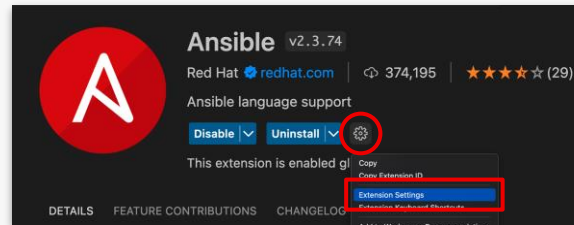
Click “install” on the Ansible VS Code extension published by Red Hat.



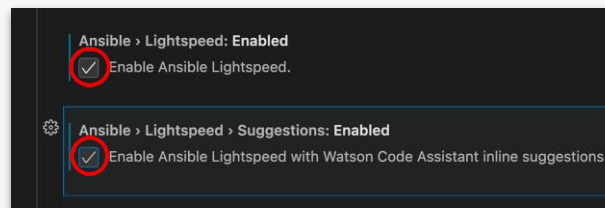
2

Enable Ansible Lightspeed in the Ansible VS Code extension.

Enable the Ansible Lightspeed service in the extension by clicking the “gear” icon and selecting “Extension Settings.”



Enable Ansible Lightspeed in VS Code “Workspace” settings or configure it in the “User” settings based on preference.

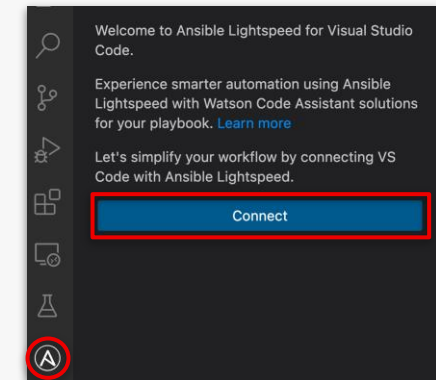


Then, enable two checkboxes; “Ansible Lightspeed enabled” and “Enable Ansible LightSpeed with Watson Code Assistant inline suggestions.”

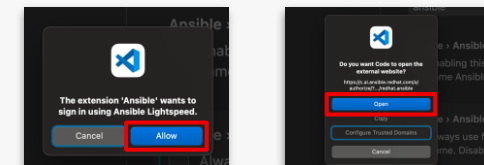
3

Connect and Sign into Lightspeed Service.

Click on the Ansible “A” in the VS Code activity bar on the left hand-side of editor.



Click “Connect”.

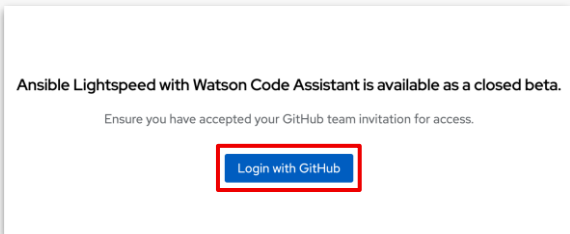


In the new pop up window, confirm that you want to sign in to the Ansible Lightspeed service. Click “Allow” and “Open” in the following dialog box.

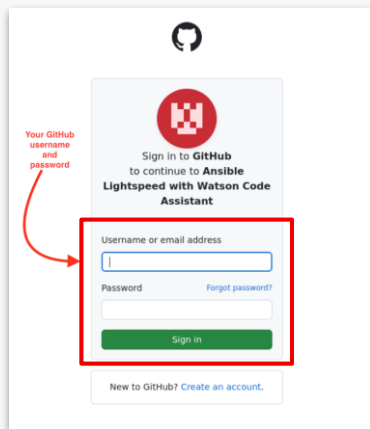
Setting up access via GitHub

4

Logging into Github and entering credentials.

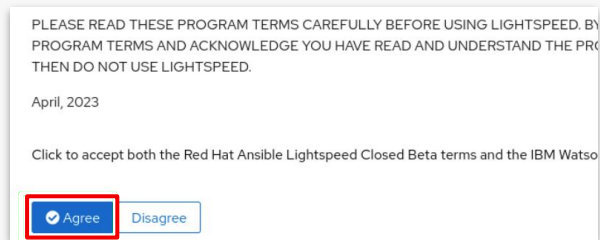


A new window will open in default browser. Click on the “Login with GitHub” button and enter your GitHub credentials.

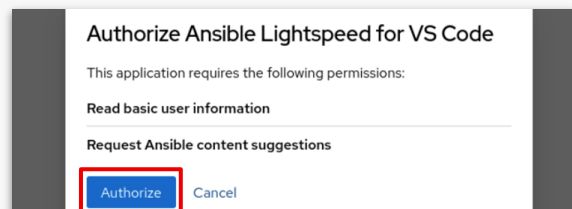


5

Accepting Terms & Conditions and Authorize Lightspeed.



Read the Ansible Lightspeed Technical Preview terms and conditions and click “Agree”.

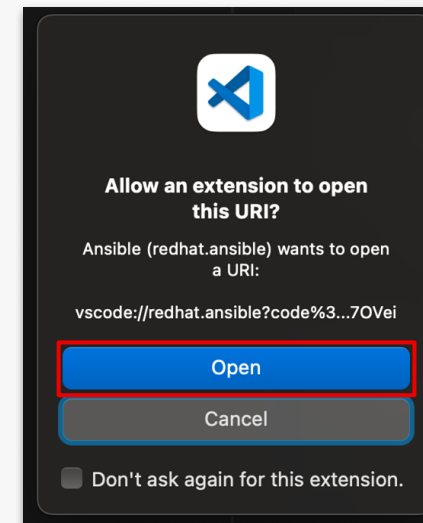


Authorize Ansible Lightspeed for VS Code by clicking the “Authorize” button and following the browser prompts to redirect you back to VS Code.

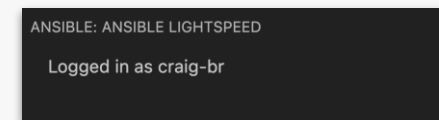
6

Final Step:

Click “Open” in the VS Code confirmation dialog box.



Ansible LightSpeed is now configured in VS Code.



Providing feedback

We encourage all Tech Preview users to provide feedback about their experience.

This will help improve Ansible Lightspeed in multiple ways:

- ▶ Ensure that the service is meeting your automation development needs
- ▶ Improve the quality and relevance of recommendations from our foundation model
- ▶ Inform our engineering approach to the forthcoming commercial version of Ansible Lightspeed with Watson Code Assistant

