Ansible awx api guide

Note: This API documentation assumes that you are using 3.2.0 and above versions of asible-tower-cli. If you are running a lower version than 3.2.0, there is no guarantee that using the API in this documentation will work as expected. Like tower interface, Tower CLI is a client that talks to several REST services of tower server, but using a Python script or UNIX command line. Thus, the use of the CLI Tower API is pretty straight forward: get a resource that matches its counterpart in the Tower backend, and call the public methods of this resource, which within the term requests a specific REST endpoint and receive/reproduce the JSON response. Here's a simple example of creating a new organization using Tower CLI on Python: with tower_cli importing get_resource from import tower_cli.exceptions Found with tower_cli.conf import options from settings.runtime_values (username = 'user', password = 'pass'): try: res = get_resource ('organization') new_org = res.create(name = 'foo', description='bar', fail_on_found=True) except found: print('This organization already exists.') claims that isinstance(new_org, dict) print(new_org['id') The above example shows a template for most cases using the Tower CLI API, which consists of 3 parts: execution configuration, resource retrieval, and call of its public methods and exception processing. Tower CLI requires a set of configurations to function properly, all configuration settings are stored in a single tower_cli.conf.settings object that provides the public context manager runtime_values temporary override options in a file with temporary execution values. Learn more about CLI Tower configurations under Configuration. Most of the resources listed at tower/api/v2/ endpoint have client proxy classes in Tower CLI. The two main ways to get resources in the CLI Tower are from tower_cli import get_resource res =<resource name=>get_resource(" ' ') and import tower_cli.resources.<resource module= name=>. Resource as <alias>res = <alias>() The default resource in the CLI tower has 2 components: fields and public methods. Resource fields can be treated as wrappers around actual resource fields exhibited by the Tower REST API. They are usually used by public methods to create and change resources and filter when searching for specific resources; Public methods are actual wrappers around Tower REST API requests, they can be used for both general CRUD operations against Tower resources, such as removing a user and for specific tasks such as running a command without printing, monitoring task launch or building workflow schedules from a script. The following table of contents lists all available Tower CLI resources, the documentation for each of them follows one structure: section Description, which gives accession to the resource; Field Table section, which lists all available fields resource; and section of the API Specification, which expands the details of the use of each available public method. Note that most public methods have a keyword argument **kwargs. This argument </alias> </alias> </resource> </resource> </resource> contains and contains only resource fields if they are not specified. For any errors in the use or connection exceptions situation as subclasses tower_cli.exceptions.TowerCLIError, see the Exception section below for more information. Redirect environment settings to the Action Statuses page. AnsibleFest Products Community Webinars &Amp; Training Blog Documentation Ansible Tower API Guide Thank you for your interest in Ansible Tower. Ansible Tower is a commercial offering that helps teams manage complex multilevel deployments by adding control, knowledge and delegation in environments running at a possible capacity. Ansible Tower API Guide focuses on helping you understand the API of a possible tower. This document has been updated to include information for the latest release of Ansible Tower v3.7.3. We need feedback! If you notice a typo in this documentation, or if you thought about a way to make this tutorial better, we'd like to hear from you! Please send an email to: docs@ansible.com you have an offer, try to be as specific as possible by describing it. If you find an error, please include the guide name, chapter number/section number and some of the surrounding texts so we can easily find it. We may not be able to reply to every message sent to us, but you can rest assured that we will read them all! Possible version of tower 3.7.3; September 30, 2020; Ansible, Ansible Tower, Red Hat and Red Hat Enterprise Linux are trademarks of Red Hat, Inc., registered in the United States and other countries. If you distribute this document or the modified version of it, you must provide attribution to Red Hat, Inc. and provide a link to the original version. The rights of third parties Ubuntu and Canonical are registered trademarks of Canonical Ltd. The CentOS Project is copyrighted. CentOS signs are trademarks of Red Hat, Inc. (Red Hat). Microsoft, Windows, Windows Azure, and Internet Explore are Microsoft trademarks, Inc. VMware is a registered trademark or trademark of VMware, Inc. Rackspace, service marks, logos and domain names are trademarks/trademarks of common law or registered trademarks/service marks of Rackspace US, Inc. or its subsidiaries, as well as protected trademarks and other laws in the United States of America and other countries. Amazon Web Services, AWS, Amazon EC2 and EC2, are trademarks of Amazon Web Services, Inc. or its affiliates. The OpenStack ™ OpenStack are trademarks of OpenStack LLC. Chrome™ and Google Compute Engine™ registered trademarks of Google Inc. Safari® is a registered trademark of Apple, Inc. Firefox® is a registered trademark of the Mozilla Foundation. All other trademarks are the property of their respective owners. Content © Red Hat, Inc. Red Hat Ansible created using the Sphinx using the theme provided by Read the Docs. Photo Farzad Nazifi on UnsplashThe first thing you need is to understand the AWX REST API to notice their available resources and how with them. Next, you'll use the tower-cli tool to manage AWX and perform tasks such as creating inventory, adding hosts, disabling, and removing hosts from inventory. ConceptsREST means representative state transfer and is sometimes written as ReST. It relies on a stateless communication protocol, client server, and communication caching, usually HTTP. ¹ Let explores and understands the example of interaction with the AWX REST API, this topic is intended to explain no direct action is required. First, you need to know which resources are available through the API. All resources are described in detail here. Expand inventory resource. A practical example is the documentation. Ther is 3 available response status code for this resource:201 - created400 - bad request403 - forbiddencURL example:Following topics practical! Prerequisites Before you can work with the API, you need to install some tools. Install ansible-tower-cli using pip3. Install the perl-JSON-PP and jq packages. CLI First Steps Select-cli is the recommended tool for managing Ansible AWX and Tower, but you can use awx or awx-cli as well, awx-cli will be the successor to tower-cli. The initial configuration of the kley tower. ²A show of the configuration of the tower-cli:Creation of static inventoryC create static inventory. Here's another way to do the same task using curl (REST API): Here's another way to do the same task using the web user interface: left menu (inventory) > click [+] (select Inventory) > fill out the form > click [SAVE]Add hosts to static inventoryAdd hosts to inventory. Here's another way to do the same task using curl (REST API): Here's another way to do the same task using the web user interface: left menu (inventory) > click (Inventory Example) > click [HOSTS] > click [+] > Fill in form > click [SAVE]Inventory Identification by IDBefore to make any changes to static inventory, you will need to check the inventory ID first. Here's another way to do the same task using the curl command (REST API): Identifying the host by IDNow, you'll need to check only the enabled hosts. Here's another way to do the same task using the curl command (REST API) :D unable to host from IDOnce our target host is nodee that has ID 6, you can disable it on inventory. Here's another way to do the same task using curl (REST API): Here's another way to do the same task using the web UI: menu left (inventory) > click (Inventory Example) > click [HOSTS] > deselect keyDelete host IDDelete nodee from inventory. Here's another way to do the same task using the curl command (REST API): NOTE: There are two ways recommended curl command. Here's another way to do the same task using the web user interface: left menu (inventory) > click (Inventory example) > click [HOSTS] > click [recycle bin icon]Summary This topic has been presented:Understanding AWX REST API resources and how to interact with them;Installing and configuring the tower-cli tool; Work with CLI, REST API and web interface for AWX management. History[1] - - - ReadingNext is a Playbook to automate the interaction of AWX REST API:See the first page of this guide: guide: guide: