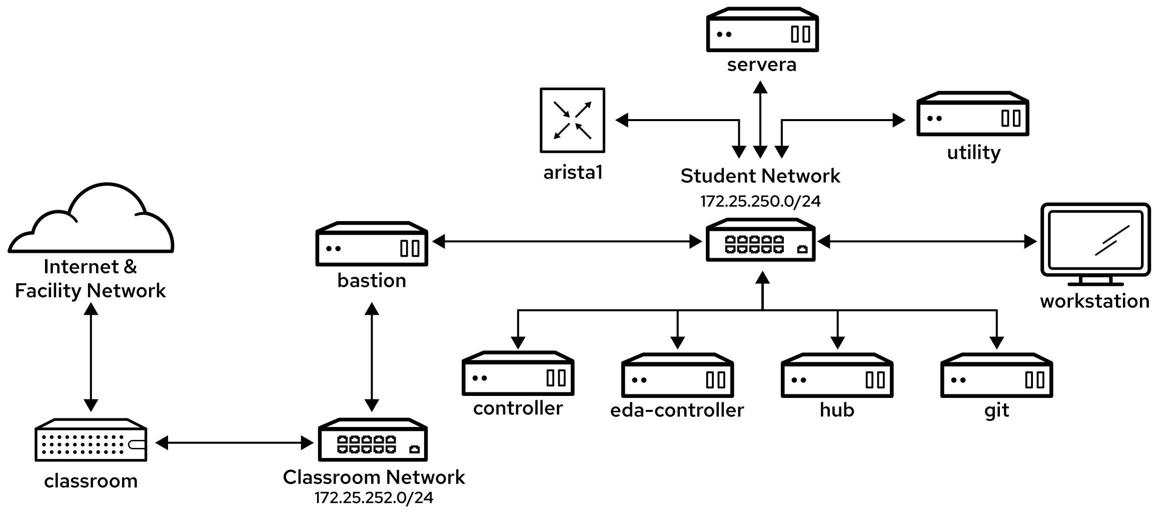
Red Hat
Training and
Certification

DO274

Travis Michette

Table of Contents

Introduction



Repositories for this Course

Main Repository

- **DO274_Notes**: https://github.com/tmichett/DO274_Notes
  - Contains book components and demos and builds on Jenkins server. This is a private repository.
- **DO274_Demo**: https://github.com/tmichett/DO274_Demo
  - Contains PDF copy of book, demo source, and reference materials. This is a public repository and shared as part of the course delivery.

Demo Repositories

- **Controller Playbooks**: https://github.com/tmichett/AAP2_Demos
- **EDA Controller Rulebooks**: https://github.com/tmichett/do274_eda_rulebooks
- **Main Demo Repository**: https://github.com/tmichett/DO274_Demo

URLs and Login Information for the Course

- **EDA Controller**: https://eda-controller.lab.example.com
- **Ansible Automation Controller**: https://controller.lab.example.com
- **Private Automation Hub**: https://hub.lab.example.com
  a. **Username**: admin
  b. **Password**: redhat
- **Mattermost**: http://mattermost.lab.example.com:8065

a. **Username**: student

b. **Password**: Stud3nt123

The Mattermost DNS entry is not in the DNS server. It is only available from the Workstation VM as it is an entry in **/etc/hosts**.

```
172.25.250.220 mattermost.lab.example.com
```

1. Getting Started with Event- Driven Ansible

1.1. Introduction to Event-Driven Ansible

1.1.1. Event-Driven Ansible

1.1.2. Event-Driven Ansible Components

1.1.3. Running Ansible Rulebooks

1.1.3.1. Running Ansible Rulebooks from the Command Line

1.1.3.2. Running Ansible Rulebooks with Event-Driven Ansible Controller

1.1.4. Content for the Event Source Plug-ins

1.1.4.1. Red Hat Ansible Certified Content

1.1.4.2. Ansible Validated Content

1.1.4.3. Getting Content from Private Automation Hub

1.1.5. Event-Driven Ansible Use Cases

1.1.5.1. Fact and Ticket Enrichment

1.1.5.2. High Occurrence of Low-complexity Issues

1.1.5.3. Security and Compliance Automation

1.2. Creating and Testing Ansible Rulebooks

Section Info Here

1.2.1. Reading and Writing Ansible Rulebooks

1.2.2. Selecting Actions for Rules

1.2.2.1. Actions on an Automation Controller

1.2.3. Event Source Plug-ins and Sample Rulebooks

1.2.3.1. Reacting to Webhook Events

1.2.3.2. Reacting to Log Events

1.2.3.3. Reacting to URL Check Events

1.2.4. Testing Ansible Rulebooks

1.3. DEMO - Acting on Webhook Events

Section Info Here

Listing 1. Example Code box for CLI

```
[student@workstation ~]$ sudo yum module install container-tools
```

Listing 2. Example Code box for YAML

```
---
- name: Deploy HTTPD Server Demo
  hosts: server
  collections:

  tasks:

## Start and Run the HTTPD Container
    - name:  Start the Apache Container
      podman_container:
```

Example 1. LAB/Exercise: Hands-On Activity Example

1. Download a container image.
   a. Registry: **registry.access.redhat.com**
   b. Image: **ubi7**
2. Run the container

1.3.1. <Section_Sub_Intro_Here>

1.4. DEMO - Acting on System Journal Events

Section Info Here

Listing 3. Example Code box for CLI

```
[student@workstation ~]$ sudo yum module install container-tools
```

Listing 4. Example Code box for YAML

```
---
- name: Deploy HTTPD Server Demo
  hosts: server
  collections:

  tasks:

## Start and Run the HTTPD Container
    - name:  Start the Apache Container
      podman_container:
```

Example 2. LAB/Exercise: Hands-On Activity Example

1. Download a container image.
   a. Registry: **registry.access.redhat.com**
   b. Image: **ubi7**
2. Run the container

1.4.1. <Section_Sub_Intro_Here>

1.5. DEMO - Acting on Results from the URL Check Plug-in

Section Info Here

Listing 5. Example Code box for CLI

```
[student@workstation ~]$ sudo yum module install container-tools
```

Listing 6. Example Code box for YAML

```
---
- name: Deploy HTTPD Server Demo
  hosts: server
  collections:

  tasks:

## Start and Run the HTTPD Container
    - name:  Start the Apache Container
      podman_container:
```

Example 3. LAB/Exercise: Hands-On Activity Example

1. Download a container image.
   a. Registry: **registry.access.redhat.com**
   b. Image: **ubi7**
2. Run the container

1.5.1. <Section_Sub_Intro_Here>

1.6. DEMO - Watching the Sudoers File

Section Info Here

Example 4. Creating Events Based on Sudoers File Changes

1. Download a container image.
   a. Registry: **registry.access.redhat.com**
   b. Image: **ubi7**
2. Run the container

2. Getting Started with Event- Driven Ansible Controller

2.1. Installing Event-Driven Ansible Controller

Section Info Here

2.1.1. Planning the Installation

2.1.1.1. Automation Controller, Private Automation Hub, and Event- Driven Ansible Controller with External Database Servers

2.1.2. Event-Driven Ansible Controller Installation Options

2.1.2.1. Installation Requirements

2.1.2.2. Database Storage

2.1.3. Subscription and Support

2.1.4. Installing Red Hat Ansible Automation Platform

2.1.4.1. Installing Event-Driven Ansible Controller

2.1.5. Replacing the CA Certificate

2.1.5.1. Gathering Certificates and Private Keys

2.1.5.2. Preparing the Systems

2.1.6. Trusting Custom CA Certificates

2.1.7. Updating RPM Packages on Ansible Automation Platform Servers

2.2. Configuring Event-Driven Ansible Controller to Run Ansible Rulebooks

Section Info Here

2.2.1. Event-Driven Ansible Controller Resources

2.2.2. Creating Credentials

2.2.3. Creating Projects

2.2.4. Creating Controller Tokens

2.2.5. Creating Ansible Rulebook Activations

2.2.6. Launching an Automation Controller Job Template or Workflow Template Using a Rulebook Activation

2.2.7. Viewing Rule Audits

2.3. DEMO - Configuring Event-Driven Ansible Controller to Run Ansible Rulebooks

Section Info Here

Listing 7. Example Code box for CLI

```
[student@workstation ~]$ sudo yum module install container-tools
```

Listing 8. Example Code box for YAML

```
---
- name: Deploy HTTPD Server Demo
  hosts: server
  collections:

  tasks:

## Start and Run the HTTPD Container
    - name:  Start the Apache Container
      podman_container:
```

Example 5. LAB/Exercise: Hands-On Activity Example

1. Download a container image.
   a. Registry: **registry.access.redhat.com**
   b. Image: **ubi7**
2. Run the container

2.3.1. <Section_Sub_Intro_Here>

3. Example Use Cases for Event- Driven Ansible

3.1. GitOps with Event-Driven Ansible

Section Info Here

3.1.1. Using Webhooks in Event-Driven Ansible

3.1.2. Configuring Webhooks in the Git Repository Server

3.1.2.1. Configuring GitLab to use Webhooks

3.1.2.2. Configuring Projects in GitLab to use Webhooks

3.1.2.3. Testing a Webhook in GitLab

3.1.3. Using Tests from GitLab to Create Rules in Rulebooks

3.2. DEMO - GitOps with Event-Driven Ansible

Section Info Here

Listing 9. Example Code box for CLI

```
[student@workstation ~]$ sudo yum module install container-tools
```

Listing 10. Example Code box for YAML

```
---
- name: Deploy HTTPD Server Demo
  hosts: server
  collections:

  tasks:

## Start and Run the HTTPD Container
    - name:  Start the Apache Container
      podman_container:
```

Example 6. LAB/Exercise: Hands-On Activity Example

1. Download a container image.
   a. Registry: **registry.access.redhat.com**
   b. Image: **ubi7**
2. Run the container

3.2.1. <Section_Sub_Intro_Here>

3.3. Event-Driven Ansible and NetOps

Section Info Here

3.3.1. Reacting to Network Events

3.3.2. Managing Network Devices

3.3.3. Running Playbooks that Include Networking Modules

3.3.3.1. Run Playbooks on Your Local System

3.3.3.2. Run Playbooks on an Automation Controller

3.3.4. Using Network Telemetry

3.3.4.1. Configuring Network Telemetry

3.3.4.2. Configuring gNMI on Network Switches

3.3.4.3. Configuring Telegraf

3.3.4.4. Managing and Querying Apache Kafka Topics

3.3.4.5. Using EDA to Query Apache Kafka

3.3.5. Integrating EDA with Chat Services

3.3.5.1. Configuring an Incoming Webhook

3.3.5.2. Configuring an Outgoing Webhook

3.4. DEMO - Event-Driven Ansible using Rulebooks, EDA Controller, and Mattermost

The demo for URL Check Mattermost will launch a rulebook locally on workstation which is similar to Chapter 1. However, this rulebook will notify Mattermost with an error message for the website being down because the service is stopped or because the index.html file is missing.

The demo also utilizes EDA controller as a listener that will listen for events from MatterMost and kick off another set of tasks which integrate with Ansible Automation Controller to run a playbook based on the given events. The playbook will then trigger another notification to Mattermost that the event has been resolved.

Webhook and Mattermost Trigger Phrases

- **website-index-fix**: Used to trigger EDA controller which calls a job on Automation Controller to repair the index.html file and notifiy Mattermost that job is fixed.
- **website-service-fix**: Used to trigger EDA controller which calls a job on Automation Controller to repair the services (systemctl restart httpd) and notifiy Mattermost that job is fixed.

You must first run the **Setup_Demo.sh** script to ensure all pieces are available. Then you can run the rulebook command. To demonstrate both conditions and fixes, you will need to have a separate window open and delete the **index.html** file and **systemctl stop httpd**.

It is also necessary to configure the WebHooks for Mattermost and ensure they are correct in the playbooks.

Incoming Webhooks

Incoming webhooks are needed so that Mattermost can receive events and notifications from EDA Controller and rulebooks. This webhook needs to be updated in the playbooks to ensure that they are triggered correctly.

Incoming Webhook Playbooks

- **website_notify_down.yml**
- **website_notify_missing_index.yml**

System Console ⇒ **Environment** ⇒ **Developer** ⇒ **Allow Untrusted connections** and enter in EDA controller and the 172.25.250.0/24 subnet.

[README 24 05 15 14 42 39] | *images/README-24-05-15-14-42-39.png*

Figure 1. Allowing Webhooks (System Console - Untrusted Connections)

You also need to setup the WebHook **Integrations** ⇒ **Incoming Webhooks**

[README 24 05 15 14 46 59] | *images/README-24-05-15-14-46-59.png*

Figure 2. Allowing Incoming Webhooks (Integrations)

[README 24 05 15 14 50 20] | *images/README-24-05-15-14-50-20.png*

Figure 3. Webhook URL

The **incoming webhook** gets created by the start and setup script, just copy the webhook and place in the playbooks above.

Outgoing Webhooks

The outgoing webhooks must be configured so that Mattermost can connect and contact the EDA Controller based on the service needed.

Outgoing Webhook Playbooks

- **website_notify_down.yml**
- **website_notify_missing_index.yml**

[README 2024 05 16T16 23 06 811Z] | *images/README-2024-05-16T16-23-06-811Z.png*

Figure 4. Outgoing Webhook URL

It is very important that if you are using Rulebooks with the **run_playbook** command that the inventory/host specified in the ruleset is the same host as specified in the playbooks, otherwise, there are issues with **no hosts matched**.

```
PLAY [Send a message to the chat group - Index Missing]
************************
skipping: no hosts matched
```

3.4.1. Demo Instructions - Mattermost URL Check and Fix Demo

3.4.1.1. Setup for the Demo

1. Run the setup script to prepare the environment.

```
[student@workstation ~]$  cd
/home/student/Github/DO274_Demo/Demos/CH3/URL_Check_MM_Demo
```

```
[student@workstation URL_Check_MM_Demo]$ ./Setup_Demo.sh

**********************************************
***** Copying Collections ********************

... OUTPUT OMITTED ...

PLAY RECAP ***************************************************************************
servera                    : ok=6    changed=4    unreachable=0    failed=0
skipped=0    rescued=0    ignored=0

SUCCESS Checking lab systems
SUCCESS Installing required software
SUCCESS Performing common exercise tasks
SUCCESS Creating the exercise directory
SUCCESS Preparing the exercise
```

The **Setup_Demo.sh** script prepares the lab environment and also calls the ***lab start example-chat*** script to setup and pre-configure MatterMost and some of the server configuration settings there.

The **Setup_Demo.sh** script takes about five (5) minutes to run to completion.

2. Disable the **Switch Maintenance Rulebook** in EDA Controller
   a. Login to EDA Controller
   b. Click **Rulebook Activations** then click the switch to **Disable** the activation

      [README 24 05 16 13 07 51]

   c. Confirm **Disable rulebook activations**

      [README 24 05 16 13 11 28]

3. Login to Mattermost (http://mattermost.lab.example.com:8065/)
   a. **UN/PW**: student/Stud3nt123
4. Go to the **System Console** ⇒ **Environment** ⇒ **Developer** ⇒ **Allow Untrusted connections** menu and add the EDA Controller and the classroom subnet.
   - eda-controller.lab.example.com
   - 172.25.250.0/24
     a. Click **Save**
     b. Click **Back to labexamplecom**
5. Select **Integrations** from the System Menu

6. Select Incoming Webhooks
   - http://mattermost.lab.example.com:8065/hooks/a8my9aia1fbs9kw97bp87y1g5r
     a. Copy the webhook for the playbooks

> This webhook will hopefully be the same and no change would need to be made to the playbooks. If that is the case, we are just verifying. However, if a change needs to be made to the webhook, it should be copied and applied to the playbook.
>
> The playbooks that would need to be modified are located in the DO274_Demo project and would not need to be updated in Github.
>
> - **/home/student/Github/DO274_Demo/Demos/CH3/URL_Check_MM_Demo/website_notify_index_missing.yml**
> - **/home/student/Github/DO274_Demo/Demos/CH3/URL_Check_MM_Demo/website_notify_down.yml**

7. Return to the **Integrations** from the System Menu and select **Outgoing Webhooks**
8. Click **Add Outgoing Webhook**
   - **TITLE**: Controller Notification Hook
   - **Description**: Notifications from Ansible Automation Controller
   - **Content Type**: application/json
   - **Channel**: Town Square
   - **Trigger Words**
     - website-index-fix
     - website-service-fix
   - **Callback URL**: http://eda-controller.lab.example.com:5000

     [README 24 05 16 14 30 52] | *images/README-24-05-16-14-30-52.png*

     a. Click **Save**

     b. Click **Done**

3.4.1.2. Running the Demo

1. Create the **EDA Controller** Rulebook Activation
   a. Navigate to **Rulebook Activations**
   b. Click **Create Rulebook Activation**
      - https://eda-controller.lab.example.com/overview

c. Complete the Web Form
   - **Name**: DEMO EDA MatterMost
   - **Project**: Weebsite Maintenance Demo
   - **Rulebook**: web_mattermost_demo.yml
   - **Decision Environment**: Automation Hub Default Decision Environment
   - **Controller Token**: controller.lab.example.com

2. Launch the **url_check.yml** Rulebook

   Listing 11. Running the Rulebook on Workstation

```
[student@workstation URL_Check_MM_Demo]$ ansible-rulebook -r url_check.yml -i
inventory
```

3. In another window or frame, SSH to ServerA as root so that you can stop the services or delete the **index.html** file.

```
[student@workstation ~]$ ssh root@servera
```

4. Delete the **/var/www/index.html** file and observe the Rulebook and Mattermost notifications.

```
[root@servera ~]# rm /var/www/html/index.html
rm: remove regular file '/var/www/html/index.html'? y
```

[README 24 05 16 14 37 28] | *images/README-24-05-16-14-37-28.png*

Figure 5. Mattermost Notification

[README 24 05 16 14 38 50] | *images/README-24-05-16-14-38-50.png*

Figure 6. Rulebook Notification - Workstation

5. Type **website-index-fix** in the MatterMost **Town Square** channel.

[README 24 05 16 14 40 57] | *images/README-24-05-16-14-40-57.png*

Figure 7. Mattermost Notification - Trigger Fix

6. Verify that the website is up and running via the MatterMost alert and the running rulebook

[README 24 05 16 14 46 59] | *images/README-24-05-16-14-46-59.png*

Figure 8. Mattermost Notification - Website Fixed

[README 24 05 16 14 48 00] | *images/README-24-05-16-14-48-00.png*

Figure 9. Rulebook Notification - Workstation - Website Fixed

The process can be repeated for the **httpd** service. In order to test the service triggers, run **systemctl stop httpd** on serverA and to fix the service, type **website-service-fix** in Mattermost.

It is also important that again, the URLs might need changed for the incoming webhooks. If this is the case, please submit an issue as I can turn this into a variable for Ansible Automation Controller as there is no direct "commit" access for my main branch. Another solution would be to "fork" the repository and make it your own.

Playbooks that need to be modified for the Ansible Automation Controller are in the **DO274 Demo Project** which belong to the https://github.com/tmichett/AAP2_Demos.git repository.

Playbooks to modify the URL for MatterMost Incoming Webhook

- website_notify_index_restored.yml
- website_notify_service_restored.yml

The Webhook being used should be part of the course, but it may change with course updates.

3.4.1.3. Repositories and Items in Demo

There are several public repositories used in this demo. The reason these needed to be used is based on the project structure, file locations, and because they are also shared resources in other courses.

- **Controller Playbooks**: https://github.com/tmichett/AAP2_Demos
- **EDA Controller Rulebooks**: https://github.com/tmichett/do274_eda_rulebooks
- **Main Demo Repository**: https://github.com/tmichett/DO274_Demo

include::Chapters/CH3/Section2Demo1.adoc[]

include::Chapters/CH3/Section2Demo2.adoc[]

include::Chapters/CH3/Section2Demo3.adoc[]