TECHNICAL WHITE PAPER: **August 2024**

VMware Cloud Foundation Automation ABX Actions for Ansible Automation Platform

User Guide

vmware by Broadcom

Table of contents

Revision History

Date	Version	Description	Modified By
08/22/2025	1	Initial release	• Charles Lee • Dharmesh Bhatt

About the VMware Cloud Foundation Automation ABX Actions for Ansible Automation Platform User Guide

The *VMware Cloud Foundation (VCF) Automation ABX Actions for Ansible Automation Platform Deployment guide* provides instructions on using our sample Aria custom resource to interface with Red Hat Ansible Automation Platform.

Intended Audience

The VMware Cloud Foundation User Guide is intended for data center cloud administrators and developers creating automation using VMware Aria Automation in their organization. The information in this guide is written for experienced data center cloud administrators and automation developers who are familiar with:

- VMware Aria Automation: How to use and develop in Aria Automation Assembler.
- Red Hat Ansible Automation Platform: Creating and managing users, organizations, and interfacing with revision control platforms, such as GitLab and GitHub. Knowledge of Ansible inventories and its components. How to develop and use playbooks to provide configuration management.
- Python 3: Knowledge of Python 3 and Python packages.
- Yaml: Knowledge of yaml syntax and data types.

Related Publications

- VMware Aria Ansible Automation Platform API - Deployment Guide
- Getting Started with VMware Aria Automation
- Using Automation Assembler
- Red Hat Ansible Automation Platform operations guide

Preparing the Environment

This solution expects the following components already installed and configured:

VMware Cloud Foundation Automation 8

Red Hat Ansible Automation Platform 2.4

Aria Automation Ansible Automation Platform API 0.4.1

VMware Aria Automation

VMware Aria Automation™ is a cloud infrastructure automation solution powering VMware Cloud Foundation. It delivers a self-service private cloud with governance and resource lifecycle management across on-premises data centers or on any supported public cloud. It leverages a service-driven cloud computing interface, a policy controlled self-service catalog, Infrastructure as Code (IaC), and infrastructure pipelining. It enables Cloud Ops teams to maintain frictionless governance and control while empowering developers with a high level of agility and flexibility.

Red Hat Ansible Automation Platform

Red Hat® Ansible® Automation Platform is a unified solution for strategic automation. It combines the security, features, integrations, and flexibility needed to scale automation across domains, orchestrate essential workflows, and optimize IT operations to successfully adopt enterprise AI.

Aria Automation Ansible Automation Platform API

Open-source project implementing an interface to Red Hat Ansible Automation Platform. The custom resource created by this project allows Aria developers to pass infrastructure created in Aria Assembler to Red Hat Ansible Automation Platform as inventories and running Ansible Automation Projects against those inventories.

Solution Configuration

This section introduces the resources and configurations:

- Architecture diagram
- Software resources

Software Resources

The following table lists the software resources used in this solution.

Table 1 - Software Resources

Software	Version	Purpose
VMware Cloud Foundation Automation	8.16+	VMware Automation Infrastructure as a Service (IaaS) Platform
Red Hat Ansible Automation Platform	2.4+	Red Hat Ansible Configuration Management (CM) server
AAP API	0.4.1	Open-source ABX actions: *https://github.com/vmware-workloads/aap-api*

Usage

Assembler Templates

This section outlines how to use the Aria Ansible Automation Platform API.

1. Open a VMware Aria Automation Assembler Template.
2. Select the **Ansible Automation Platform** custom resource and drag it to the canvas.
3. Once added to the canvas, edit the code section and add the variables.

Mandatory Fields

The following fields are mandatory and must be completed. For more details about these variables, consult the following Variables section.

- job_template_name
- inventory_name
- hosts

Optional Fields

The following fields are optional and can be used as needed to create the required inventory file. For more details about these variables, consult the following Variables section.

- inventory_variables
- host_variables
- host_groups
- group_variables

Code Sample

This section provides an overview of the custom resource, and a code sample illustrating how the solution is invoked the same as any other native resource Assembler resource.

In the graphical component, the resource is created by dragging the resource from the left menu bar onto the canvas. Dependencies between the Ansible API custom resource and other resources can be created by adding the appropriate connections between resources on the canvas.

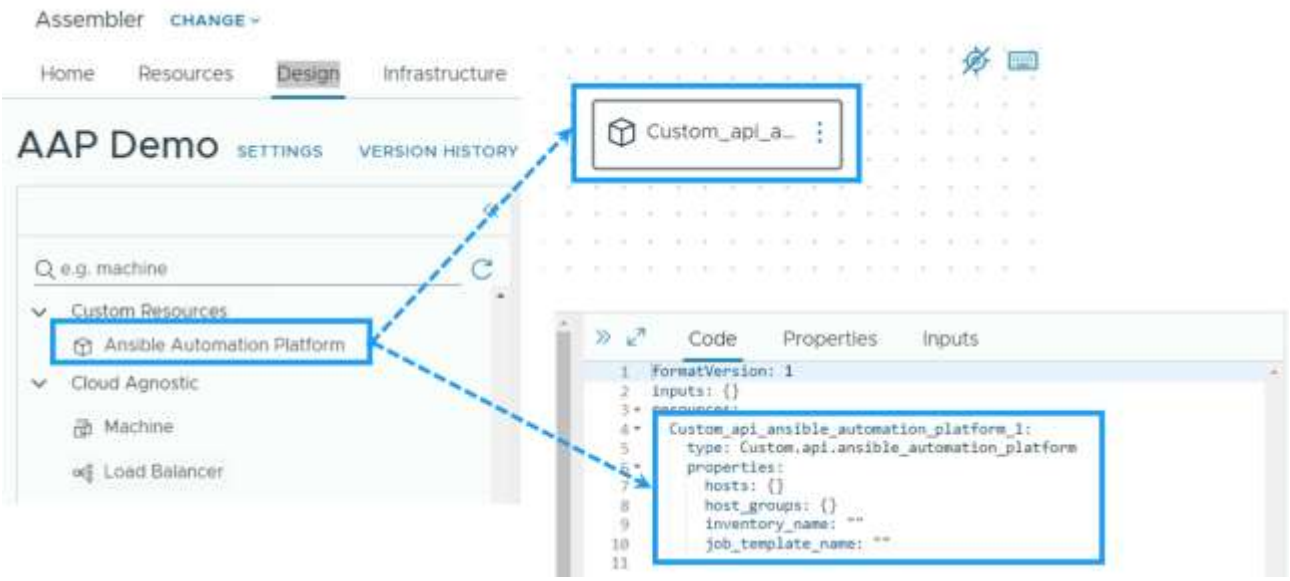

Figure 1 VMware Cloud Foundation Automation with Ansible Automation Platform Custom Resource

In the template YAML code block, Assembler field completion provides the list of available keys and values. Using the auto-completion feature users can rapidly create the code to connect their VMware Cloud Foundation infrastructure to their Ansible playbooks.

```
#
# Code sample with all the options configured with example values.
#
resources:
  Custom_api_ansible_automation_platform_1:
    type: Custom.api.ansible_automation_platform
    properties:
      organization_name: Default
      job_template_name: Ansible Template
      inventory_name: ${env.deploymentId}
      inventory_variables:
        use_ssl: true
        lb_address:
          - ${resource.web_lb.address}
      hosts:
        - ${resource.vm-1.*}
        - ${resource.vm-2.*}
        - ${resource.vm-3.*}
      group_variables:
        group1:
          sql_port: 26257
          rpc_port: 26357
        group2:
           service_name: myservice
      host_variables:
        crdb-vm:
          rack: 1
          verbose: true
      host_groups:
        group1:
          - ${resource.vm-1.*}
        group2:
          - ${resource.vm-2.*}
        group3:
          - ${resource.vm-1.*}
          - ${resource.vm-2.*}
#
# End
#
```

Variables

This section provides a list and description of the various Ansible Automation Platform API variables.

organization_name

Description: This variable maps to the name of the Ansible Automation Platform organization. Each inventory must be associated with an organization on the Ansible Automation Platform server (**Figure 1**). Ansible Automation Platform comes with a default organization named *Default*. If this variable is not set, the inventory is created in the *Default* organization.

Type: String

Required: No

Default Value: "Default"

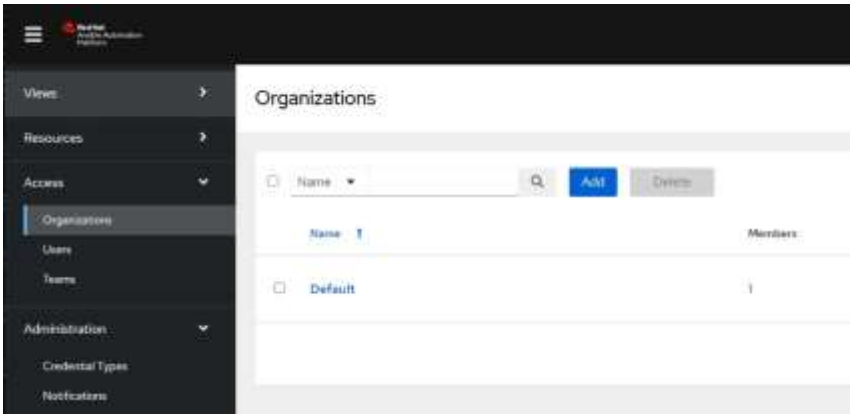

Figure 2 Ansible Automation Platform Organizations

job_template

Description: A string containing the name of the Ansible job template that will be run against the inventory. This value must be specified and must exist in the Ansible Automation Platform Templates (**Figure 3**).

Type: String

Required: Yes

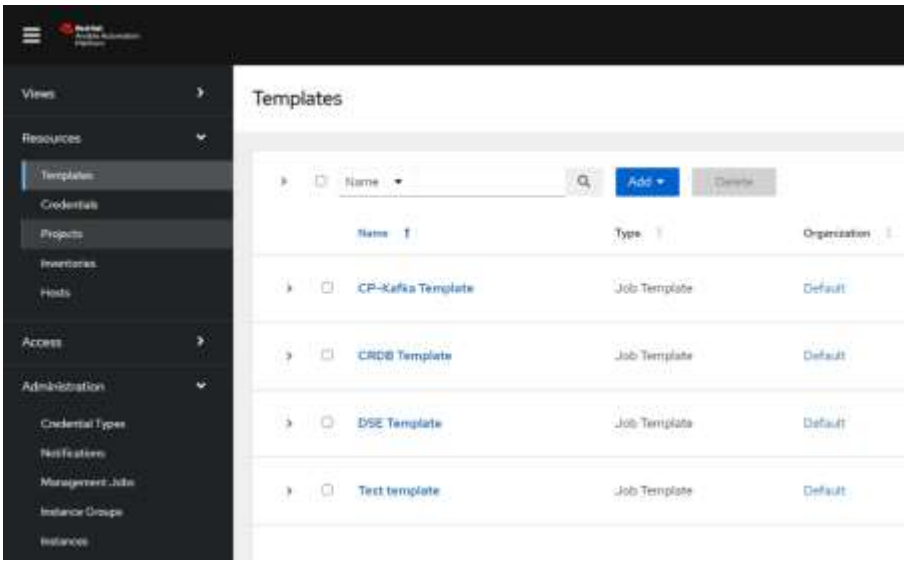

Figure 3 Ansible Automation Platform Templates

inventory_name

Description: Specifies the name of the inventory on the Ansible Automation Platform server. This value must be unique in the Ansible Automation Platform inventories (**Figure 4**). To ensure the name is unique, it is recommended to use VMware Cloud Foundation template variables, such as the Assembler variable *deploymentId.*

Type: String

Required: Yes

Constraints: Name must be unique on the Ansible Automation Platform server inventories.

```
#
# Code samples to generate unique inventory names
#
# This will create an inventory with the Aria Deployment UUID
inventory_name: ${env.deploymentId}

# This deployment ID can be prefixed or suffixed with other variables or strings
inventory_name: kafka-${env.deploymentId}
inventory_name: ${env.orgId}-${env.deploymentId}
inventory_name: ${env.deploymentId}-${env.requestedBy}
#
# End
#
```

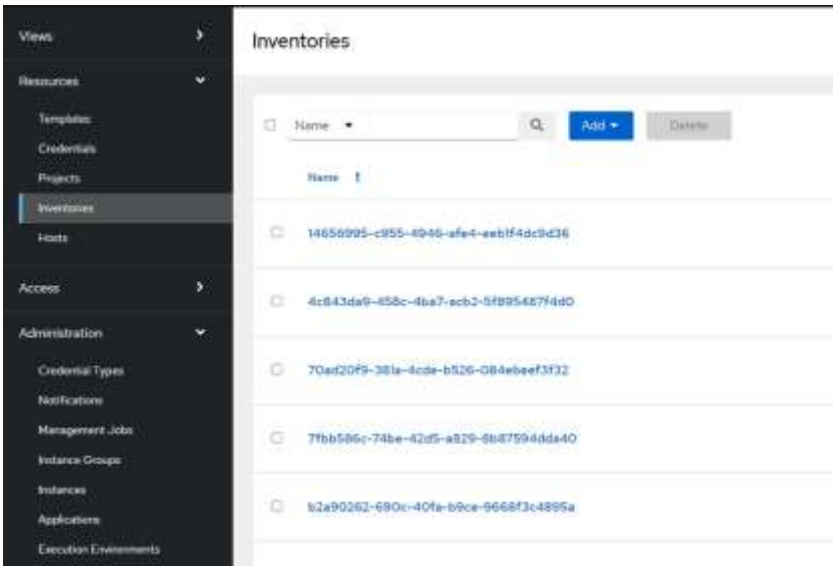

Figure 4 Ansible Automation Platform Inventories

inventory_variables

Description: Accepts a mapping (i.e. hashes / dictionary) with any valid yaml datatypes, including other mappings. The variables defined here are added as inventory variables in the specified Ansible Automation Platform inventory (**Figure 5**).

Type: Mapping

Required: No

```
#
# This is a sample of a yaml mapping for the inventory variables
#
inventory_variables:
  use_ssl: true
  lb_addresses:
    172.16.60.14
#
# End
#
```

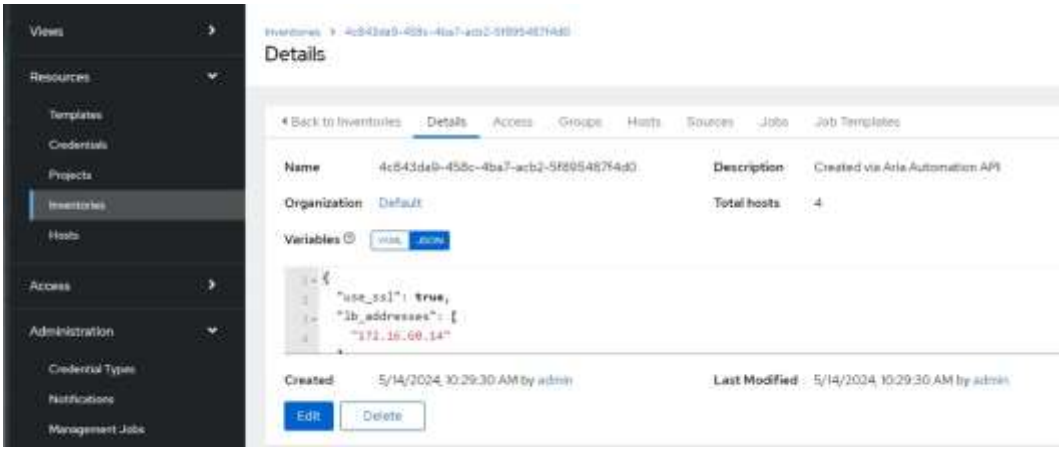

Figure 5 Ansible Automation Platform Inventory Variables

hosts

Description: List of all the inventory hosts that will be added to the Ansible Automation Platform inventory (**Figure 6**). This variable expects a list of Aria Automation resources of type *Cloud.vSphere.Machine*.

Type: List of Cloud.vSphere.Machine

Required: Yes

```
#
# Hosts sample taken from a Kafka Assembler Template
#
Resources:
  control-center:
    type: Cloud.vSphere.Machine
# ...
  zookeeper:
    type: Cloud.vSphere.Machine
# ...
    ...
  kafka-broker:
    type: Cloud.vSphere.Machine
# ...
  Custom_api_ansible_automation_platform_1:
    type: Custom.api.ansible_automation_platform
      hosts:
        - ${resource.control-center.*}
        - ${resource.zookeeper.*}
        - ${resource.kafka-broker.*}
#
# End
#
```

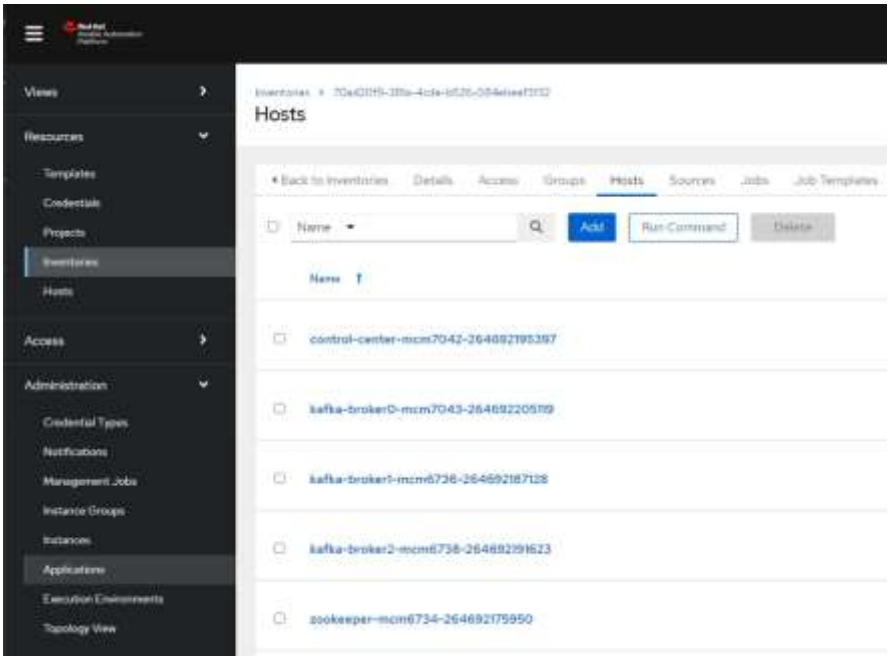

Figure 6 Ansible Automation Platform Inventory Hosts

host_variables

Description: Expects a mapping (i.e. hashes / dictionary) of variables that will be assigned to the host(s). The mapping uses the *Cloud.vSphere.Machine* name to select the host(s) then applies the specified variables to the hosts in the Ansible Automation Platform Inventory (**Figure 7**).

Type: Mapping

Required: No

Note: When creating vSphere Machine resources using the *count* parameter, the resource name will be the same for all the machines. This means the host variables will be applied to all the machines created by the resource. To create differentiated variables, it is necessary to create multiple individual (i.e. count: 1) Cloud.vSphere.Machine resources and apply to each the unique variables.

```
#
# Sample Host Variables
#
resources:
  crdb-vm:       # The resource name
    type: Cloud.vSphere.Machine
    ...
Custom_api_ansible_automation_platform_1:
    type: Custom.api.ansible_automation_platform
    host_variables:
      crdb-vm:   # The resource name is used to locate the variables to be assigned
        rack: 1
        port: 80
        verbose: true
        disks: ${input.diskConfig}
#
# End
#
```

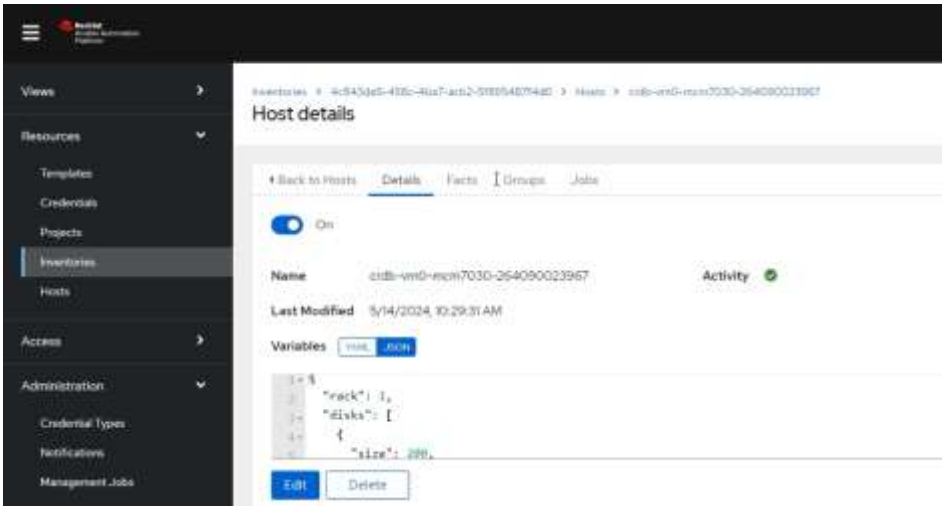

Figure 7 Ansible Automation Platform Inventory Host Variables

host_groups

Description: Expects a mapping of lists of *Cloud.vSphere.Machine* that will be used to define the groups and group members it the Ansible inventory (**Figure 8**)

Type: Mapping

Required: No

```
#
# Sample host groups
#
Resources:
  control-center:
    type: Cloud.vSphere.Machine
# ...
  zookeeper:
    type: Cloud.vSphere.Machine
# ...
  kafka-broker:
    type: Cloud.vSphere.Machine
# ...
  Custom_api_ansible_automation_platform_1:
    type: Custom.api.ansible_automation_platform
      host_groups:
        control_center:
          - ${resource.control-center.*}
        zookeeper:
          - ${resource.zookeeper.*}
        kafka_broker:
          - ${resource.kafka-broker.*}
#
# End
#
```

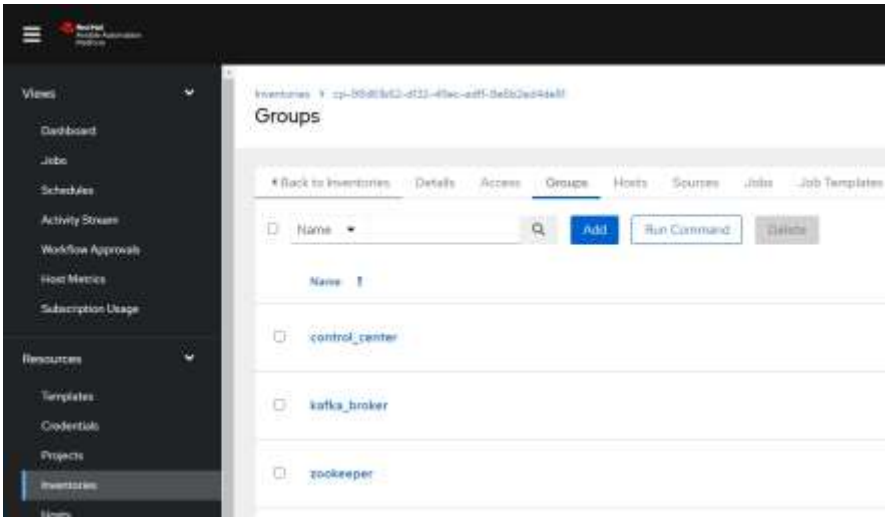

Figure 8 Ansible Automation Platform Inventory Groups

group_variables

Description: Expects a mapping with the name of the Ansible inventory group and the variables that will be applied to the inventory group (**Figure 9**). The group variables can be of any type, including mappings.

Type: Mapping

Required: No

```
#
#
#
Resources:
  crdb-vm:
    type: Cloud.vSphere.Machine
# ...
  Custom_api_ansible_automation_platform_1:
    type: Custom.api.ansible_automation_platform
      host_groups:
        crdb:    # group name
          - ${resource.crdb-vm.*}
      group_variables:
        crdb:    # group name
          psql_port: 26257
          rpc_port: 26357
#
# End
#
```

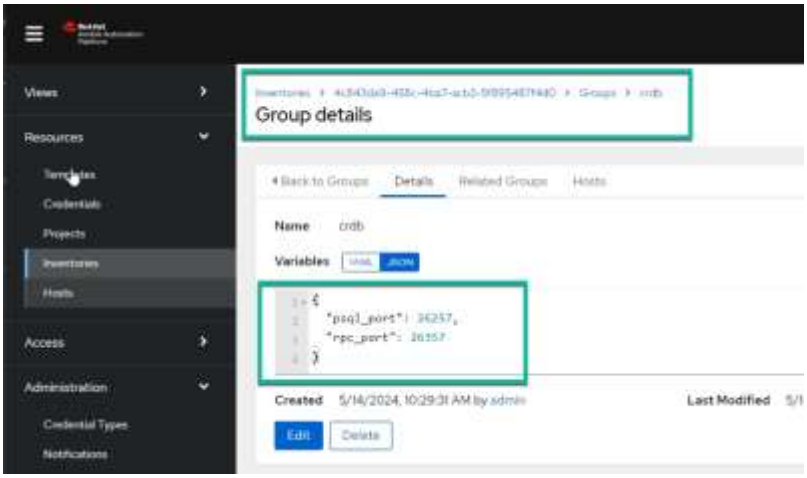

Figure 9 Ansible Automation Platform Inventory Group Variables

References

- *VMware Cloud Foundation*
- *VMware Aria Automation*
- *Red Hat Ansible Automation Platform*
- *VMware Aria Ansible Automation API*

About the Authors

Charles Lee, *Product Marketing Engineer,* in the VCF Technical Marketing team of VMware Cloud Foundation Business Unit of VMware by Broadcom, wrote the original version of this paper.

Dharmesh Bhatt, *Product Marketing Engineer,* in the VCF Technical Marketing team of VMware Cloud Foundation Business Unit of VMware by Broadcom, wrote the original version of this paper

Item No: vmw-bc-wp-tech-temp-uslet-word-2024 1/24