Red Hat Ansible Automation Platform 2.4

Getting Started with Event-Driven Ansible Guide

Learn about the benefits and how to get started using Event-Driven Ansible.
Learn about the benefits and how to get started using Event-Driven Ansible.
Abstract

Event-Driven Ansible is a new way to enhance and expand automation by improving IT speed and agility while enabling consistency and resilience. This feature is designed for simplicity and flexibility.
# Table of Contents

PREFACE ................................................................. 3

MAKING OPEN SOURCE MORE INCLUSIVE .............................................. 4

PROVIDING FEEDBACK ON RED HAT DOCUMENTATION .................................. 5

CHAPTER 1. EVENT-DRIVEN ANSIBLE AUTOMATION ........................................ 6
  1.1. EVENT-DRIVEN ANSIBLE BENEFITS .............................................. 6

CHAPTER 2. ANSIBLE RULEBOOKS ......................................................... 7
  2.1. DECISION ENVIRONMENTS ....................................................... 7
  2.2. RULEBOOK ACTIONS ........................................................... 7

CHAPTER 3. INSTALLATION OF EVENT-DRIVEN ANSIBLE CONTROLLER ................. 9
  3.1. INSTALLING EVENT-DRIVEN ANSIBLE CONTROLLER ON RED HAT ANSIBLE AUTOMATION PLATFORM ......................................................... 9

  3.2. DEPLOYING EVENT-DRIVEN ANSIBLE CONTROLLER WITH ANSIBLE AUTOMATION PLATFORM OPERATOR ON OPENSSHIFT CONTAINER PLATFORM ......................................................... 9

CHAPTER 4. USING EVENT-DRIVEN ANSIBLE CONTROLLER ................................ 11
  4.1. EVENT-DRIVEN ANSIBLE CONTROLLER TASKS .................................... 11
PREFACE

Thank you for your interest in Event-Driven Ansible. Event-Driven Ansible is a new way to enhance and expand automation. It helps teams automate decision-making and improve IT speed and agility.

This guide provides the conceptual framework of Event-Driven Ansible and links you to information on installing and using Event-Driven Ansible controller.
MAKING OPEN SOURCE MORE INCLUSIVE

Red Hat is committed to replacing problematic language in our code, documentation, and web properties. We are beginning with these four terms: master, slave, blacklist, and whitelist. Because of the enormity of this endeavor, these changes will be implemented gradually over several upcoming releases. For more details, see our CTO Chris Wright’s message.
PROVIDING FEEDBACK ON RED HAT DOCUMENTATION

If you have a suggestion to improve this documentation, or find an error, please contact technical support at https://access.redhat.com to create an issue on the Ansible Automation Platform Jira project using the docs-product component.
CHAPTER 1. EVENT-DRIVEN ANSIBLE AUTOMATION

Event-Driven Ansible is a new way to connect to sources of events and act on those events using rulebooks. This technology improves IT speed and agility, and enables consistency and resilience.

1.1. EVENT-DRIVEN ANSIBLE BENEFITS

Event-Driven Ansible is designed for simplicity and flexibility. With these enhancements, you can:

- Automate decision making
- Use numerous event sources
- Implement event-driven automation within and across multiple IT use cases
- Achieve new milestones in efficiency, service delivery excellence and cost savings

Event-Driven Ansible minimizes human error and automates processes to increase efficiency in troubleshooting and information gathering.

This guide helps you get started with Event-Driven Ansible by providing links to information on understanding, installing, and using Event-Driven Ansible controller.
CHAPTER 2. ANSIBLE RULEBOOKS

Event-Driven Ansible controller provides the interface in which Event-Driven Ansible automation performs. Ansible rulebook provides the framework for Event-Driven Ansible automation. Ansible rulebook is essentially a collection of rulesets, which in turn, consists of one or more sources, rules, and conditions.

2.1. DECISION ENVIRONMENTS

Event-Driven Ansible includes, by default, an ansible.eda collection, which contains sample sources, event filters and rulebooks. All the collections, ansible rulebooks and their dependencies use a Decision Environment, which is an image that can be run on either Podman or Kubernetes.

In Decision Environments, sources, which are typically Python code, are distributed through ansible-collections. They inject external events into a rulebook for processing. The rulebook consists of the following:

- The python interpreter
- Java Runtime Environment for Drools rule engine
- ansible-rulebook python package
- ansible.eda collection

You can use the base Decision Environment and build your own customized Decision Environments with additional collections and collection dependencies. You can build a Decision Environment using a Dockerfile or optionally you can deploy your CA certificate into the image.

2.2. RULEBOOK ACTIONS

A rulebook specifies actions to be performed when a rule is triggered. A rule gets triggered when the events match the conditions for the rules. The following actions are currently supported:

- run_job_template
- run_playbook (only supported with ansible-rulebook CLI)
- debug
- print_event
- set_fact
- post_event
- retract_fact
- shutdown

Additional resources

- For more information on using rulebooks, see the Event-Driven Ansible controller User’s Guide.
For more detailed information on Ansible rulebooks, see the section on Rulebooks in the Ansible Rulebook documentation.
CHAPTER 3. INSTALLATION OF EVENT-DRIVEN ANSIBLE CONTROLLER

Similar to the automation controller and automation hub components, the setup for Event-Driven Ansible controller includes default settings for specific variables in the inventory files.

IMPORTANT

Event-Driven Ansible controller must be installed on a separate server and cannot be installed on the same host as automation hub and automation controller.

3.1. INSTALLING EVENT-DRIVEN ANSIBLE CONTROLLER ON RED HAT ANSIBLE AUTOMATION PLATFORM

To prepare for installation of Event-Driven Ansible controller, review the following information in the Red Hat Ansible Automation Platform Planning Guide:

- Example Ansible Automation Platform architecture, including the Event-Driven Ansible controller
- Event-Driven Ansible controller general information
- Event-Driven Ansible controller system requirements

When you are ready to install the Event-Driven Ansible controller, refer to the procedures in the Red Hat Ansible Automation Platform Installation Guide beginning with Chapter 3. Installing Red Hat Ansible Automation Platform. For a specific code example related to Event-Driven Ansible controller, refer to the Single Event-Driven Ansible controller node with internal database scenario in this chapter.

To further assist you in getting started with Event-Driven Ansible, review the following existing installation scenarios that have been updated with code examples and variables that are required to install Event-Driven Ansible controller:

- Ansible Automation Platform with an external (installer managed) database.
- Ansible Automation Platform with an external (customer provided) database.


3.2. DEPLOYING EVENT-DRIVEN ANSIBLE CONTROLLER WITH ANSIBLE AUTOMATION PLATFORM OPERATOR ON OPENShift CONTAINER PLATFORM

Event-Driven Ansible is not limited to Ansible Automation Platform on VMs. You can also access this feature on Ansible Automation Platform Operator on OpenShift Container Platform. To deploy Event-Driven Ansible with Ansible Automation Platform Operator, follow the instructions in Deploying Event-Driven Ansible controller with Ansible Automation Platform Operator on OpenShift Container Platform.

After successful deployment, you can connect to event sources and resolve issues more efficiently.

Additional resources
- For more detailed information on planning, see the *Red Hat Ansible Automation Platform Planning Guide*.

- For a comprehensive list of predefined variables for the Event-Driven Ansible controller, see the *Red Hat Ansible Automation Platform Installation Guide*.

- For more information about deployment on Ansible Automation Platform operator, see the *Deploying the Red Hat Ansible Automation Platform operator on OpenShift Container Platform Guide*.
CHAPTER 4. USING EVENT-DRIVEN ANSIBLE CONTROLLER

After you have successfully installed the Event-Driven Ansible controller, you can access the interface to manage your IT responses across all event sources. Since Event-Driven Ansible controller is integrated with automation controller, you can automate a combination of processes, including issue remediation, user administration tasks, operational logic, and the like.

4.1. EVENT-DRIVEN ANSIBLE CONTROLLER TASKS

Depending on your role, you can use Event-Driven Ansible controller for any of the following tasks:

- Configuring a new project
- Setting up a new decision environment
- Creating a new authentication token
- Setting up a rulebook activation

Next steps

- For further details on using Event-Driven Ansible controller, see the Event-Driven Ansible controller User’s Guide.