



Ansible Lightspeed with Watson Code Assistant Lab

Craig Brandt

Principal Technical Marketing Manager

Ansible



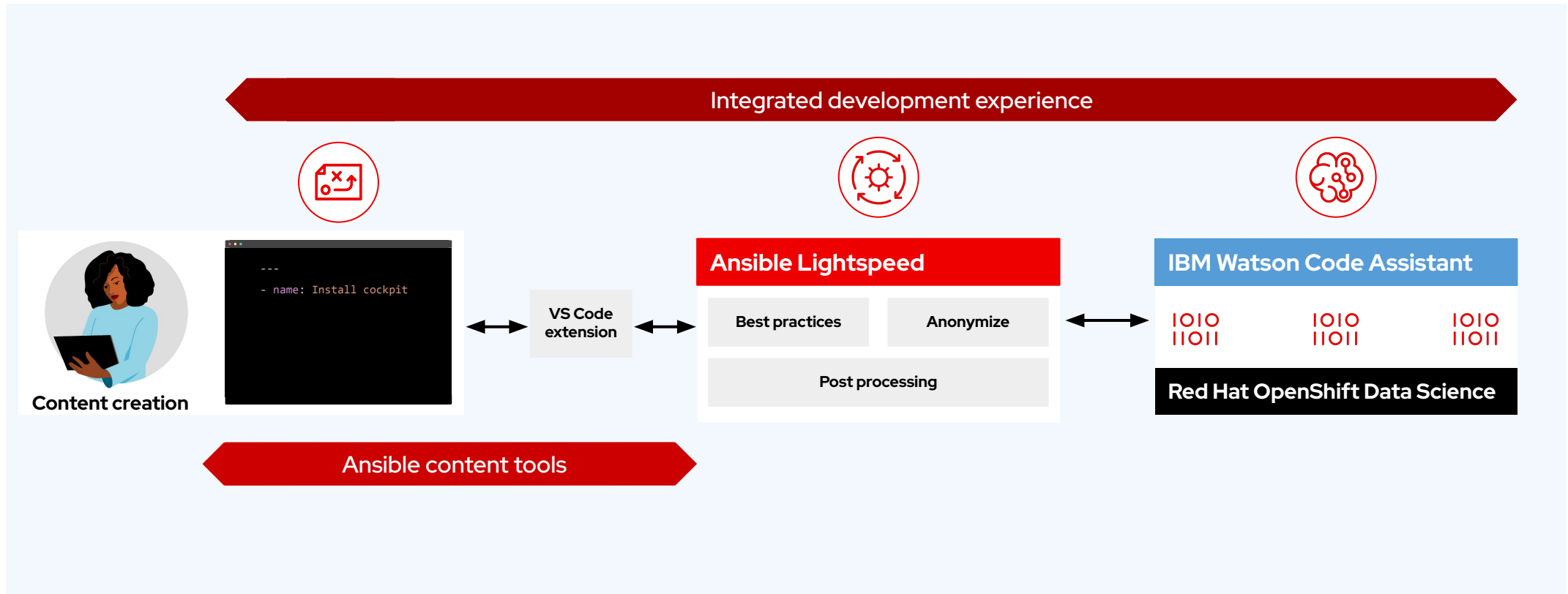
Ansible Lightspeed

with IBM **Watson** Code Assistant

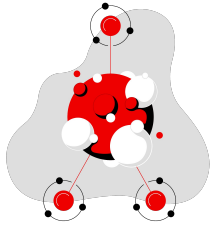
What we'll cover...

- ▶ What is Ansible Lightspeed?
- ▶ Why Ansible Lightspeed?
- ▶ Ansible Lightspeed lab overview
- ▶ Lab tips and tricks

What is Ansible Lightspeed?



Why Ansible Lightspeed?



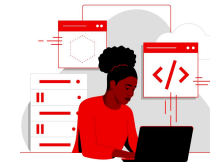
Transparency, collaboration, and choice

- ▶ **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

It starts with Ansible Playbooks

Ansible Playbook tasks

- ▶ **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: Deploy RHEL monitoring  
  hosts: monitoring  
  become: true  
  
  tasks:  
    - name: Install cockpit on RHEL  
      when: ansible_distribution == 'RedHat'  
      ansible.builtin.package:  
        name:  
          - cockpit  
        state: present
```

Ansible Lightspeed will transform content creation

```
> Generate a  
playbook task 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 it's  
impact
```

**Content
Explanation**

Roadmap features

Optional section marker or title

Lab overview

Intro to Ansible Lightspeed lab overview

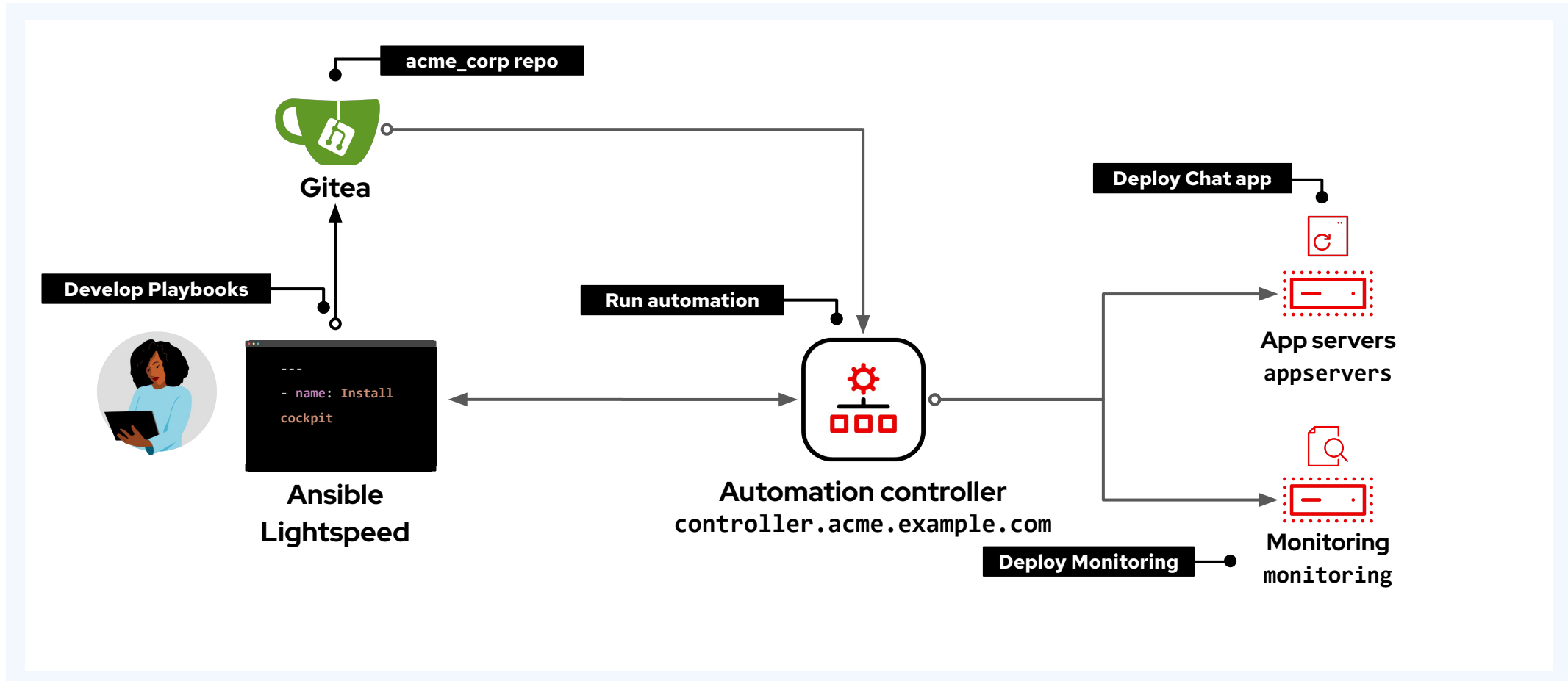
Notes

- ▶ Have your **GitHub ID** ready.
- ▶ Remember it's still in **Beta**!
- ▶ Lab is **55 minutes** long.
- ▶ **Configure Ansible Lightspeed** in VS Code.
- ▶ **Generating** Ansible Playbook tasks.
- ▶ Running Playbooks in **automation controller**.
- ▶ Bonus Mission! - **Orchestration using Workflows**.
- ▶ Ansible Lightspeed Closed Beta **Playground**.



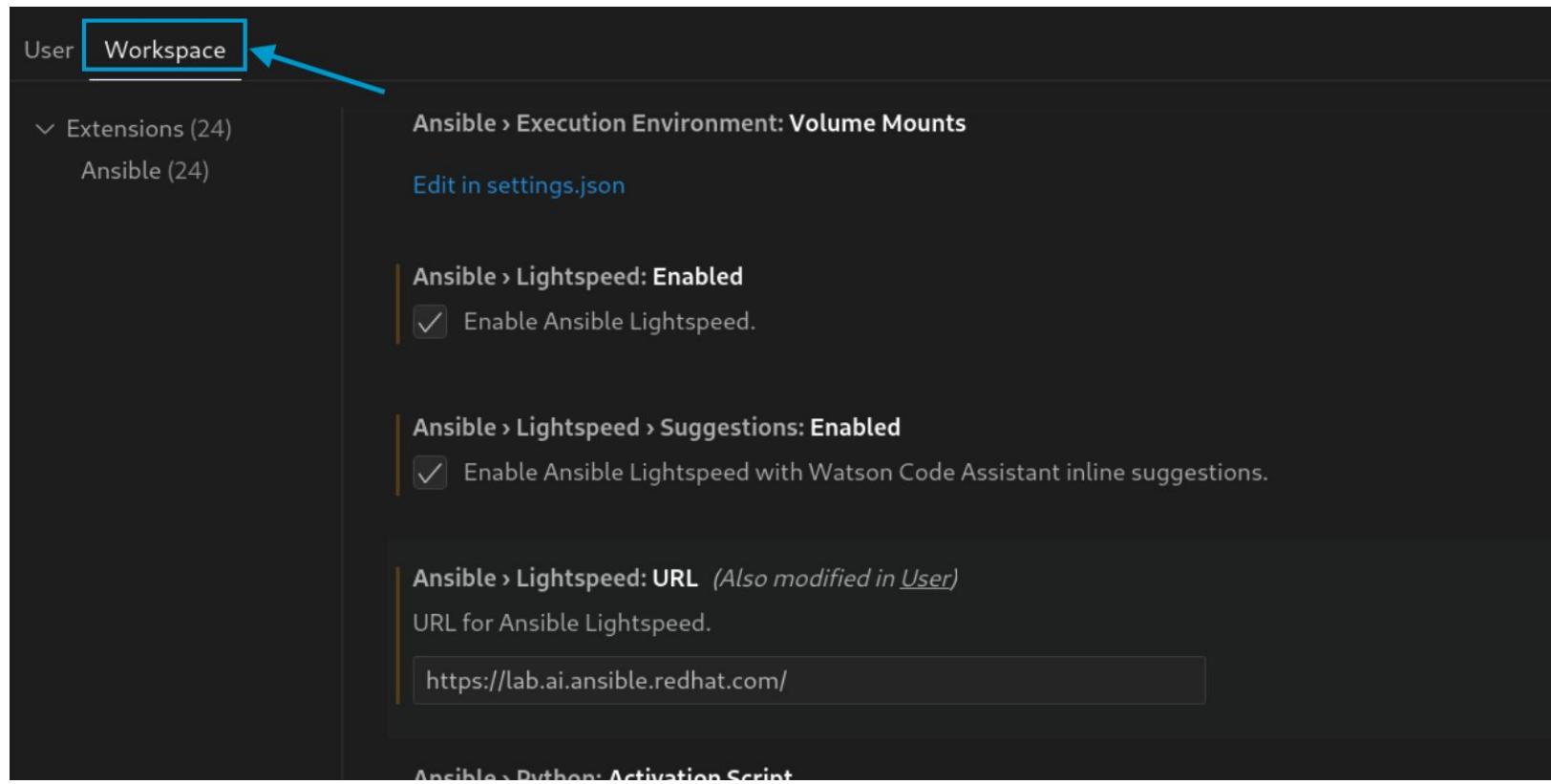
Lab environment

ACME Corp deploy Chat application and monitoring



Enabling Ansible Lightspeed

Configuring Ansible Lightspeed in Visual Studio code




Logging into the Ansible Lightspeed service

Using GitHub ID and authorization

Ansible Lightspeed with Watson Code Assistant is available as a closed beta.

Ensure you have accepted your GitHub team invitation for access.



Login with GitHub

Generating Ansible Lightspeed suggestions

The Ansible Lightspeed experience

```
---  
- name: Permit HTTPS through firewall
```

Generate Ansible Lightspeed suggestion

1. Write prompt in Ansible task description.
2. Press ENTER at the end of task description line to request Ansible Lightspeed suggestion.

```
---  
- name: Permit HTTPS through firewall  
  ansible.posix.firewalld:  
    service: https  
    permanent: true  
    state: enabled
```

Accept / Reject Ansible Lightspeed suggestion

3. Press TAB to accept suggestion. Playbook task is automatically generated.
4. Press ESC to reject suggestion. Rejected suggestion analytics train the model.

```
---  
- name: Permit HTTPS through firewall  
  ansible.posix.firewalld:  
    service: https  
    permanent: true  
    state: enabled
```


Generate next or modify current suggestion

5. Modified suggestion analytics used to train the model.


Ansible Lightspeed context

Using Playbook keywords with Ansible Lightspeed suggestions

```
---  
- name: Configure firewall  
  hosts: appservers  
  become: true  
  
  vars:  
    firewall:  
      - permanent: true  
        port: ['9090/tcp', '8065/tcp']  
        state: enabled
```



```
- name: Deploy RHEL monitoring  
  hosts: monitoring  
  become: true  
  
  | | # # module_defaults section below.  
  
  module_defaults:  
    ansible.builtin.copy:  
      mode: "644"
```

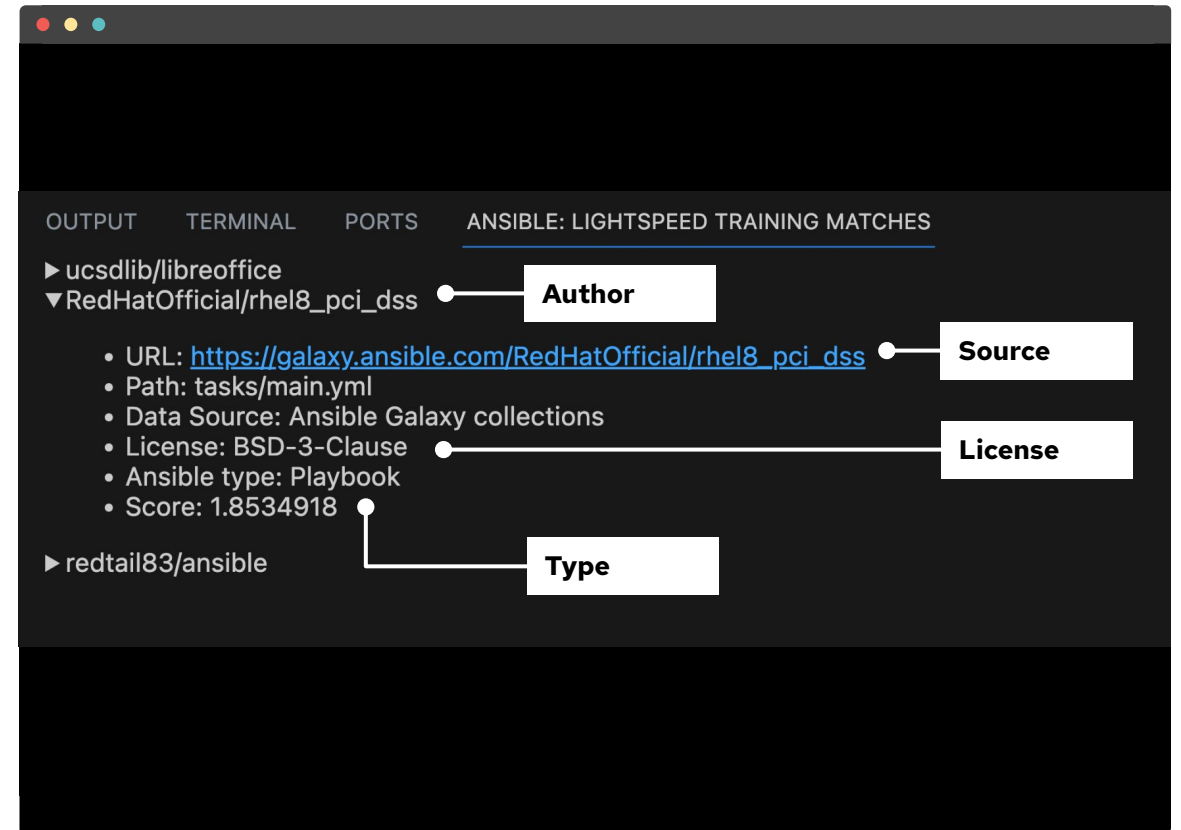


Ansible Lightspeed Content source matching

Transparency, collaboration, and trust

Content source matching

- ▶ Displayed with **Ansible Lightspeed suggestion**.
- ▶ Information provided on **potential source, its author and Open Source license**.
- ▶ **Contributors will have a choice** as to whether or not their work is used to fine-tune the model.



Automation controller

Run automation controller job templates

The screenshot displays the Ansible Automation Controller web interface. On the left, a dark sidebar contains a 'Resources' section with a dropdown menu. The 'Templates' option is selected and highlighted. The main content area, titled 'Templates', shows a table of job templates. The table has columns for Name, Type, Organization, Last Ran, and Actions. Four templates are listed: 'Configure firewall', 'Deploy ACME Corp app', 'Deploy monitoring', and 'Harden app servers'. All are of type 'Job Template' and belong to 'ACME Corp'. The 'Deploy ACME Corp app' and 'Deploy monitoring' templates show a 'Last Ran' timestamp of '14/05/2023, 21:04:44' and '14/05/2023, 21:04:31' respectively. The 'Harden app servers' template shows a 'Last Ran' timestamp of '14/05/2023, 21:03:10'. The 'Actions' column for each template contains a rocket icon and a pencil icon. A blue arrow points to the 'Templates' link in the sidebar, and another blue arrow points to the 'Actions' column header in the table.


Name	Type	Organization	Last Ran	Actions
Configure firewall	Job Template	ACME Corp		
Deploy ACME Corp app	Job Template	ACME Corp	14/05/2023, 21:04:44	
Deploy monitoring	Job Template	ACME Corp	14/05/2023, 21:04:31	
Harden app servers	Job Template	ACME Corp	14/05/2023, 21:03:10	

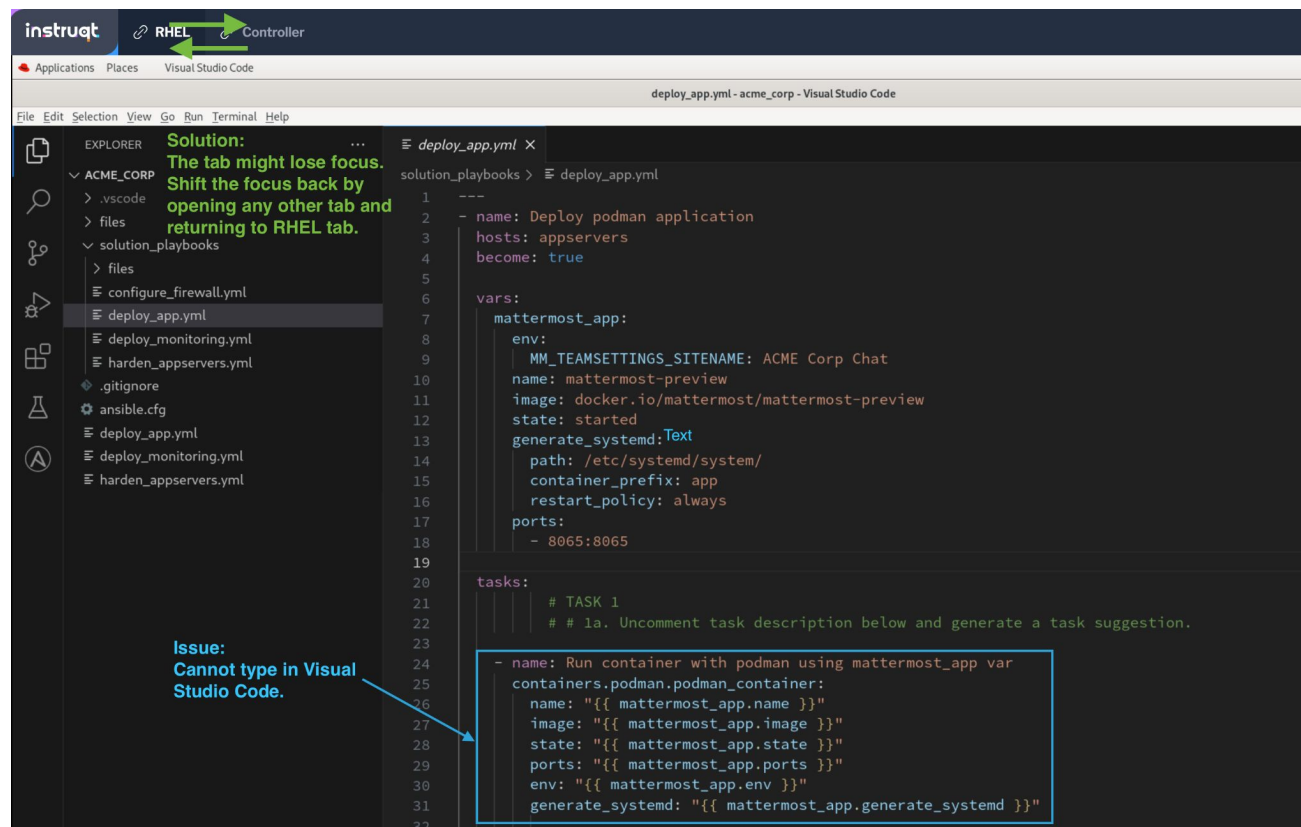
Optional section marker or title

Lab tips and tricks

Lab environment

Notes

- ▶ “ Need help?” section available.
- ▶ **Switch Top tabs** and back again if **you can't type in VS Code** to re-focus on VS Code
- ▶ Credentials: *student* and *learn_ansible*



The screenshot shows the Instruct IDE interface. At the top, there are tabs for 'RHEL' and 'Controller'. Below them, a 'Visual Studio Code' window is open, displaying a file explorer on the left and a code editor on the right. The file explorer shows a project structure for 'ACME_CORP' with files like 'deploy_app.yml', 'deploy_monitoring.yml', and 'hardened_appservers.yml'. The code editor shows an Ansible playbook named 'deploy_app.yml'. A green box highlights a task in the playbook, and a blue box highlights a note about focus.

Solution:
The tab might lose focus. Shift the focus back by opening any other tab and returning to RHEL tab.

Issue:
Cannot type in Visual Studio Code.

```
---
- name: Deploy podman application
  hosts: appservers
  become: true

  vars:
    mattermost_app:
      env:
        MM_TEAMSETTINGS_SITENAME: ACME Corp Chat
      name: mattermost-preview
      image: docker.io/mattermost/mattermost-preview
      state: started
      generate_systemd: Text
      path: /etc/systemd/system/
      container_prefix: app
      restart_policy: always
      ports:
        - 8065:8065

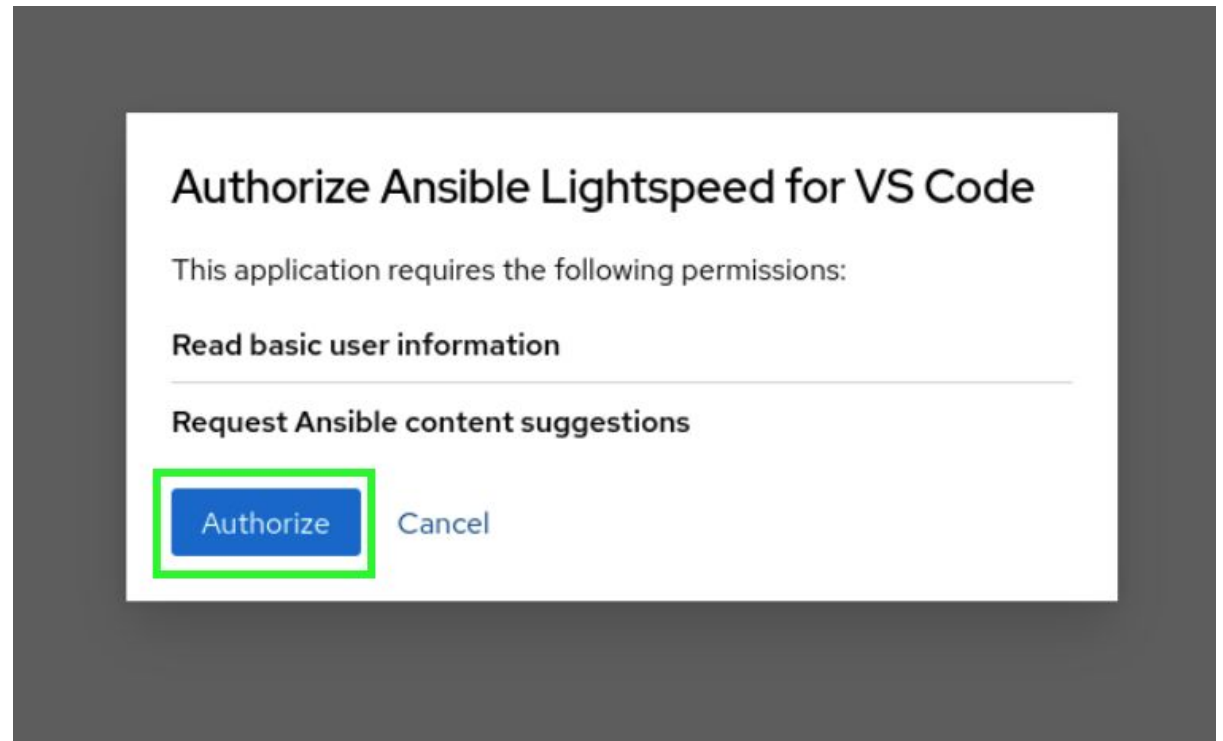
  tasks:
    # TASK 1
    # # 1a. Uncomment task description below and generate a task suggestion.

    - name: Run container with podman using mattermost_app var
      containers.podman.podman_container:
        name: "{{ mattermost_app.name }}"
        image: "{{ mattermost_app.image }}"
        state: "{{ mattermost_app.state }}"
        ports: "{{ mattermost_app.ports }}"
        env: "{{ mattermost_app.env }}"
        generate_systemd: "{{ mattermost_app.generate_systemd }}"
```

GitHub

Notes

- ▶ Environments are **automatically deleted** after the time limit. **No credentials are saved.**
- ▶ **GitHub authorization** process might differ slightly.
- ▶ If login times out. **Click on *Connect* and *Authorize* again.**
- ▶ **No lab data** is pushed to your GitHub account.



Lab Playbook development

Notes

- ▶ **Instructions** are also in Playbooks.
- ▶ Use **CTRL+/ **to uncomment lines.****
- ▶ **Playbook syntax** is important.
- ▶ **It takes a few seconds** to get a suggestion.
- ▶ Solution Playbooks available in ***solution_playbooks*** folder in VS Code.
- ▶ Will need to **reload VS Code** for content source matching.

```
# TASK 2
# # 2a. Uncomment task description below and generate an Ans
# # 2b. Uncomment module_defaults: section above. Save and r
# #      the "mode:" argument.
# # 2c. Remove current Ansible Lightspeed task suggestion be
# # 2d. Note the ansible-lint warning for the task.
# # 2e. Add "# noqa risky-file-permissions" without quotatio
```

Have fun and we're here to help!



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.



facebook.com/redhatinc



twitter.com/ansible



linkedin.com/company/red-hat



youtube.com/AnsibleAutomation