Ansible Tower User Guide

Release Ansible Tower 3.4.4

Red Hat, Inc.

May 05, 2021

CONTENTS

1	Over	view	2
	1.1	Real-time Playbook Output and Exploration	2
	1.2		2
	1.3		2
	1.4	Cloud & Autoscaling Flexibility	3
	1.5		3
	1.6		3
	1.7		3
	1.8		3
	1.9		4
	1.10		4
	1.11		4
	1.12		4
	1.13		4
	1.14		4
	1.15		5
	1.16		5
	1.17		5
	1.18		5
	1.19		5
	1.20	11	5
	1.21		6
	1.22		6
2	Towe	er Licensing, Updates, and Support	7
-	2.1		, 7
	2.2		, 7
	2.2		, 7
	2.3		8
	2.5		8
		1	
3	Logg	ing In	9
4	Impo	ort a License 10	0
	4.1	Adding a Tower License Manually 1	1
5	The T	Tower User Interface 11	3
	5.1	Activity Streams	3
	5.2	Views	5
	5.3	Resources and Access	8

	5.4	Tower Administration Menu 19
6	Searc	21
v	6.1	Searching Tips
	6.2	Sort
7	0	nizations 25
	7.1	Creating a New Organization
8	Users	36
Ŭ	8.1	Create a User
	8.2	User Types - Quick View
	8.3	Users - Organizations
	8.4	Users - Teams
	8.5	Users - Permissions
	8.6	Users - Tokens
9	Team	44
	9.1	Create a Team
10		entials 50
	10.1	Understanding How Credentials Work
	10.2	Getting Started with Credentials
		Add a New Credential 52 Credential Types 54
	10.4	Greuential Types
11	Cust	om Credential Types 66
		Backwards-Compatible API Considerations
		Getting Started with Credential Types 67
	11.3	Create a New Credential Type
12	Annli	ications 73
14		Getting Started with Applications 73
		Create a new application
13	Proje	
		Add a new project
		Work with Permissions 82 Work with Notifactions 86
	13.3	Work with Notifications 86 Work with Job Templates 87
		Work with Schedules 87 87
	1010	
14	Inver	ntories 90
	14.1	Smart Inventories 92
	14.2	Add a new inventory
	14.3	Running Ad Hoc Commands
15	Job 1	Templates 118
	15.1	Create a Job Template
	15.2	Add Permissions
	15.3	Work with Notifications
	15.4	View Completed Jobs
	15.5	Scheduling
	15.6	Surveys
	15.7 15.8	Launch a Job Template 134 Copy a Job Template 137
	10.0	Copy a Job Template

	15.10 15.11 15.12	Scan Job Templates 1 Fact Caching 1 Utilizing Cloud Credentials 1 Provisioning Callbacks 1 Extra Variables 1	41 43 46
16	16.2	licing 1 Job slice considerations 1 Job slice execution behavior 1 Search job slices 1	52
17	17.2 17.3	flows 1 Workflow scenarios and considerations 1 Extra Variables 1 Workflow States 1 Role-Based Access Controls 1	157 158
18	18.1 18.2 18.3 18.4 18.5 18.6 18.7 18.8 18.9	flow Job Templates1Create a Workflow Template1Work with Permissions1Work with Notifications1View Completed Jobs1Work with Schedules1Surveys1Workflow Visualizer1Launch a Workflow Template1Copy a Workflow Template1Extra Variables1	63 64 65 67 69 71 80 81
19		1	1 83 183
20	20.2 20.3	Job Details - Inventory Sync 1 Job Details - SCM 1 Job Details - Playbook Run 1 Ansible Tower Capacity Determination and Job Impact 1	91 92
21	21.1 21.2 21.3 21.4	Notifier Hierarchy 2 Workflow 2 Create a Notification Template 2 Notification Types 2	200 200 201 201 202 202
22	 22.1 22.2 22.3 22.4 22.5 	Create Insights Credential 2 Create an Insights Project 2 Create Insights Inventory 2 Create a Scan Project 2	209 209 211 212 213 214 216
23			220 220

	23.2	Ansible file and directory structure	220			
	23.3	Use Dynamic Inventory Sources	220			
	23.4	Variable Management for Inventory	221			
	23.5	Autoscaling	221			
	23.6	Larger Host Counts	221			
	23.7	Continuous integration / Continuous Deployment	221			
24	Secur	ity	222			
	24.1	Playbook Access and Information Sharing	222			
	24.2	Role-Based Access Controls	224			
	24.3	Function of roles: editing and creating	232			
25	Index		235			
26	Сору	right © 2019 Red Hat, Inc.	236			
Inc	lex	237				

Thank you for your interest in Red Hat Ansible Tower. Ansible Tower is a commercial offering that helps teams manage complex multi-tier deployments by adding control, knowledge, and delegation to Ansible-powered environments.

The Ansible Tower User Guide discusses all of the functionality available in Ansible Tower and assumes moderate familiarity with Ansible, including concepts such as **Playbooks**, **Variables**, and **Tags**. For more information on these and other Ansible concepts, please see the Ansible documentation at http://docs.ansible.com/. This document has been updated to include information for the latest release of Ansible Tower 3.4.4.

We Need Feedback!

If you spot a typo in this documentation, or if you have thought of a way to make this manual better, we would love to hear from you! Please send an email to: docs@ansible.com

If you have a suggestion, try to be as specific as possible when describing it. If you have found an error, please include the manual's title, chapter number/section number, and some of the surrounding text so we can find it easily. We may not be able to respond to every message sent to us, but you can be sure that we will be reading them all!

Ansible Tower Version 3.4.4; July 10, 2019; https://access.redhat.com/

CHAPTER

OVERVIEW

Thank you for your interest in Ansible Tower. Tower is a graphically-enabled framework accessible via a web interface and a REST API endpoint for Ansible, the open source IT orchestration engine. Whether sharing operations tasks with your team or integrating with Ansible through the Tower REST API, Tower provides many powerful tools to make your automation life easier.

1.1 Real-time Playbook Output and Exploration

Watch playbooks run in real time, seeing each host as they check in. Easily go back and explore the results for specific tasks and hosts in great detail. Search for specific plays or hosts and see just those results, or quickly zero in on errors that need to be corrected.

1.2 "Push Button" Automation

Access your favorite projects and re-trigger execution from the web interface with a minimum of clicking. Tower will ask for input variables, prompt for your credentials, kick off and monitor the job, and display results and host history over time.

1.3 Enhanced and Simplified Role-Based Access Control and Auditing

Ansible Tower allows for the granting of permissions to perform a specific task (such as to view, create, or modify a file) to different teams or explicit users through role-based access control (RBAC).

Keep some projects private, while allowing some users to edit inventory and others to run playbooks against only certain systems–either in check (dry run) or live mode. You can also allow certain users to use credentials without exposing the credentials to them. Regardless of what you do, Tower records the history of operations and who made them–including objects edited and jobs launched.

Based on user feedback, Ansible Tower both expands and simplifies its role-based access control. No longer is job template visibility configured via a combination of permissions on inventory, projects, and credentials. If you want to give any user or team permissions to use a job template, just assign permissions directly on the job template. Similarly, credentials are now full objects in Tower's RBAC system, and can be assigned to multiple users and/or teams for use.

A new 'Auditor' type has been introduced in Tower as well, who can see all aspects of the systems automation, but has no permission to run or change automation, for those that need a system-level auditor. (This may also be useful for a service account that scrapes automation information from Tower's API.) Refer to *Role-Based Access Controls* for more information.

Subsequent releases of Ansible Tower provides more granular permissions, making it easier to delegate inside your organizations and remove automation bottlenecks.

1.4 Cloud & Autoscaling Flexibility

Tower features a powerful provisioning callback feature that allows nodes to request configuration on demand. While optional, this is an ideal solution for a cloud auto-scaling scenario, integrating with provisioning servers like Cobbler, or when dealing with managed systems with unpredictable uptimes. Requiring no management software to be installed on remote nodes, the callback solution can be triggered via a simple call to 'curl' or 'wget', and is easily embeddable in init scripts, kickstarts, or preseeds. Access is controlled such that only machines in inventory can request configuration.

1.5 The Ideal RESTful API

The Tower REST API is the ideal RESTful API for a systems management application, with all resources fully discoverable, paginated, searchable, and well modeled. A styled API browser allows API exploration from the API root at http://<Tower server name>/api/, showing off every resource and relation. Everything that can be done in the user interface can be done in the API - and more.

1.6 Backup and Restore

The ability to backup and restore your system(s) has been integrated into the Tower setup playbook, making it easy for you to backup and replicate your Tower instance as needed.

1.7 Ansible Galaxy Integration

When it comes to describing your automation, everyone repeats the DRY mantra-"Don't Repeat Yourself." Using centralized copies of Ansible roles, such as in Ansible Galaxy, allows you to bring that philosophy to your playbooks. By including an Ansible Galaxy requirements.yml file in your project directory, Tower automatically fetches the roles your playbook needs from Galaxy, GitHub, or your local source control. Refer to *Ansible Galaxy Support* for more information.

1.8 Inventory Support for OpenStack

Ansible is committed to making OpenStack simple for everyone to use. As part of that, dynamic inventory support has been added for OpenStack. This allows you to easily target any of the virtual machines or images that you're running in your OpenStack cloud.

1.9 Remote Command Execution

Often times, you just need to do a simple task on a few hosts, whether it's add a single user, update a single security vulnerability, or restart a misbehaving service. Beginning with version 2.2.0, Tower includes remote command execution–any task that you can describe as a single Ansible play can be run on a host or group of hosts in your inventory, allowing you to get managing your systems quickly and easily. Plus, it is all backed by Tower's RBAC engine and detailed audit logging, removing any questions regarding who has done what to what machines.

1.10 System Tracking

System tracking (historical facts) feature was deprecated starting with Ansible Tower 3.2. However, you can collect facts by using the fact caching feature. Refer to *Fact Caching* for more detail.

1.11 Integrated Notifications

Starting with version 3.0, Ansible Tower allows you to easily keep track of the status of your automation. You can configure stackable notifications for job templates, projects, or entire organizations, and configure different notifications for job success and job failure. The following notification sources are supported: - Slack - E-mail - SMS (via Twilio) - HipChat - Pagerduty - IRC - Webhooks (post to an arbitrary webhook, for integration into other tools)

1.12 Satellite and CloudForms Integration

Ansible Tower 3.0 also adds dynamic inventory sources for Red Hat Satellite 6 and Red Hat CloudForms.

1.13 Run-time Job Customization

Bringing the flexibility of the command line to Tower, you can now prompt for any of the following:

- inventory
- credential
- job tags
- limits

1.14 Red Hat Insights Integration

Ansible Tower 3.1 supports integration with Red Hat Insights, which allows Insights playbooks to be used as a Tower Project.

1.15 Enhanced Tower User Interface

In Ansible Tower 3.3, the layout of the user interface was reorganized to improve navigational elements. With more information displayed at-a-glance, it is more intuitive to find and use the automation you need.

1.16 Custom Virtual Environments

Custom Ansible environment support allows you to have different Ansible environments for different teams and jobs.

1.17 Authentication Enhancements

Ansible Tower 3.3 enhanced LDAP and SAML support and introduced token-based authentication. Enhanced LDAP and SAML support allows you to integrate your enterprise account information in a more flexible manner. Token-based Authentication allows for easy authentication of third-party tools and services with Tower via integrated OAuth 2 token support.

1.18 Cluster Management

Run-time management of cluster groups allows for easily configurable scaling.

1.19 Container Platform Support

Tower is available as a containerized pod service for Red Hat OpenShift Container Platform that can be scaled up and down easily as needed.

1.20 Workflow Enhancements

In order to better model your complex provisioning, deployment, and orchestration workflows, Ansible Tower expanded workflows in a number of ways:

- **Inventory overrides for Workflows**. You can now override an inventory across a workflow at workflow definition time, or even at launch time. Define your application deployment workflow, and then easily re-use them in multiple environments.
- **Convergence nodes for Workflows**. When modeling complex processes, you sometimes need to wait for multiple steps to finish before proceeding. Now Ansible Tower workflows can easily replicate this; workflow steps can now wait for any number of prior workflow steps to complete properly before proceeding.
- Workflow Nesting. Re-use individual workflows as components of a larger workflow. Examples include combining provisioning and application deployment workflows into a single master workflow.

1.21 Job Distribution

As automation moves enterprise-wide, the need to automate at scale grows. Now with Ansible Tower 3.4, we offer the ability to take a fact gathering or configuration job running across thousands of machines and slice it into individual job slices that can be distributed across your Ansible Tower cluster for increased reliability, faster job completion, and better cluster utilization. If you need to change a parameter across 15,000 switches at scale, or gather information across your multi-thousand-node RHEL estate, you can now do so easily.

1.22 Support for deployment in a FIPS-enabled environment

If you require running your environment in restricted modes such as FIPS, Ansible Tower now deploys and runs in such environments.

TOWER LICENSING, UPDATES, AND SUPPORT

Red Hat Ansible Tower ("Ansible Tower") is a software product provided as part of an annual subscription entered into between you and Red Hat, Inc. ("Red Hat").

Ansible is an open source software project and is licensed under the GNU General Public License version 3, as detailed in the Ansible source code: https://github.com/ansible/ansible/blob/devel/COPYING

2.1 Support

Red Hat offers support to paid Red Hat Ansible Automation customers.

If you or your company has purchased a subscription for Ansible Automation, you can contact the support team at https://access.redhat.com. To better understand the levels of support which match your Ansible Tower Subscription, refer to *Subscription Types*. For details of what is covered under an Ansible Automation subscription, please see the Scopes of Support at: https://access.redhat.com/support/policy/updates/ansible-tower#scope-of-coverage-4 and https://access.redhat.com/support/policy/updates/ansible-engine.

2.2 Trial / Evaluation

While a license is required for Ansible Tower to run, there is no fee for a trial license.

- Trial licenses for Red Hat Ansible Automation are available at: http://ansible.com/license
- Support is not included in a trial license or during an evaluation of the Tower Software.

2.3 Subscription Types

Red Hat Ansible Automation is provided at various levels of support and number of machines as an annual Subscription.

- Standard (F.K.A. "Enterprise: Standard")
 - Manage any size environment
 - Enterprise 8x5 support and SLA
 - Maintenance and upgrades included
 - Review the SLA at: https://access.redhat.com/support/offerings/production/sla
 - Review the Red Hat Support Severity Level Definitions at: https://access.redhat.com/support/policy/ severity

- Enterprise (F.K.A. "Enterprise: Premium")
 - Manage any size environment, including mission-critical environments
 - Premium 24x7 support and SLA
 - Maintenance and upgrades included
 - Review the SLA at: https://access.redhat.com/support/offerings/production/sla
 - Review the Red Hat Support Severity Level Definitions at: https://access.redhat.com/support/policy/ severity

All Subscription levels include regular updates and releases of Ansible Tower.

For more information, contact Ansible via the Red Hat Customer portal at https://access.redhat.com/ or at http://www. ansible.com/pricing/.

2.4 Node Counting in Licenses

The Tower license defines the number of Managed Nodes that can be managed by Ansible Tower. A typical license will say 'License Count: 500', which sets the maximum number of Managed Nodes at 500.

Ansible Tower counts Managed Nodes by the number of nodes in inventory. If more Managed Nodes are in the Ansible Tower inventory than are supported by the license, you will be unable to start any Jobs in Ansible Tower. If a dynamic inventory sync causes Ansible Tower to exceed the Managed Node count specified in the license, the dynamic inventory sync will fail.

For more information on managed node requirements for licensing, please see https://access.redhat.com/articles/ 3331481.

2.5 Tower Component Licenses

To view the license information for the components included within Ansible Tower, refer to /usr/share/doc/ ansible-tower-<version>/README where <version> refers to the version of Ansible Tower you have installed.

To view a specific license, refer to /usr/share/doc/ansible-tower-<version>/*.txt, where * is replaced by the license file name to which you are referring.

CHAPTER

THREE

LOGGING IN

To log in to Tower, browse to the Tower interface at: http://<Tower server name>/

ANSIBLE TOWER by Red Hat*
Welcome to Ansible Tower! Please sign in.
USERNAME
admin
PASSWORD
•••••
SIGN IN

Log in using a valid Tower username and password.

The default username and password set during installation are *admin* and *password*, but the Tower administrator may have changed these settings during installation. If the default settings have not been changed, you can do so by

accessing the Users link from the Settings () Menu.

CHAPTER

FOUR

IMPORT A LICENSE

Tower requires a valid license to run. If you did not receive a license from Ansible directly or via email, or have issues with the license you received, refer to http://www.ansible.com/license for free and paid license options (including free trial licenses) or contact Ansible via the Red Hat Customer portal at https://access.redhat.com/.

Note: To successfully add your license, you must be logged on as the Superuser. Otherwise, the operation will fail.

LICENSE MANAGEMENT
Choose your license file, agree to the End User License Agreement, and click submit.
* LICENSE FILE
BROWSE No file selected.
* END USER LICENSE AGREEMENT
ANSIBLE TOWER BY RED HAT END USER LICENSE AGREEMENT
This end user license agreement ("EULA") governs the use of the Ansible Tower software and any related updates, upgrades, versions, appearance, structure and organization (the "Ansible Tower Software"), regardless of the delivery mechanism.
1. License Grant. Subject to the terms of this EULA, Red Hat, Inc. and its affiliates ("Red Hat") grant to you
I agree to the End User License Agreement

To add your license:

1. Save your license (or save the license contents to a text file locally, if needed).



- 2. Click the Settings () icon from the left navigation bar and select the License tab from the Settings screen.
- 3. Click the **Browse** button and navigate to the location where the license file is saved to upload it. The uploaded license may be a plain text file or a JSON file, and must include properly formatted JSON code.
- 4. Once uploaded, check to agree to the End User License Agreement and click Submit.

Once your license has been accepted, Tower navigates you to the main Ansible interface for the Dashboard (which you can access by clicking on the Ansible Tower logo at the top left of the screen as well).

For later reference, you can view this license from the License tab of the Settings screen, accessible through the

Settings () icon from the left navigation bar.

DETAILS		LICENSE MANAGEMENT
ICENSE	Valid License	Choose your license file, agree to the End User License Agreement, and click submit.
/ERSION	3.4.0	* LICENSE FILE
ICENSE TYPE	Enterprise	BROWSE No file selected.
UBSCRIPTION	Enterprise Tower Up To 30 Nodes	* END USER LICENSE AGREEMENT
ICENSE KEY	367ce98cba3e62f4c5665c567f60 18ec74fdb35530247d6e0bcbe75 3cbb8fa88	ANSIBLE TOWER BY RED HAT END USER LICENSE AGREEMENT This end user license agreement ("EULA") governs the use of the Ansible Tower software and any related updates, upgrades, versions,
EXPIRES ON	01/01/2025	appearance, structure and organization (the "Ansible Tower Software"), regardless of the delivery mechanism.
IME REMAINING	2262 Days	 License Grant. Subject to the terms of this EULA, Red Hat, Inc. and its affiliates ("Red Hat") grant to you ("You") a non-transferable, non- exclusive, worldwide, non-sublicensable, limited, revocable license to use the Ansible Tower Software for the term of the associated Red Hat
OSTS AVAILABLE	30	I agree to the End User License Agreement
IOSTS USED	1	SUBMIT
OSTS REMAINING	29	
f you are ready to upg outton below	grade, please contact us by clicking the	

4.1 Adding a Tower License Manually

If you are in a situation where uploading a file is not allowed due to a locked down environment, you can add the Ansible Tower license by hand using Tower's API.

Note: To successfully add your license, you must be logged on as the Superuser. Otherwise, the operation will fail. Use only the procedure described here for applying a license via the API. Do not put the license in a file, and manually placing it in the license directory of your Ansible Tower install. The ability to do so has been deprecated in version 3.1.0.

To add the license file manually:

- 1. In Tower's REST API, at the /api/v2/config/ endpoint, scroll down to the POST text entry box.
- 2. Add your valid license, the one you received directly from Ansible, to the POST box using the following as an example:

3. When finished, click the **POST** button and review your license.

THE TOWER USER INTERFACE

The Tower User Interface offers a friendly graphical framework for your IT orchestration needs. The left navigation bar provides quick access to resources, such as **Projects**, **Inventories**, **Job Templates**, and **Jobs**.

Across the top-right side of the interface, you can access your user profile, the About page, view related documentation, and log out. Right below these options, you can view the activity stream for that user by clicking on the Activity Stream



5.1 Activity Streams

Most screens in Tower have an Activity Stream (¹¹) button. Clicking this brings up the Activity Stream for this object.

ARCH		Q. KEY	All Activity
IME 👻	INITIATED BY \$	EVENT	ACTIONS
/29/2016 10:20:41 AM	admin	created job Demo Job Template	Q
/29/2016 9:12:08 AM	admin	updated team Production Operations	Q
/29/2016 9:11:15 AM	admin	associated Production Operations member_role to jdoge	Q
/29/2016 9:11:15 AM	admin	associated Production Operations admin_role to gdoge	Q
/29/2016 9:10:54 AM	admin	created team Production Operations	Q
/29/2016 9:10:12 AM	admin	associated cdoge member_role to Honey Dog, Inc.	Q
/29/2016 9:10:12 AM	admin	updated user cdoge	Q
/29/2016 9:10:12 AM	admin	created user cdoge	Q
/29/2016 9:09:39 AM	admin	associated jdoge member_role to Honey Dog, Inc.	Θ
/29/2016 9:09:39 AM	admin	updated user jdoge	Q
/29/2016 9:09:39 AM	admin	created user jdoge	Q
/29/2016 9:09:12 AM	admin	associated gdoge member_role to Honey Dog, Inc.	Q
/29/2016 9:09:12 AM	admin	updated user gdoge	Θ
/29/2016 9:09:12 AM	admin	created user gdoge	Q
/29/2016 9:08:40 AM	admin	associated admin member_role to Honey Dog, Inc.	Θ
/29/2016 9:08:25 AM	admin	created organization Honey Dog, Inc.	Θ
/29/2016 9:08:25 AM	admin	associated system_auditor to Honey Dog, Inc.	Θ
/29/2016 9:08:25 AM	admin	associated system_administrator to Honey Dog, Inc.	Θ
/29/2016 9:08:02 AM	admin	Event summary not available	Q
/29/2016 9:08:02 AM	admin	Event summary not available	Q

An Activity Stream shows all changes for a particular object. For each change, the Activity Stream shows the time of

the event, the user that initiated the event, and the action. Clicking on the Examine () button shows the event log for the change.

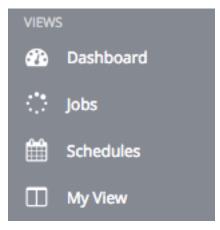
ACTIVITY STREAM		•
SEARCH	EVENT 8	All Activity *
TIME 🗸 INITIA	INITIATED BY system on 2/27/2017 12:39:31 PM TED ACTION created host localhost	ACTIONS
2/27/2017 12:39:31 PM system		Q
2/27/2017 12:39:31 PM system	"name": "localhost",	Q
2/27/2017 12:39:31 PM system	"instance_id": "",	ପ୍
2/27/2017 12:39:30 PM system	"inventory": "Demo Inventory-1", "id": 1, "description": ""	ପ୍
2/27/2017 12:39:30 PM system		Q
2/27/2017 12:39:30 PM system	ОК	Q
2/27/2017 12:39:30 PM system	n Event summary not available	Q
2/27/2017 12:39:30 PM syste	n created project Demo Project	Q

The Activity Stream can be filtered by the initiating user (or the system, if it was system initiated), and by any related Tower object, such as a particular credential, job template, or schedule.

The Activity Stream on the main Dashboard shows the Activity Stream for the entire Tower instance. Most pages in Tower allow viewing an activity stream filtered for that specific object.

5.2 Views

The Tower User Interface provides several options for viewing information.

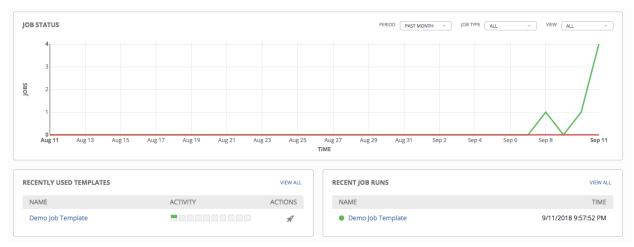


5.2.1 Dashboard view

The **Dashboard** view begins with a summary of your hosts, inventories, and projects. Each of these is linked to the corresponding objects in Tower for easy access.

1	0	1	0	1	0
HOSTS	FAILED HOSTS	INVENTORIES	INVENTORY SYNC FAILURES	PROJECTS	PROJECT SYNC FAILURES

On the main Tower Dashboard screen, a summary appears listing your current **Job Status**. Also available for review are summaries of **Recently Used Templates** and **Recent Job Runs**.



The **Job Status** graph displays the number of successful and failed jobs over a specified time period. You can choose to limit the job types that are viewed, and to change the time horizon of the graph.

The Recently Used Templates section of this display shows a summary of the most recently used templates. You can



also access this summary by clicking the Templates () icon from the left navigation bar.

The **Recent Job Runs** section displays which jobs were most recently run, their status, and time when they were run as well.

Note: Clicking on the Dashboard () icon from the left navigation bar or the Ansible Tower logo at any time returns you to the Dashboard.

5.2.2 Jobs view



Access the **Jobs** view by clicking the Jobs (**Jobs**) icon from the left navigation bar. This view shows all the jobs that have ran in Tower, including projects, templates, management jobs, SCM updates, playbook runs, etc.

A TOWER	💄 admin	0		ტ
JOBS				٥
JOBS T3 Jobs SEARCH Q KEY				
My View • 17 - Demo Job Template Playbook Run Resources STARTED 9/11/2018 11:33:38 PM FINISHED 9/11/2018 11:33:43 PM				
Credentials LAUNCHED BY admin JOB TEMPLATE Demo Job Template Q Credentials INVENTORY PROJECT Demo Project		ø	Û	
CREDENTIALS & Demo Credential				_
inventory Scripts To - Demo Project Scki update Access STARTED 9/11/2018 11:33:33 PM PROJECT Demo Project		R	Û	
Users Id - Demo Job Template Playbook Run STARTED 9/11/2018 11:33:10 PM FINISHED 9/11/2018 11:33:16 PM				
ADMINISTRATION JOB TEMPLATE Demo Job Template INVENTORY Demo Inventory PROJECT Demo Project		đ	Ŵ	
Notifications CREDENTIALS CREDENT				_
Instance Groups STARTED 9/11/2018 FINISHED 9/11/2018 11:33:10 PM Applications PROJECT Demo Project Demo Project		R	Ŵ	
Settings				

5.2.3 Schedules view



Access the **Schedules** view by clicking the Schedules (**Lease**) icon from the left navigation bar. This view shows all the scheduled jobs that are configured.

A TOWER				💄 admin	0		
≡	SCHEDULES						
VIEWS							
🚯 Dashboard	SCHEDULED JOBS						
::: Jobs	SEARCH	Q KEY					
Schedules							
My View	NAME 🗢	TYPE 🔺	NEXT RUN 🗢		AC	TIONS	
RESOURCES	Cleanup Job Schedule	Management Job	9/16/2018 12:46:50 PM		All	Î	
🖋 Templates	Cleanup Activity Schedule	Management Job	9/18/2018 12:46:50 PM		AP.	Ê	
ୟ Credentials						ITEMS 1-2	
🗁 Projects							

5.2.4 My View

My View, is a user's single-page view of jobs and job templates. It can be accessed by clicking the My View (III) icon from the left navigation bar or by navigating to https://<Tower server name>/portal.

My View is a simplified interface for users who need to run Ansible jobs, but that do not need an advanced knowledge of Ansible or Tower. My View could be used by, for instance, development teams, or even departmental users in non-technical fields.

My View offers Tower users a simplified, clean interface to the jobs that they are able to run, and the results of jobs that they have run in the past.

Pressing the button beside a job in My View launches it, potentially asking some survey questions if the job is configured to do so.

A TOWER		🛔 admin 🚯 🗐 (
	MY VIEW	
Dashboard	JOB TEMPLATES	JOBS C MYJOBS ALLJOBS
∴ Jobs ∰ Schedules	SEARCH Q KEY	SEARCH Q KEY
My View	Demo Job Template Job Template	17 - Demo Job Template Playbook Run
RESOURCES	ACTIVITY Demo Inventory	STARTED 9/11/2018 11:33:38 PM FINISHED 9/11/2018 11:33:43 PM
🦧 Credentials	PROJECT Demo Project 🖋 CREDENTIALS 4 Demo Credential	LAUNCHED BY admin JOB TEMPLATE Demo Job Template
Projects	LAST MODIFIED 9/11/2018 11:33:43 PM by admin LAST RAN 9/11/2018 11:33:43 PM	INVENTORY Demo Inventory PROJECT Demo Project
	ПЕМS 1-1	CREDENTIALS
ACCESS		STARTED 9/11/2018 11:33:10 PM
Users		FINISHED 9/11/2018 11:33:16 PM LAUNCHED BY admin
😁 Teams		JOB TEMPLATE Demo Job Template INVENTORY Demo Inventory
ADMINISTRATION		PROJECT Demo Project CREDENTIALS Q Demo Credential

My View displays two main sections-Job Templates and Jobs. The Job Templates panel shows the job templates that

are available to be run. To launch a job template, click the button. This launches the job, which can be viewed in the Jobs panel.

The Jobs pane shows the list of jobs that have run in the past. Sort for jobs specific to you by clicking the **My Jobs** button or review all jobs you have access to view by clicking the **All Jobs** button, above the search bar.

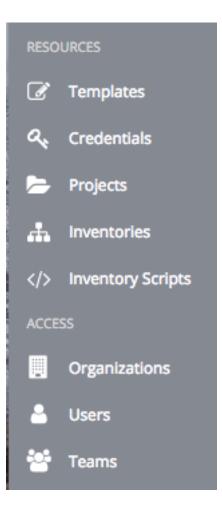
- My Jobs: View jobs that you (as the user) ran.
- All Jobs: View your team members' completed jobs, viewable based on your RBAC permissions.

For each job, you can view and sort by any number of the job's attributes shown. Clicking on the link for the job opens a new window with the **Job Details** for that job (refer to *Jobs* for more information).

Other portions of the interface are hidden from view until My View is exited.

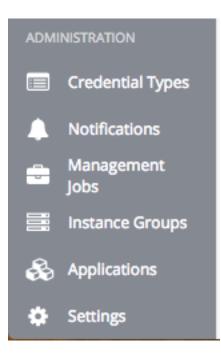
5.3 Resources and Access

The **Resources** and **Access** menus provide you access to the various components of Ansible Tower and allow you to configure who has permissions for which of those resources.



5.4 Tower Administration Menu

The Administration menu provides access to the various administrative options:



From here, you can create, view, and edit *custom credential types*, *notifications*, management jobs, *tokens and applications*, and configure Tower settings. Configuring Tower settings is accomplished through the **Settings** menu, which is described in further detail in the proceeding section.

5.4.1 Settings Menu

Starting with Ansible Tower 3.0, the Settings () menu offers access to administrative configuration options. Users of older versions of Ansible Tower (2.4.5 or older) can access most of these through the top-level navigational menu or from their "Setup" menu button.

To enter the Settings window for Ansible Tower, click the Settings icon at the bottom of the left navigation bar. This page allows you to modify your Tower's configuration, such as settings associated with authentication, jobs, system, user interface, and view or import your license.

INGS				
Authentication	Jobs	System	User Interface	License
Enable simplified login for your Tower applications	Update settings pertaining to Jobs within Tower	Define system-level features and functions	Set preferences for data collection, logos, and logins	View and edit your license information

For more information on configuring these settings, refer to Tower Configuration section of the Ansible Tower Administration Guide.

For further detail on configuring these options, refer to the Tower Configuration section.

CHAPTER

SEARCH

Ansible Tower release 3.1 introduced the Tower Search, a powerful search tool that provides both search and filter capabilities that span across multiple functions.

SEARCH	Q	KEY
--------	---	-----

Acceptable search criteria are provided in an expandable "cheat-sheet" accessible from the Key button.

JDOGE AUDITOR	0
DETAILS ORGANIZATIONS TEAMS PERMISSIONS	×
SEARCH	д <mark>ке</mark> у
EXAMPLES: Id>10 created>="2000-01-01T00:00:002" created:<2000-01-01 name:foobar	
FIELDS: id, type, created, modified, name, description	
RELATED FIELDS: modified_by, workflow_job_template, label, project, inventory, activity_stream, notification_templates_any, notification_templ	on_templates_error, team, credential, notification_template, custom_inventory_script, created_by, notification_templates_success
ADDITIONAL INFORMATION: For additional information on advanced search search syntax please see the Ansible Tower documenta	ition.
NAME *	DESCRIPTION \$
Default	
	ITEMS 1 - 1 OF 1
	TEMS 1-10*1
Use the Clear All to clear the search criteria.	

SEARCH		Q	KEY
× hartman × brown CLEAR	ALL		

6.1 Searching Tips

These searching tips assume that you are not searching hosts. Most of this section still applies to hosts but with some subtle differences. A typical syntax of a search consists a field (left-hand side) and a value (right-hand side). A colon is used to separate the field that you want to search from the value. If a search doesn't have a colon (see example 3) it is treated as a simple string search where ?search=foobar is sent. Here are the examples of syntax used for searching:

1. name:localhost In this example, the string before the colon represents the field that you want to search on. If that string does not match something from **Fields** or **Related Fields** then it's treated the same way Example 3 is (string search). The string after the colon is the string that you want to search for within the name attribute.

- 2. organization.name:Default This example shows a Related Field Search. The period in the left-hand portion separates the model from the field in this case. Depending on how deep/complex the search is, you could have multiple periods in that left-hand portion.
- 3. foobar Simple string (key term) search that will find all instances of that term using an icontains search against the name and description fields. If a space is used between terms (e.g. foo bar), then any results that contain both terms will be returned. If the terms are wrapped in quotes (e.g. "foo bar"), Tower will search for the entire string with the terms appearing together. Specific name searches will search against the API name. For example, Management job in the user interface is system_job in the API.
- 4. organization: Default This example shows a Related Field search but without specifying a field to go along with the organization. This is supported by the API and is analogous to a simple string search but done against the organization (will do an icontains search against both the name and description).

6.1.1 Values for search fields

To find values for certain fields, refer to the API endpoint for extensive options and their valid values. For example, if you want to search against /api/v2/jobs -> type field, you can find the values by performing an **OPTIONS** request to /api/v2/jobs and look for entries in the API for "type". Additionally, you can view the related searches by scrolling to the bottom of each screen. In the example for /api/v2/jobs, the related search shows:



The values for Fields come from the keys in a **GET** request. url, related, and summary_fields are not used. The values for Related Fields also come from the **OPTIONS** response, but from a different attribute. Related Fields is populated by taking all the values from related_search_fields and stripping off the __search from the end.

Any search that does not start with a value from Fields or a value from the Related Fields, will be treated as a generic string search. Searching for something like localhost will result in the UI sending ?search=localhost as a query parameter to the API endpoint. This is a shortcut for an icontains search on the name and description fields.

6.1.2 Searching using values from Related Fields

Searching a Related Field requires you to start the search string with the Related Field. This example describes how to search using values from the Related Field, *organization*.

The left-hand side of the search string must start with *organization* (ex: organization:Default). Depending on the related field, you might want to provide more specific direction for the search by providing secondary/tertiary fields. An example of this would be to specify that you want to search for all job templates that use a project matching a certain name. The syntax on this would look like: job_template.project.name:"A Project".

Note: This query would execute against the unified_job_templates endpoint which is why it starts with job_template. If we were searching against the job_templates endpoint, then you wouldn't need the job_template portion of that query.

6.1.3 Other search considerations

The following are a few things about searching in Tower that you should be aware of:

- There's currently no supported syntax for **OR** queries. All search terms get **AND**'d in the query parameters.
- The left-hand portion of a search parameter can be wrapped in quotes to support searching for strings with spaces.
- Currently, the values in the Fields are direct attributes expected to be returned in a GET request. Whenever you search against one of the values, Tower essentially does an __icontains search. So, for example, name:localhost would send back ?name__icontains=localhost. Tower currently performs this search for every Field value, even id, which is not ideal.

6.2 Sort

Where applicable, use the arrows in each column to sort by ascending or descending order (following is an example from the schedules list).

SEARCH	Click to change sort or	der Q KEY	Sort order set at ascending (up arrow)	
NAME 💠 🗡		туре	NEXT RUN 🗢	ACTIONS
ON sync vmware	inventory daily	Inventory Sync	12/18/2018 9:30:00 PM	Q
OFF Duplicate and	upgrade tower.testing on a schedule	Playbook Run	11/29/2018 4:15:00 AM	Q
ON cleanup towe	r.testing docker stuff	Playbook Run	12/22/2018 10:00:00 PM	Θ
ON Build Develop	ment Container Image	Playbook Run	12/19/2018 12:30:00 AM	Q

The direction of the arrow indicates the sort order of the column.

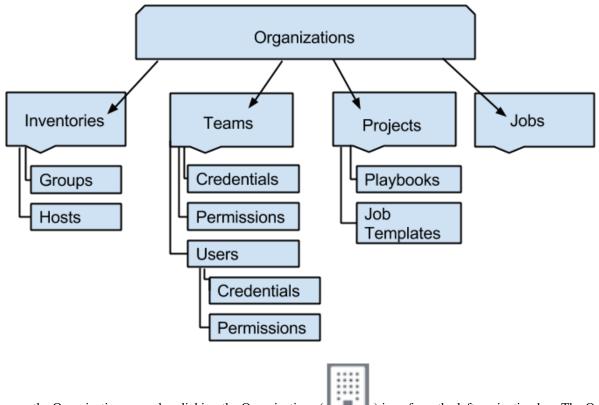
HEDULED JOBS 13			
EARCH	Q KEY		
NAME 🔶	TYPE 🗢	NEXT RUN 🗢	ACTIONS
ON Biweekly	Workflow Job	12/21/2018 9:00:00 PM	Θ
ON Build Developer Isolated Container Image	Playbook Run	12/19/2018 12:30:00 AM	ପ୍
ON Build Development Container Image	Playbook Run	12/19/2018 12:30:00 AM	Q
ON Check users daily	Playbook Run	12/19/2018 5:00:00 AM	Q
ON Cleanup Activity Stream Schedule	Management Job	12/25/2018 1:00:00 AM	ବ୍
ON Cleanup Job Details Schedule	Management Job	12/22/2018 10:00:00 PM	Q
ON Cleanup tower.testing docker stuff	Playbook Run	12/22/2018 10:00:00 PM	ଷ୍
OFF Duplicate and upgrade tower.testing on a schedu	e Playbook Run	11/29/2018 4:15:00 AM	ବ୍
ON Every day at 2am	Playbook Run	12/18/2018 9:02:00 PM	Q

CHAPTER

SEVEN

ORGANIZATIONS

An Organization is a logical collection of Users, Teams, Projects, and Inventories, and is the highest level in the Tower object hierarchy.



Access the Organizations page by clicking the Organizations () icon from the left navigation bar. The Organizations page displays all of the existing organizations for your installation of Tower. Organizations can be searched by **Name** or **Description**. Modify and remove organizations using the **Edit** and **Delete** buttons.

Note: Tower creates a default organization automatically. Users of Tower with a Self-Support level license (formerly called Basic) only have the default organization available and should **not** delete it. Users of older versions of Tower (prior to 2.2) will not see this default organization.

A	TOWER	🛔 admin	0		ሳ
≡	ORGANIZATIONS				
4					
0	ORGANIZATIONS				
	SEARCH Q KEY			+	
	Default e 🖻				
ľ	O USERS O TEAMS				
Q _t	INVENTORIES PROJECTS				
6	1 JOB TEMPLATES O ADMINS				
#					
				ITEMS 1-	1

7.1 Creating a New Organization

"Enterprise: Standard" and "Enterprise: Premium" Tower licenses allow you to create a new Organization by selecting



Note: If you are using Ansible Tower with a Self-Support level license (formerly called Basic), you must use the default Organization. Do not delete it and try to add a new Organization, or you will break your Tower setup. Only two Tower license types (Enterprise: Standard or Enterprise: Premium) have the ability to add new Organizations beyond the default.

NEW ORGANIZATION			8
DETAILS USERS PER	MISSIONS		
* NAME	DESCRIPTION	INSTANCE GROUPS 😨	
		Q	
ANSIBLE ENVIRONMENT 🔞			
Select Ansible Environment	•		

An organization has several attributes that may be configured:

- 1. Enter the Name for your Organization (required).
- 2. Enter a **Description** for the Organization.
- 3. Enter an **Instance Group** on which to run this organization.
- 4. Select from the drop-down menu list a custom virtual Ansible Environment on which to run this organization.
- 5. Click Save to finish creating the Organization.

Once created, Tower displays the Organization details, and allows for the managing of users and administrators for the Organization.

oney Dog, Inc.					8
DETAILS	PERMISSIONS				
NAME		DESCRIPTION		INSTANCE GROUPS 🔞	
Honey Dog, Inc.		A capable company making capa	ble things	Q tower ×	
NSIBLE ENVIRONMENT 🚱	×				
Select Ansible Environment		·			
					CANCEL
					CANCEL SAVE
RGANIZATIONS 2					
RGANIZATIONS		Q KEY			+ ADD
SEARCH					+ ADD
			2		+ ADD
SEARCH	TEAMS				+ ADD
SEARCH		Honey Dog, Inc.			+ ADD
Default USERS	0 TEAMS	Honey Dog, Inc.	TEAMS		+ ADD
Default O USERS INVENTORIES	TEAMSPROJECTS	Honey Dog, Inc. USERS INVENTORIES	TEAMSPROJECTS		+ ADD

7.1.1 Organizations - Users

Clicking on **Users** (beside **Details** when viewing your organization), displays all the Users associated with this Organization. A User is someone with access to Tower with associated roles and Credentials.

HONEY DOG, INC.		٢
DETAILS USERS PERMISSIONS NOTIFICATIONS		
SEARCH	Q, KEY	+ ADD
USER 🔺	ROLE	
admin	SYSTEM ADMINISTRATOR	
austin78	× ADMIN × MEMBER	
gdoge		
jdoge	SYSTEM AUDITOR	
		ITEMS 1 - 4 OF 4

As you can manage the user membership for this Organization here, you can manage user membership on a per-user

basis from the Users page by clicking the Users () icon from the left navigation bar. The user list from the Organizations view may be sorted by username. Use the Tower Search to search for users by various attributes. Click **Key** for using the search, or refer to the *Search* chapter for more information.

Clicking on a user brings up that user's details, allowing you to review, grant, edit, and remove associated permissions for that user. For more information, refer to *Users*.

Add a User

In order to add a user to an organization, the user must already be created in Tower. Refer to *Create a User* to create a user. To add existing users to the Organization:





2. Select one or more users from the list of available users by clicking the checkbox next to the user(s). Doing so expands the lower part of the Wizard to assign roles to each user.

DNEY DOG, INC.	HONEY DOG, INC. ADD US				0	G
DETAILS USERS PERMISSIONS	1 Please select Users from	the list below.				
SEARCH	SEARCH			Q	KEY	+ ADD
webb CLEAR ALL	USERNAME 🔶	FIR	ST NAME 🗢	LAST NAME 🗘		
USER *	🗆 admin					
austin78	austin78	Joh	ın	Webb		
	🛛 gdoge	Ge	rry	Doge		
	gonzalezheidi	Wil	lliam	Thomas		ITEMS 1 - 1 OF
	hartmanandrea	Ka	rla	Brown		
RGANIZATIONS 2	< 1 2 3 > PAGE 1 OF 3			ITEMS 1	- 5 OF 15	
SEARCH	2 Please assign roles to the	e selected users/teams			KEY	+ ADD
	John Webb USER	SELECT ROLES			×	
DEFAULT	Gerry Doge USER	SELECT ROLES			×	
Place organization description here						
S USERS 2 T				CANCEL		

3. For each user, click from the drop-down menu to select one or more roles for that user.

Note: For help on what the roles mean, click the **Key** button. For more information, refer to the *Roles* section of this guide.

HONEY DOG, INC.	HONEY DOG, INC. ADD USERS	.wc	8
SEARCH	SEARCH		Q KEY
USER 🛧	USERNAME 🔶	FIRST NAME 🗢	LAST NAME 🗢
admin	🗆 admin		
gdoge	austin78	John	Webb
jdoge	✓ gdoge	Gerry	Doge
	gonzalezheidi	William	Thomas
	hartmanandrea	Karla	Brown
RGANIZATIONS 2	< 1 2 3 > PAGE 1 OF 3		ITEMS 1 - 5 OF 15
SEARCH	2 Please assign roles to the selected u	sers/teams	KEY
	John Webb USER 🛛 🗙 Adr	nin 🛛 × Member	ж
DEFAULT	Gerry Doge USER × Auc	litor × Member	×
Place organization description here			CANCEL
5 USERS 2 T	EAMS USERS	O TEAMS	

In this example, two users have been selected and each have been granted certain roles within this organization.

4. Click the **Save** button when done.

7.1.2 Organizations - Permissions

Clicking on **Permissions** (beside **Users** when viewing your organization), allows you to easily manage permissions for this organization.

ORGANIZATIONS / Default / PERMISSIONS					0
Default	DEFAULT ADD USERS / TEA			۵	©
DETAILS USERS PERMISSIONS T	USERS TEAMS			Q KEY	0
USER A admin	NAME *		ORGANIZATION \$		TEAM ROLES
althea	New team		Default	ITEMS 1-1	
	2 Please assign roles to the	selected users/teams		KEY	ITEMS 1-2
ORGANIZATIONS 🛃	New team TEAM	SELECT ROLES Execute Notification Admin		X iAVE	٥
Default	reams	Workflow Admin Credential Admin Job Template Admin		Ē	
2 INVENTORIES 2 F	PROJECTS	Project Admin Auditor JOB TEMPLATES	ADMINS		

Organizations have a unique set of roles not described here. You can assign specific users certain levels of permissions within your organization, or allow them to act as an admin for a particular resource. Refer to *Role-Based Access Controls* for more information.

The **Permissions** tab allows you to review, grant, edit, and remove associated permissions for users as well as team members. To assign permissions to a particular user for this resource:

- 1. Click the **Permissions** tab.
- Click the button to open the Add Users/Teams window.

/ DEMO EXAMPLE / PERMISSIONS						
	DEMO EXAMPLE ADD USER	8				
MPLE	1 Please select Users / Teams from the lists below.					
PERMISSIONS	USERS					
	SEARCH		Q KEY			
	USERNAME [▲]	FIRST NAME	LAST NAME 🗘			
	althea	Althea	Bully			
	austin78	Austin	Texas			
	gdoge	Gerry	Doge			
ES HOSTS	🗆 jdoge	Josie	Doge			
	🗆 jgarcia	Jerry	Garcia			
NAME *	< 1 2 > PAGE 1 OF 2		ITEMS 1 - 5 OF 6			
Database Servers DEMO EXAMPLE			CANCEL SAVE			

- 3. Specify the users or teams that will have access then assign them specific roles:
 - a. Click to select one or multiple checkboxes beside the name(s) of the user(s) or team(s) to select them.

Note: You can select multiple users and teams at the same time by navigating between the **Users** and **Teams** tabs without saving.

After selections are made, the window expands to allow you to select a role from the drop-down menu list for each user or team you chose.

/ DEMO EXAMPLE / P	PERMISSIONS				
MPLE	DEMO EXAMPLE ADD US	0			
PERMISSIONS	USERS TEAMS SEARCH		Q	KEY	
	USERNAME [▲]	FIRST NAME	LAST NAME 🗘		
	🛛 althea	Althea	Bully		
	austin78	Austin	Texas		
	gdoge	Gerry	Doge		
ES HOSTS	□ jdoge	Josie	Doge	_	
	🗆 jgarcia	Jerry	Garcia		
NAME 📍	< 1 2 > PAGE 1 OF 2		п	'EMS 1 - 5 OF 6	
Database Servers	2 Please assign roles to the selected users/teams			KEY	
DEMO EXAMPLE	Althea Bully USER	SELECT ROLES		×	
Demo Inventory		Admin			
King PLC		Update		SAVE	
		Ad Hoc			
		Use			
		Read			

The example above shows options associated with inventories. Different resources have different options available:

- Admin allows read, run, and edit privileges (applies to all resources)
- Use allows use of a resource in a job template (applies all resources except job templates)
- Update allows updating of project via the SCM Update (applies to projects and inventories)
- Ad Hoc allows use of Ad Hoc commands (applies to inventories)
- Execute allows launching of a job template (applies to job templates)
- Read allows view-only access (applies to all resources)

Tip: Use the Key button in the roles selection pane to display a description of each of the roles.

b. Select the role to apply to the selected user or team.

Note:

You can assign roles to multiple users and teams by navigating between the **Users** and **Teams** tabs without saving.

/ DEMO EXAMPLE / PEF	RMISSIONS				
	DEMO EXAMPLE ADD USER	S / TEAMS		0	
MPLE	1 Please select Users / Team	s from the lists below.		_	
PERMISSIONS	USERS				
	SEARCH		Q	KEY	
	USERNAME [▲]	FIRST NAME	LAST NAME	_	
	althea	Althea	Bully		
	austin78	Austin	Texas		
	□ gdoge	Gerry	Doge		
ES HOSTS	☑ jdoge	Josie	Doge		
	🗆 jgarcia	Jerry	Garcia		
NAME 🕈	< 1 2 > PAGE 1 OF 2			TEMS 1 - 5 OF 6	
Database Servers	2 Please assign roles to the	selected users/teams		KEY	
DEMO EXAMPLE	Althea Bully USER	SELECT ROLES		×	
Demo Inventory	Josie Doge USER	SELECT ROLES		×	
King PLC	Production Operatio TEAM	SELECT ROLES		×	
			CANCEL	SAVE	

4. Review your role assignments for each user and team.

/ DEMO EXAMPLE / PE	ERMISSIONS				
	DEMO EXAMPLE ADD USE	RS / TEAMS		8	
MPLE	1 Please select Users / Tear	ns from the lists below.			
PERMISSIONS	USERS				
	SEARCH		Q	KEY	
	USERNAME [▲]	FIRST NAME	LAST NAME 🗢		
	althea	Althea	Bully		
	austin78	Austin	Texas		
	gdoge	Gerry	Doge		
ES HOSTS	🗹 jdoge	Josie	Doge		
	🗆 jgarcia	Jerry	Garcia		
NAME 🔶	< 1 2 > PAGE 1 OF 2		ITEM	IS 1 - 5 OF 6	
Database Servers	2 Please assign roles to the	selected users/teams		KEY	
DEMO EXAMPLE					
Demo Inventory	Althea Bully USER	× Update		×	
King PLC	Josie Doge USER	× Use		×	
	Production Operatio TEAM	× Admin		×	
			CANCEL	SAVE	
			CANCEL	SAVE	

5. Click **Save** when done, and the Add Users/Teams window closes to display the updated roles assigned for each user and team.

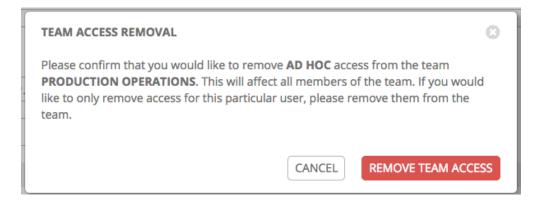
USER *	ROLE	TEAM ROLES
admin	SYSTEM ADMINISTRATOR	
althea	× AD HOC SYSTEM AUDITOR × USE	
jdoge	× UPDATE X USE	
mags3707	SYSTEM ADMINISTRATOR	× AD HOC 앞 × ADMIN 삼 × USE 삼
yser	SYSTEM AUDITOR	

To remove Permissions for a particular user, click the Disassociate (x) button next to its resource.

USER *	ROLE	TEAM ROLES
admin	SYSTEM ADMINISTRATOR	
althea	× AD HOC SYSTEM AUDITOR X USE	\mathbf{i}
jdoge	× UPDATE × USE	
mags3707	SYSTEM ADMINISTRATOR	X AD HOC W X ADMIN W X USE W
yser	SYSTEM AUDITOR	
		ITEMS 1-5

This launches a confirmation dialog, asking you to confirm the disassociation.

TEMS 1-5



7.1.3 Organizations - Notifications

Clicking on **Notifications** (beside **Permissions** when viewing your organization), allows you to easily manage notifications for this organization.

GANIZATIONS / Honey Dog, Inc. / NOT	FICATIONS		
loney Dog, Inc.			G
DETAILS USERS PERMISS	NOTIFICATIONS		
SEARCH	Q	KEY	GO TO NOTIFICATIONS T ADD A NEW TEMPLAT
NAME [•]	TYPE 🗢	SUCCESS	FAILURE
Email notification sample	Email	OFF	OFF
IRC Notification sample	IRC	OFF	OFF
Slack Notification sample	Slack	OFF	OFF
			ITEMS 1

To create a new notification, click the NOTIFICATIONS link from the upper-right side of the notifications list view.

Note: If no notifications have been set up, click the **NOTIFICATIONS** link from above or inside the gray box to add a new notification:

ORGANIZATIONS / Honey Dog, Inc. / NOTIFICATIONS	0
Honey Dog, Inc.	Click to add a ontification of notification GO TO NOTIFICATIONS TO ADD A NEW TEMPLATE
THIS LIST IS POPULATED BY NOTIFICATION TEMPLATES ADDED FROM THE NOTIFICATIONS SECTION	

Supported notification sources include Slack, Email, SMS (via Twilio), HipChat, and more. Refer to *Notifications* for more information.

NEW NOTIFICATION TEMPLATE			e
* NAME	DESCRIPTION	* ORGANIZATION	
		Q	
* TYPE			
Choose a type	•		
		CANCEL	

7.1.4 Organization - Summary

An at-a-glance view of various resources associated with an organization displays at the bottom of each Organization view, called the Organization Summary.

EARCH		c	KEY KEY	
DEFAULT	ø î	HONEY DOG, INC.	2	
Place organization descrip	tion here	A capable company making	g capable things	
5 USERS	2 TEAMS	2 USERS	0 TEAMS	
3 INVENTORIES	2 PROJECTS	0 INVENTORIES	O PROJECTS	
5 JOB TEMPLATES	0 ADMINS	JOB TEMPLATES	ADMINS	

Click on each of the categories to view a list of resources associated with them. Some allow resources to be added, edited, or deleted, such as Users and Admins, while others require editing from another area of the user interface.

From the summary, you can edit the details of an organization (\checkmark) or delete it altogether (\square).

Note: If deleting items that are used by other work items, a message opens listing the items are affected by the deletion and prompts you to confirm the deletion. Some screens will contain items that are invalid or previously deleted, so they will fail to run. Below is an example of such a message:

		💄 admin	đ			(
DJECTS						
PROJECTS 2	DELETE PROJECT FROM GIT Solution Content of the project is currently being used by other resources. Are you sure you want to delete this project?				F	
NAME 🔶	Job Templates 1			AC	TIONS	
O Demo Project	CANCEL	Ø	C	2	Ŵ	
Project from Git	Git b2cflf0 🗋 10/5/2018 3:55:16 PM	ſ	C	2	Ô	
					ITEMS 1	- 2
						ITEMS 1

CHAPTER

EIGHT

USERS

A User is someone who has access to Tower with associated permissions and credentials. Access the Users page by

clicking the Users () icon from the left navigation bar. The Users page allows you to manage all Tower users. The User list may be sorted and searched by Username, First Name, or Last Name and click the headers to toggle your sorting preference.

SEARCH		Q KEY	
USERNAME 🕈	FIRST NAME 🗢	LAST NAME 🗢	ACTION
admin			d ^a
austin78	Austin	Texas	ø 🛍
gdoge	Gerry	Doge	ø û
idoge	Josie	Doge	a 10

8.1 Create a User

To create a new user:



1. Click the **button**, which opens the Create User dialog.

- 2. Enter the appropriate details into the following required fields:
- First Name
- Last Name
- Organization (Choose from an existing organization-this is the default organization if you are using a Self-Supported level license.)
- Email
- Username
- Password
- Confirmation Password
- User Type

Note: When modifying your own password, log out and log back in again in order for it to take effect.

Three types of Tower Users can be assigned:

- **Normal User:** Normal Users have read and write access limited to the resources (such as inventory, projects, and job templates) for which that user has been granted the appropriate roles and privileges.
- System Auditor: Auditors implicitly inherit the read-only capability for all objects within the Tower environment.
- **System Administrator**: A Tower System Administrator (also known as Superuser) has full system administration privileges for Tower with full read and write privileges over the entire Tower installation. A System Administrator is typically responsible for managing all aspects of Tower and delegating responsibilities for day-to-day work to various Users. Assign with caution!

NEW USER			
NEW OSEK			
DETAILS ORGANIZATIONS TEAMS	PERMISSIONS		
* FIRST NAME	* LAST NAME	* ORGANIZATION	
		Q	
EMAIL	* USERNAME	* PASSWORD	
		SHOW	
CONFIRM PASSWORD	USER TYPE		
SHOW	Normal User	*	
	Normal User		
	System Auditor		CANCEL
	System Administrator		

Note: The initial user (usually "admin") created by the Tower installation process is a Superuser. One Superuser must always exist. To delete the "admin" user account, you must first create another Superuser account.

3. Select Save when finished.

Once the user is successfully created, the **User** dialog opens for that newly created User. Note the count for the number of users has also been updated, and a new entry for the new user is added to the list of users below the edit form. The

same window opens whether you click on the user's name, or the Edit () button beside the user. Here, the User's **Organizations**, **Teams** and **Permissions**, as well as other user membership details, may be reviewed and modified.

austin78 ADMIN		c	Э
DETAILS ORGANIZATIONS TEAMS PERMISSIONS			
* FIRST NAME	* LAST NAME	* EMAIL	
Austin	Texas	austin78@mail.com	
* USERNAME	PASSWORD	CONFIRM PASSWORD	
austin78	SHOW	SHOW	
USER TYPE			
System Administrator 🔹			
		CANCEL SAVE	

When you log in as yourself, and view the details of your own user profile, you can manage tokens from your user profile. See *Users - Tokens* for more detail.

austin78 Admin DETAILS ORGANIZATIONS TEAMS PERMISSIONS	TOKENS		8
FIRST NAME	LAST NAME	* EMAIL	
Austin	Texas	austin78@mail.com	
* USERNAME	PASSWORD	CONFIRM PASSWORD	
austin78	SHOW	SHOW	
USER TYPE			
System Administrator			
		CANCEL SAVE	E

8.2 User Types - Quick View

Once a user has been created, you can easily view permissions and user type information by looking beside their user name in the User overview screen.

ι	USERS / jdoge			•	
	View user labels he	re for			
	jdoge AUDITOR Auditor, Admin, LDA	NP, etc.		0	
	DETAILS ORGANIZATIONS TEAMS PERMISSIONS				
	* FIRST NAME	* LAST NAME	* EMAIL		
	Josie	Doge	jdoge@mail.com		

If the user account is associated with an enterprise-level authentication method (such as SAML, RADIUS, or LDAP), the user type may look like:

USERS / jdoge			•
jdoge RADIUS			
DETAILS ORGANIZATIONS TEAMS PERMISSIONS			
* FIRST NAME Josie	* LAST NAME Doge	* EMAIL jdoge@mail.com	

If the user account is associated with a social authentication method, the user type will look like:

USERS / jdoge			•
jdoge social			8
DETAILS ORGANIZATIONS TEAMS PERMISSIONS * FIRST NAME * <td>* LAST NAME</td> <td>* EMAIL</td> <td></td>	* LAST NAME	* EMAIL	
Josie	Doge	jdoge@mail.com	

8.3 Users - Organizations

This displays the list of organizations of which that user is a member. This list may be searched by Organization Name or Description. Organization membership cannot be modified from this display panel.

jdoge Auditor	0
DETAILS ORGANIZATIONS TEAMS PERMISSIONS	
SEARCH Q KEY	
NAME [▲]	DESCRIPTION \$
Honey Dog, Inc.	A capable company making capable things
	ITEMS 1 - 1

8.4 Users - Teams

This displays the list of teams of which that user is a member. This list may be searched by **Team Name** or **Description**. Team membership cannot be modified from this display panel. For more information, refer to *Teams*.

Until a Team has been created and the user has been assigned to that team, the assigned Teams Details for the User appears blank.

jdoge Auditor	Θ
DETAILS ORGANIZATIONS TEAMS PERMISSIONS	
THIS USER IS NOT A MEMBER OF ANY TEAMS	

8.5 Users - Permissions

The set of Permissions assigned to this user (role-based access controls) that provide the ability to read, modify, and administer projects, inventories, job templates, and other Tower elements are Privileges.

Note: It is important to note that the job template administrator may not have access to any inventory, project, or credentials associated with the template. Without access to these, certain fields in the job template aren't editable.

This screen displays a list of the roles that are currently assigned to the selected User and can be sorted and searched by **Name**, **Type**, or **Role**.

idoge			0
DETAILS ORGANIZATIONS	TEAMS		
SEARCH		Q KEY	•
NAME	ТҮРЕ	ROLE	ACTIONS
Default	Organization	Auditor	×
Honey Dog, Inc.	Organization	Member	×
			ITEMS 1 - 2

8.5.1 Add Permissions

To add permissions to a particular user:

+

1. Click the

button, which opens the Add Permissions Wizard.

USERS / jdoge / PERMISSIO	NS		٢
	JDOGE ADD PERMISSIONS	Θ	
jdoge	1 Please select resources from the lists below.		
DETAILS	JOB TEMPLATES WORKFLOW TEMPLATES PROJECTS INVENTORIES CREDENTIALS ORGANIZATIO	ONS	
SEARCH	SEARCH	KEY	
NAME	NAME *		ACTIONS
Default	Basic Job Template		×
Honey Dog, Inc.	Demo Job Template		×
		ITEMS 1-2	ITEMS 1-2
	2 Please assign roles to the selected resources		
USERS 4	ORGANIZATIONS		
SEARCH			
USERNAME 🔶		KEY	ACTIONS
admin	CANCEL		1
austin78	Austin Texas		1

- 2. Click to select the Tower object for which the user will have access:
- Job Templates. This is the default tab displayed in the Add Permissions Wizard.
- Workflow Templates
- Projects
- Inventories
- Credentials
- Organizations

Note: You can assign different roles to different resources all at once to avoid having to click the button. To do so, simply go from one tab to another after making your selections without saving.

8.5. Users - Permissions

- 3. Perform the following steps to assign the user specific roles for each type of resource:
 - a. In the desired tab, click the checkbox beside the name of the resource to select it.

The dialog expands to allow you to select the role for the resource you chose.

b. Select the role from the drop-down menu list provided. Only some roles are applicable to certain resources.

USERS / jdoge / PERMISSION	NS	۵
jdoge	JOGE ADD PERMISSIONS	
	1 Please select resources from the lists below.	
DETAILS ORGANIZAT	JOB TEMPLATES WORKFLOW TEMPLATES PROJECTS INVENTORIES CREDENTIALS ORGANIZATIONS	_
SEARCH	SEARCH Q KEY	+
NAME	NAME ^	ACTIONS
Default	Basic Job Template	×
Honey Dog, Inc.	Demo Job Template	х
	тем5 1-2	ITEMS 1-2
USERS (4)	Please assign roles to the selected resources	
SEARCH	JOB TEMPLATES ORGANIZATIONS	•
USERNAME [▲]	Select a role KEY	ACTIONS
admin	NAME ACTIONS	I
austin78	Basic Job Template	1
gdoge	CANCEL SAVE	e 🗇 🛱
jdoge	josie Uoge	2

Tip: Use the Key button to display the help text for each of the roles applicable to the resource selected.

c. Review your role assignments for each of the Tower objects by clicking on their respective buttons in the expanded section 2 of the Add Permissions Wizard.

USERS / jdoge / PERMISSIO	ONS				Q
jdoge	JDOGE ADD PERMISSIONS			0	
Jooge	1 Please select resources from the lists below	v.		_	
DETAILS	JOB TEMPLATES WORKFLOW TEMPLATES	PROJECTS	CREDENTIALS	NS	
SEARCH	SEARCH		Q	KEY	
NAME	NAME A	ORGAN	ization ≑	- 1	ACTIONS
Default	Demo Inventory	Default		- 1	×
Honey Dog, Inc.				_	×
			п	'EMS 1 - 1	
	2 Please assign roles to the selected resource	es		- 1	ITEMS 1-2
USERS (4)	JOB TEMPLATES PROJECTS INVENTORIES			- 1	
SEARCH	Use		•	KEY	
USERNAME 🔶	NAME	ORGANIZATION	ACT	IONS	ACTIONS
admin	Demo Inventory	Default		0	1
austin78			CANCEL	SAVE	P 🗊
gdoge	Geny		Doge		/ 🖻

d. Click **Save** when done, and the Add Permissions Wizard closes to display the updated profile for the user with the roles assigned for each selected resource.

oge			
DETAILS ORGANIZATIONS TE	AMS		
SEARCH		QKEY	l
NAME	TYPE	ROLE	ACTION
Default	Organization	Auditor	\$
Demo Inventory	Inventory	Use	3
Honey Dog, Inc.	Organization	Member	3
Sample Project	Project	Admin	3
Basic Job Template	Job Template	Admin	2

To remove Permissions for a particular User, click the Disassociate (X) button under Actions. This launches a **Remove Role** dialog, asking you to confirm the disassociation.

Note: You can also add teams or individual users and assign them permissions at the object level (projects, inventories, job templates, and workflow templates) as well. Ansible Tower release 3.1 introduces the ability to batch assign permissions. This feature reduces the time for an organization to onboard many users at one time. For more details, refer to their respective chapters in the *Ansible Tower User Guide v3.4.4*.

8.6 Users - Tokens

Before you add a token for your user, you may want to create an application if you want to associate your token to it. You may also create a personal access token (PAT) without associating it with any application. To create a token for your user:

- 1. If not already selected, click on your user from the Users list view to configure your OAuth 2 tokens.
- 2. Click the Tokens tab from your user's profile.

When no tokens are present, the Tokens screen prompts you to add them:

admin admin	0
DETAILS ORGANIZATIONS TEAMS PERMISSIONS TOKENS	
SEARCH Q KEY	+
PLEASE ADD ITEMS TO THIS LIST.	



button, which opens the Create Token window.

- 4. Enter the following details in Create Token window:
- Application: enter the name of the application with which you want to associate your token. Alternatively, you

can search for it by clicking the Q button. This opens a separate window that allows you to choose from the

available options. Use the Search bar to filter by name if the list is extensive. Leave this field blank if you want to create a Personal Access Token (PAT) that is not linked to any application.

- Description: optionally provide a short description for your token.
- Scope (required): specify the level of access you want this token to have.
- 5. When done, click Save or Cancel to abandon your changes.

After the token is saved, the newly created token for the user displays with the token information and when it expires.

	TOKEN INFORMAT	ION	8
EA	TOKEN REFRESH TOKEN EXPIRES	ufCk6HsQB5b89ALtXQHDYQbreR2BDt CeBZ5MUOj3AzgbBDVAF0TmsHgHLYh2 11/25/3017 5:27:34 PM	
ł			ОК

Note: This is the only time the token value and associated refresh token value will ever be shown.

In the user's profile, the application for which it is assigned to and its expiration displays in the token list view.

min ADMIN		
DETAILS ORGANIZATIONS TEAMS	PERMISSIONS TOKENS	
EARCH	Q KEY	
my creds app Token		
APPLICATION my creds app		t
EXPIRATION 11/25/3017 11:00:22 AM		

CHAPTER

TEAMS

A Team is a subdivision of an organization with associated users, projects, credentials, and permissions. Teams provide a means to implement role-based access control schemes and delegate responsibilities across organizations. For instance, permissions may be granted to a whole Team rather than each user on the Team.

You can create as many Teams of users as make sense for your Organization. Each Team can be assigned permissions, just as with Users. Teams can also scalably assign ownership for Credentials, preventing multiple Tower interface click-throughs to assign the same Credentials to the same user.



Access the Teams page by clicking the Teams (**I**) icon from the left navigation bar. The Teams page allows you to manage the teams for Tower. The team list may be sorted and searched by **Name** or **Organization**.

EARCH	Q KEY	
NAME [▲]	ORGANIZATION 🗢	ACTIO
Engineering	Honey Dog, Inc.	di ^a
т	Honey Dog, Inc.	di s
Sales and Marketing	Honey Dog, Inc.	Ø
Services and Support	Honey Dog, Inc.	de la

9.1 Create a Team

DESCRIPTION	* ORGANIZATION
	Q
	C
	DESCRIPTION

- 2. Enter the appropriate details into the following fields:
- Name
- Description (optional)
- Organization (Choose from an existing organization)
- 3. Click Save.

Once the Team is successfully created, Tower opens the Details dialog, which also allows you to review and edit your

Team information. This is the same menu that is opened if the Edit () button is clicked from the **Teams** link. You can also review **Users** and **Permissions** associated with this Team.

Production Operations				0
DETAILS USERS PERMISSIONS * NAME Production Operations	DESCRIPTION ProOps team	* ORGANIZATION Q Honey Dog, Inc.	CANCEL	SAVE
TEAMS S SEARCH	Q KEY			•
NAME *	ORGANIZATION 💠		AC	TIONS
Engineering	Honey Dog, Inc.		dit.	Û
іт -	Honey Dog, Inc.		dit	卣
Production Operations	Honey Dog, Inc.		ø	Û
Sales and Marketing	Honey Dog, Inc.		ß	Ē
Services and Support	Honey Dog, Inc.		Ø	Û
				ITEMS 1-5

9.1.1 Teams - Users

This tab displays the list of Users that are members of this Team. This list may be searched by Username, First Name, or Last Name. For more information, refer to *Users*.

Production Operations	0
DETAILS USERS PERMISSIONS	
SEARCH Q K	+
USER A	ROLE
admin	SYSTEM ADMINISTRATOR
austin78	SYSTEM ADMINISTRATOR
gdoge	SYSTEM AUDITOR
jdoge	× MEMBER
	ITEMS 1-4

Add a User

In order to add a user to a team, the user must already be created in Tower. Refer to *Create a User* to create a user. To add existing users to the Team:



- button.
- 2. Select one or more users from the list of available users by clicking the checkbox next to the user(s). Doing so expands the lower part of the Wizard to assign roles to each user.

TEAMS / Production Operations /	USERS			٠
Production Operations	PRODUCTION OPERATIONS ADD USERS 1 Please select Users from the list below.		©	
DETAILS USERS SEARCH	SEARCH	FIRST NAME	Q KEY	
USER *	admin			
austin78	austin78gdoge	Austin Gerry	Texas	
gdoge	jdoge	Josie	Doge	ITEMS 1-3
TEAMS SEARCH	Please assign roles to the selected users/ Josie Doge user SELECT ROLE		KEY	
NAME *	noney or	υ <u>β</u> , IIIC.	CANCEL	ACTIONS

3. For each user, click from the drop-down menu to select one or more roles for that user.

Note: For help on what the roles mean, click the **Key** button. For more information, refer to the *Roles* section of this guide.

TEAMS / Production Operations /	USERS			۵
Production Operations	PRODUCTION OPERATIONS ADD I		٢	
DETAILS USERS SEARCH	SEARCH	FIRST NAME	Q KEY	
USER •	admin austin78	Austin	Texas	
austin78	□ gdoge	Gerry	Doge	
gdoge	🛛 jdoge	Josie	Doge	ITEMS 1-3
TEAMS 5	2 Please assign roles to the selected	i users/teams	KEY	
SEARCH	,		CANCEL	ACTIONS
Engineering		ioney bog, me.		I DE CARACTERISTE DE CARACTERI

In this example, two users have been selected and each have been granted certain roles within this team.

4. Click the **Save** button when done.

9.1.2 Teams - Permissions

Selecting the **Permissions** view displays a list of the permissions that are currently available for this Team. The permissions list may be sorted and searched by **Name**, **Inventory**, **Project** or **Permission** type.

oduction Operations			
DETAILS USERS PERMISS	SIONS		
SEARCH		Q KEY	
NAME	TYPE	ROLE	ACTIONS
Demo Project	Project	Use	х
Demo Job Template	Job Template	Execute	×
King PLC	Inventory	Ad Hoc	×

The set of privileges assigned to Teams that provide the ability to read, modify, and administer projects, inventories, and other Tower elements are permissions. By default, the Team is given the "read" permission (also called a role).

Permissions must be set explicitly via an Inventory, Project, Job Template, or within the Organization view.

Add Team Permissions

To add permissions to a Team:



button, which opens the Add Permissions Wizard.

TEAMS / Production Operatio	ons / PERMISSIONS	(
	PRODUCTION OPERATIONS ADD PERMISSIONS	
Production Operations	1 Please select resources from the lists below.	
DETAILS USERS	JOB TEMPLATES WORKFLOW TEMPLATES PROJECTS INVENTORIES ORGANIZATIONS	
	SEARCH Q KEY	+
	NAME A	
	Basic Job Template	
	Demo Job Template	
	ITEMS 1-2	
	CANCEL SAVE	

- 2. Click to select the Tower object for which the user will have access:
- Job Templates. This is the default tab displayed in the Add Permissions Wizard.
- Workflow Templates
- Projects
- Inventories
- Credentials

+

• Organizations

Note: You can assign different roles to different resources all at once to avoid having to click the **button**. To do so, simply go from one tab to another after making your selections without saving.

- 3. Perform the following steps to assign the user specific roles for each type of resource:
 - a. In the desired tab, click the checkbox beside the name of the resource to select it.

The dialog expands to allow you to select the role for the resource you chose.

b. Select the role from the drop-down menu list provided. Only some roles are applicable to certain resources.

TEAMS / Production Operatio	ons / PERMISSIONS		۵
Production Operations	PRODUCTION OPERATIONS ADD PERMISSIONS 1 Please select resources from the lists below.	0	
DETAILS USERS	JOB TEMPLATES WORKFLOW TEMPLATES PROJECTS INVENTORIES CREDENTIALS	Q KEY	
	NAME Basic Job Template Demo Job Template	_	
	2 Please assign roles to the selected resources	ITEMS 1-2	
TEAMS S	JOB TEMPLATES Select a role	KEY	
SEARCH	NAME Demo Job Template	ACTIONS	ACTIONS
Engineering		CANCEL SAVE	/ ¹

Tip: Use the Key button to display the help text for each of the roles applicable to the resource selected.

c. Review your role assignments for each of the Tower objects by clicking on their respective buttons in the expanded section 2 of the Add Permissions Wizard.

TEAMS / Production Operation	ons / PERMISSIONS				0
Production Operations	PRODUCTION OPERATIONS ADD PERMISSIO			0	
DETAILS USERS		PROJECTS	CREDENTIALS	ORGANIZATIONS	
	SEARCH			Q KEY	
	NAME 🔶	ORG	GANIZATION 🗘		
	Demo Inventory	Defa	ault		
	King PLC	Defa	ault		
				ITEMS 1 - 2	
	2 Please assign roles to the selected resources	3			
TEAMS 5	JOB TEMPLATES PROJECTS INVENTORIES				
TEAMS	Ad Hoc			* KEY	
SEARCH					•
NAME A	NAME	ORGANIZATION		ACTIONS	ACTIONS
Engineering	King PLC	Default		8	
IT			(CANCEL SAVE	/ 10 / 10

d. Click **Save** when done, and the Add Permissions Wizard closes to display the updated profile for the user with the roles assigned for each selected resource.

TEAMS / Production Operations / PERMISSIONS				0
Production Operations				8
DETAILS USERS PERMISSIONS				
SEARCH		Q KEY		•
NAME	TYPE		ROLE	ACTIONS
Demo Project	Project		Use	×
Demo Job Template	Job Template		Execute	ж
King PLC	Inventory		Ad Hoc	×
				ITEMS 1-3

To remove Permissions for a particular User, click the Disassociate (X) button under Actions. This launches a **Remove Role** dialog, asking you to confirm the disassociation.

Note: You can also add teams or individual users and assign them permissions at the object level (projects, inventories, job templates, and workflow templates) as well. Ansible Tower release 3.1 introduces the ability to batch assign permissions. This feature reduces the time for an organization to onboard many users at one time. For more details, refer to their respective chapters in the *Ansible Tower User Guide v3.4.4*.

CHAPTER

CREDENTIALS

Credentials are utilized by Tower for authentication when launching Jobs against machines, synchronizing with inventory sources, and importing project content from a version control system.

You can grant users and teams the ability to use these credentials, without actually exposing the credential to the user. If you have a user move to a different team or leave the organization, you don't have to re-key all of your systems just because that credential was available in Tower.

Note: Tower encrypts passwords and key information in the Tower database and never makes secret information visible via the API.

10.1 Understanding How Credentials Work

Ansible Tower uses SSH to connect to remote hosts (or the Windows equivalent). In order to pass the key from Tower to SSH, the key must be decrypted before it can be written a named pipe. Tower then uses that pipe to send the key to SSH (so that it is never written to disk).

If passwords are used, Ansible Tower handles those by responding directly to the password prompt and decrypting the password before writing it to the prompt.

The encryption/decryption algorithm uses a variation of Fernet: a symmetric encryption cipher utilizing AES-256 in CBC mode alongside a SHA-256 HMAC. The key is derived from the SECRET_KEY (found in the awx settings). Specific, sensitive, Model fields in Tower are encrypted and include:

Credential: password, ssh_key_data, ssh_key_unlock, become_password, vault_password UnifiedJob: start_args

Data is encrypted before it is saved to the database and is decrypted as is needed in Tower. The encryption/decryption process derives the AES-256 bit encryption key from <SECRET_KEY, field_name, primary_key> where field_name is the name of the Model field and primary_key is the database assigned auto-incremented record ID. Thus, if any attribute used in the key generation process changes, Tower fails to correctly decrypt the secret.

Note: The rules of encryption and decryption for Ansible Tower also apply to one field outside of credentials, the Unified Job start_args field, which is used through the job, ad_hoc_command, and system_job data types.

10.2 Getting Started with Credentials



Access the Credentials page by clicking the Credentials () icon from the left navigation bar. The Credentials page displays a search-able list of all available Credentials and can be sorted by **Name**.

A	TOWER			🛔 admin	0		ტ
≡	CREDENTIALS						<
26	CREDENTIALS 1						
ं	SEARCH	QKEY					+
<u> </u>	SEARCH	Q KET					
	NAME *	KIND	OWNERS			ACTION	IS
Ø	Demo Credential	Machine	admin		Can's	<i>6</i>	r
0,						ITEMS	1-1

Credentials added to a Team are made available to all members of the Team, whereas credentials added to a User are only available to that specific User by default.

Note: If deleting items that are used by other work items, a message opens listing the items are affected by the deletion and prompts you to confirm the deletion. Some screens will contain items that are invalid or previously deleted, so they will fail to run. Below is an example of such a message:

		💄 admin	0		
PROJECTS					Ø
PROJECTS 2 SEARCH	DELETE PROJECT FROM GIT The project is currently being used by other resources. Are you sure you want to delete this project? Job Templates 1				+ FIONS
O Demo Project	CANCEL DELETE	ſ	C	ACT د	۳ONS ش
Project from Git	Git b2cflf0 🗋 10/5/2018 3:55:16 PM	I	ζ	4	面 ITEMS 1 - 2

To help you get started, a Demo Credential has been created for your use.

Clicking on the link for the Demo Credential takes you to the Details view of this Credential.

Demo Credential		0
DETAILS PERMISSIONS		
* NAME Ø	DESCRIPTION 🔞	ORGANIZATION
Demo Credential		Q SELECT AN ORGANIZATION
* CREDENTIAL TYPE @		
Q Machine		
TYPE DETAILS		
USERNAME	PASSWORD	
admin	SHOW	
SSH PRIVATE KEY HINT: Drag and drop an SSH private key file on the field below.		
PRIVATE KEY PASSPHRASE Prompt on launch	PRIVILEGE ESCALATION METHOD	PRIVILEGE ESCALATION USERNAME
SHOW	×	
PRIVILEGE ESCALATION PASSWORD Prompt on launch		
SHOW		
·		
		CANCEL SAVE

Clicking on **Permissions** shows you users and teams associated with this Credential and their granted roles (owner, admin, auditor, etc.)

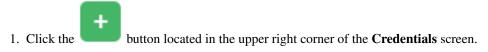
C	REDENTIALS / EDIT CREDENTIAL / PERMISSIONS			0
	CREDENTIALS PERMISSIONS DETAILS PERMISSIONS			٥
	SEARCH	Q KEY		
	USER 🔶	ROLE	TEAM ROLES	
	admin			
				ITEMS 1-1



button to assign this **Demo Credential** to additional Users or Teams.

10.3 Add a New Credential

To create a new credential:



CREDENTIALS / CREATE CREDENTIAL		6
		0
DETAILS PERMISSIONS	DESCRIPTION @	
* CREDENTIAL TYPE 🔞		Q SELECT AN ORGANIZATION
Q SELECT A CREDENTIAL TYPE		CANCEL SAVE

- 2. Enter the name for your new credential in the Name field.
- 3. Optionally enter or select the name of the organization with which the credential is associated.
- 4. Enter or select the credential type you want to create.

SEARCH	Q KEY
NAME 📤	
 Amazon Web Services 	
 Ansible Tower 	
 Google Compute Engine 	
 Insights 	
O Machine	
< 1 2 3 > PAGE 1 OF 3	ITEMS 1 - 5 OF 14
	CANCEL SELECT

- 5. Enter the appropriate details depending on the type of credential selected, as described in the following sections.
- 6. Click Save when done.

10.4 Credential Types

The following credential types are supported with Ansible Tower 3.4.4:

- Amazon Web Services
- Ansible Tower
- Google Compute Engine
- Insights
- Machine
- Microsoft Azure Resource Manager
- Network
- OpenStack
- Red Hat CloudForms
- Red Hat Satellite 6
- Red Hat Virtualization
- Source Control
- Vault
- VMware vCenter

10.4.1 Amazon Web Services

Selecting this credential type enables synchronization of cloud inventory with Amazon Web Services.

Tower uses the following environment variables for AWS credentials and are fields prompted in the user interface:

AWS_ACCESS_KEY_ID AWS_SECRET_ACCESS_KEY AWS_SECURITY_TOKEN		
NEW CREDENTIAL DETAILS PERMISSIONS		0
* NAME 🚱	DESCRIPTION @	ORGANIZATION
		Q SELECT AN ORGANIZATION
* ACCESS KEY	* SECRET KEY	STS TOKEN 🔞
	SHOW	SHOW
		CANCEL SAVE

Traditional Amazon Web Services credentials consist of the AWS Access Key and Secret Key.

Ansible Tower version 2.4.0 introduced support for EC2 STS tokens (sometimes referred to as IAM STS credentials). Security Token Service (STS) is a web service that enables you to request temporary, limited-privilege credentials

for AWS Identity and Access Management (IAM) users. To learn more about the IAM/EC2 STS Token, refer to: http://docs.aws.amazon.com/IAM/latest/UserGuide/id_credentials_temp.html

Note: If the value of your tags in EC2 contain booleans (yes/no/true/false), you must remember to quote them.

Warning: To use implicit IAM role credentials, do not attach AWS cloud credentials in Tower when relying on IAM roles to access the AWS API. While it may seem to make sense to attach your AWS cloud credential to your job template, doing so will force the use of your AWS credentials and will not "fall through" to use your IAM role credentials (this is due to the use of the boto library.)

10.4.2 Ansible Tower

Selecting this credential allows you to access another Tower instance.

NEW CREDENTIAL		0
DETAILS PERMISSIONS		
* NAME @	DESCRIPTION @	ORGANIZATION 😧
Ansible Tower Credential		Q Default
CREDENTIAL TYPE		
Q Ansible Tower		
TYPE DETAILS		
* ANSIBLE TOWER HOSTNAME	* USERNAME	* PASSWORD
		SHOW
		CANCEL

Ansible Tower credentials have the following inputs that are required:

- Ansible Tower Hostname: The base URL or IP address of the other Tower instance to connect to.
- Username: The username to use to connect to it.
- Password: The password to use to connect to it.

10.4.3 Google Compute Engine

Selecting this credential type enables synchronization of cloud inventory with Google Compute Engine (GCE).

Tower uses the following environment variables for GCE credentials and are fields prompted in the user interface:

```
GCE_EMAIL
GCE_PROJECT
GCE_CREDENTIALS_FILE_PATH
```

NEW CREDENTIAL				8
DETAILS PERMISSIONS				
* NAME 🔞	DESCRIPTION 😨	ORC	GANIZATION 🔞	
New Credential		٩	SELECT AN ORGANIZATION	
* CREDENTIAL TYPE 🔞				
Q Google Compute Engine				
TYPE DETAILS				
* SERVICE ACCOUNT EMAIL ADDRESS	PROJECT	SER	VICE ACCOUNT JSON FILE 🕜	
			CHOOSE A FILE	
* RSA PRIVATE KEY 🕖 HINT: Drag and drop an SSH private key file on the	field below			
• KSA PRIVATE KET V HINT: Drag and drop an SSH private key file on the	nela below.			
				CANCEL SAVE

GCE credentials have the following inputs that are required:

- Service Account Email Address: The email address assigned to the Google Compute Engine service account.
- **Project**: Optionally provide the GCE assigned identification or the unique project ID you provided at project creation time.
- Service Account JSON File: Optionally upload a GCE service account file. Use the folder () icon to browse for the file that contains the special account information that can be used by services and applications running on your GCE instance to interact with other Google Cloud Platform APIs. This grants permissions to the service account and virtual machine instances.
- RSA Private Key: The PEM file associated with the service account email.

10.4.4 Insights

Selecting this credential type enables synchronization of cloud inventory with Red Hat Insights.

NEW CREDENTIAL DETAILS PERMISSIONS			8
NAME New credential	DESCRIPTION @	ORGANIZATION Image: Content of the second seco	
USERNAME	PASSWORD SHOW		
		CANCEL	

Insights credentials consist of the Insights Username and Password, which is the user's Red Hat Customer Portal Account username and password.

10.4.5 Machine

Machine credentials enable Tower to invoke Ansible on hosts under your management. Just like using Ansible on the command line, you can specify the SSH username, optionally provide a password, an SSH key, a key password, or even have Tower prompt the user for their password at deployment time. They define ssh and user-level privilege escalation access for playbooks, and are used when submitting jobs to run playbooks on a remote host. Network connections (httpapi, netconf, and network_cli) use **Machine** for the credential type.

Machine/SSH credentials do not use environment variables. Instead, they pass the username via the ansible -u flag, and interactively write the SSH password when the underlying SSH client prompts for it.

		٢
PERMISSIONS NAME New Credential	DESCRIPTION @	ORGANIZATION Ø Q Default
CREDENTIAL TYPE A Machine		
TYPE DETAILS USERNAME SSH PRIVATE KEY HINT: Drag and drop an SSH private key file on the field below.	PASSWORD Prompt on launch SHOW	
PRIVATE KEY PASSPHRASE Prompt on launch SHOW		PRIVILEGE ESCALATION USERNAME
PRIVILEGE ESCALATION PASSWORD Prompt on launch SHOW		
		CANCEL

Machine credentials have several attributes that may be configured:

- Username: The username to be used for SSH authentication.
- **Password**: The actual password to be used for SSH authentication. This password will be stored encrypted in the Tower database, if entered. Alternatively, you can configure Tower to ask the user for the password at launch time by selecting **Prompt on launch**. In these cases, a dialog opens when the job is launched, promoting the user to enter the password and password confirmation.
- SSH Private Key: Copy or drag-and-drop the SSH private key for the machine credential.
- **Private Key Passphrase**: If the SSH Private Key used is protected by a password, you can configure a Key Password for the private key. This password will be stored encrypted in the Tower database, if entered. Alternatively, you can configure Tower to ask the user for the password at launch time by selecting **Prompt on launch**. In these cases, a dialog opens when the job is launched, prompting the user to enter the password and password confirmation.
- **Privilege Escalation Method**: Specifies the type of escalation privilege to assign to specific users. This is equivalent to specifying the --become-method=BECOME_METHOD parameter, where BECOME_METHOD could be sudo | su | pbrun | pfexec | dzdo | pmrun.
- empty selection: Assigns no privilege escalation to this credential.
- sudo: Performs single commands with super user (root user) privileges
- su: Switches to the super user (root user) account (or to other user accounts)

- **pbrun**: Requests that an application or command be run in a controlled account and provides for advanced root privilege delegation and keylogging.
- pfexec: Executes commands with predefined process attributes, such as specific user or group IDs.
- **dzdo**: An enhanced version of sudo that uses RBAC information in an Centrify's Active Directory service (see Centrify's site on DZDO)
- pmrun: Requests that an application is run in a controlled account (refer to Privilege Manager for Unix 6.0)
- runas: Allows you to run as current user.

PRIVATE KEY PASSPHRASE	Prompt on launch	PRIVILEGE ESCALATION METHOD 🔞		PRIVILEGE ESCALATION USERNAME
SHOW PRIVILEGE ESCALATION PASSWORD SHOW	Prompt on launch	v sudo su pbrun pfexec dzdo pmrun runas]	CANCEL SAVE

- **Privilege Escalation Username** field is only seen if an option for privilege escalation is selected. Enter the username to use with escalation privileges on the remote system.
- **Privilege Escalation Password**: field is only seen if an option for privilege escalation is selected. Enter the actual password to be used to authenticate the user via the selected privilege escalation type on the remote system. This password will be stored encrypted in the Tower database, if entered. Alternatively, you may configure Tower to ask the user for the password at launch time by selecting **Prompt on launch**. In these cases, a dialog opens when the job is launched, promoting the user to enter the password and password confirmation.

Note: Sudo Password must be used in combination with SSH passwords or SSH Private Keys, since Tower must first establish an authenticated SSH connection with the host prior to invoking sudo to change to the sudo user.

Warning:	Credentials which are used	n Scheduled Jobs must not b	be configured as "Prom	pt on launch".
----------	----------------------------	-----------------------------	------------------------	----------------

10.4.6 Microsoft Azure Resource Manager

Selecting this credential type enables synchronization of cloud inventory with Microsoft Azure Resource Manager.

NEW CREDENTIAL		8
DETAILS PERMISSIONS		
* NAME 😧	DESCRIPTION 🙆	ORGANIZATION 😧
CREDENTIAL TYPE C Microsoft Azure Resource Manager TYPE DETAILS		
* SUBSCRIPTION ID	USERNAME	PASSWORD SHOW
CLIENT ID	CLIENT SECRET	TENANT ID
AZURE CLOUD ENVIRONMENT		
		CANCEL SAVE

Microsoft Azure Resource Manager credentials have several attributes that may be configured:

- Subscription ID: The Subscription UUID for the Microsoft Azure account (required).
- Username: The username to use to connect to the Microsoft Azure account.
- Password: The password to use to connect to the Microsoft Azure account.
- Client ID: The Client ID for the Microsoft Azure account.
- Client Secret: The Client Secret for the Microsoft Azure account.
- Tenant ID: The Tenant ID for the Microsoft Azure account.
- Azure Cloud Environment: The variable associated with Azure cloud or Azure stack environments.

These fields are equivalent to the variables in the API. To pass service principal credentials, define the following variables:

AZURE_CLIENT_ID AZURE_SECRET AZURE_SUBSCRIPTION_ID AZURE_TENANT AZURE_CLOUD_ENVIRONMENT

To pass an Active Directory username/password pair, define the following variables:

```
AZURE_AD_USER
AZURE_PASSWORD
AZURE_SUBSCRIPTION_ID
```

You can also pass credentials as parameters to a task within a playbook. The order of precedence is parameters, then environment variables, and finally a file found in your home directory.

To pass credentials as parameters to a task, use the following parameters for service principal credentials:

```
client_id
secret
subscription_id
tenant
azure_cloud_environment
```

Or, pass the following parameters for Active Directory username/password:

ad_user password subscription_id

10.4.7 Network

Select the Network credential type **only** if you are using a *local* connection with *provider* to use Ansible networking modules to connect to and manage networking devices. When connecting to network devices, the credential type must match the connection type:

- For local connections using provider, credential type should be Network
- For all other network connections (httpapi, netconf, and network_cli), credential type should be Machine

For an overview of connection types available for network devices, refer to Multiple Communication Protocols.

Tower uses the following environment variables for Network credentials and are fields prompted in the user interface:

ANSIBLE_NET_USERNAME		
NEW CREDENTIAL DETAILS PERMISSIONS		0
*NAME 🚱	DESCRIPTION 🕖	ORGANIZATION
New Credential		Q Default
CREDENTIAL TYPE Q Network TYPE DETAILS		
* USERNAME	PASSWORD	
	SHOW	
SSH PRIVATE KEY HINT: Drag and drop an SSH private key file on the field below.		
PRIVATE KEY PASSPHRASE	OPTIONS	AUTHORIZE PASSWORD
SHOW	Authorize	SHOW
		CANCEL

Network credentials have several attributes that may be configured:

- Username: The username to use in conjunction with the network device (required).
- Password: The password to use in conjunction with the network device.
- **SSH Private Key**: Copy or drag-and-drop the actual SSH Private Key to be used to authenticate the user to the network via SSH.
- **Private Key Passphrase**: The actual passphrase for the private key to be used to authenticate the user to the network via SSH.
- Authorize: Select this from the Options field to control whether or not to enter privileged mode.
- If Authorize is checked, enter a password in the Authorize Password field to access privileged mode.

For more information, refer to the *Inside Playbook* blog, Porting Ansible Network Playbooks with New Connection Plugins.

10.4.8 OpenStack

Selecting this credential type enables synchronization of cloud inventory with OpenStack.

NEW CREDENTIAL		0
DETAILS PERMISSIONS		
NAME 🔞	DESCRIPTION @	ORGANIZATION @ Q Default
CREDENTIAL TYPE Q OpenStack TYPE DETAILS		
• USERNAME	* PASSWORD (API KEY) SHOW	
+ PROJECT (TENANT NAME)		
		CANCEL

OpenStack credentials have the following inputs that are required:

- Username: The username to use to connect to OpenStack.
- Password (API Key): The password or API key to use to connect to OpenStack.
- Host (Authentication URL): The host to be used for authentication.
- **Project** (**Tenant Name**): The Tenant name or Tenant ID used for OpenStack. This value is usually the same as the username.
- Domain name: Optionally provide the FQDN to be used to connect to OpenStack.

If you are interested in using OpenStack Cloud Credentials, refer to *Utilizing Cloud Credentials* in this guide for more information, including a sample playbook.

10.4.9 Red Hat CloudForms

Selecting this credential type enables synchronization of cloud inventory with Red Hat CloudForms.

Tower writes a CloudForms configuration file based on fields prompted in the user interface. The absolute path to the file is set in the following environment variable:

LOUDFORMS_INI_PATH			
NEW CREDENTIAL			0
DETAILS PERMISSIONS			
* NAME 🔞	DESCRIPTION 🔞	ORGANIZATION	
New Credential		Q. Default	
* CREDENTIAL TYPE 🔞			
Q Red Hat CloudForms			
TYPE DETAILS			
* CLOUDFORMS URL	* USERNAME	* PASSWORD	
		SHOW	
			CANCEL

CloudForms credentials have the following inputs that are required:

• CloudForms URL: The CloudForms URL or IP address to connect to.

- Username: The username to use to connect to CloudForms.
- Password: The password to use to connect to CloudForms.

Additional Resources:

Refer to Red Hat's blog post series on Ansible Tower Integration in Red Hat CloudForms 4.1 at http://cloudformsblog. redhat.com/2016/07/22/ansible-tower-in-cloudforms/.

10.4.10 Red Hat Satellite 6

Selecting this credential type enables synchronization of cloud inventory with Red Hat Satellite 6.

Tower writes a Satellite configuration file based on fields prompted in the user interface. The absolute path to the file is set in the following environment variable:

OREMAN_INI_PATH		
NEW CREDENTIAL		0
DETAILS PERMISSIONS		
* NAME 🕜	DESCRIPTION 😨	ORGANIZATION 🔞
New Credential		Q Default
* CREDENTIAL TYPE 🔞		
Q Red Hat Satellite 6		
TYPE DETAILS		
* SATELLITE 6 URL 🔞	* USERNAME	* PASSWORD
		SHOW
		CANCEL SAVE

Satellite credentials have the following inputs that are required:

- Satellite 6 URL: The Satellite 6 URL or IP address to connect to.
- Username: The username to use to connect to Satellite 6.
- Password: The password to use to connect to Satellite 6.

10.4.11 Red Hat Virtualization

This credential allows Tower to access Ansible's oVirt4.py dynamic inventory plugin, which is managed by Red Hat Virtualization (RHV).

Tower uses the following environment variables for Red Hat Virtualization credentials and are fields in the user interface:

```
OVIRT_URL
OVIRT_USERNAME
OVIRT_PASSWORD
```

NEW CREDENTIAL			0
NAME RHV Credential		ORGANIZATION @	
CREDENTIAL TYPE Red Hat Virtualization TYPE DETAILS			
* HOST (AUTHENTICATION URL)	* USERNAME	* PASSWORD SHOW	
		CANCEL	

RHV credentials have the following inputs that are required:

- Host (Authentication URL): The host URL or IP address to connect to.
- Username: The username to use to connect to oVirt4.
- Password: The password to use to connect to it.
- CA File: Optionally provide an absolute path to the oVirt certificate file (it may end in .pem, .cer and .crt extensions, but preferably .pem for consistency)

10.4.12 Source Control

SCM (source control) credentials are used with Projects to clone and update local source code repositories from a remote revision control system such as Git, Subversion, or Mercurial.

NEW CREDENTIAL			Θ
DETAILS PERMISSIONS			
* NAME @	DESCRIPTION @	ORGANIZATION @	
credential type Q Source Control			
TYPE DETAILS			
USERNAME	PASSWORD		
	SHOW		
SCM PRIVATE KEY HINT: Drag and drop an SSH private key file on the field below.			
PRIVATE KEY PASSPHRASE			
SHOW			
		CANCEL	E

Source Control credentials have several attributes that may be configured:

- Username: The username to use in conjunction with the source control system.
- Password: The password to use in conjunction with the source control system.
- SCM Private Key: Copy or drag-and-drop the actual SSH Private Key to be used to authenticate the user to the source control system via SSH.

• **Private Key Passphrase**: If the SSH Private Key used is protected by a passphrase, you may configure a Key Passphrase for the private key.

Note: Source Control credentials cannot be configured as "Prompt on launch".

10.4.13 Vault

Selecting this credential type enables synchronization of inventory with Ansible Vault.

NEW CREDENTIAL			0
DETAILS PERMISSIONS			
* NAME 🕢	DESCRIPTION 😨	ORGANIZATION	
New credential		Q Default	
CREDENTIAL TYPE Q Vault			
TYPE DETAILS			
* VAULT PASSWORD Prompt on launch	VAULT IDENTIFIER 😨		
SHOW			
n			CANCEL

Vault credentials require the **Vault Password** and an optional **Vault Identifier** if applying multi-Vault credentialing. For more information on Ansible Tower Multi-Vault support, refer to the Multi-Vault Credentials section of the *Ansible Tower Administration Guide*.

You may configure Tower to ask the user for the password at launch time by selecting **Prompt on launch**. In these cases, a dialog opens when the job is launched, promoting the user to enter the password and password confirmation.

Warning: Credentials which are used in Scheduled Jobs must not be configured as "Prompt on launch".

For more information about Ansible Vault, refer to: http://docs.ansible.com/ansible/playbooks_vault.html

10.4.14 VMware vCenter

Selecting this credential type enables synchronization of inventory with VMware vCenter.

Tower uses the following environment variables for VMware vCenter credentials and are fields prompted in the user interface:

```
VMWARE_HOST
VMWARE_USER
VMWARE_PASSWORD
VMWARE_VALIDATE_CERTS
```

NEW CREDENTIAL DETAILS PERMISSIONS			Ø
*NAME 🔞	DESCRIPTION	ORGANIZATION	
New Credential		Q Default	
CREDENTIAL TYPE Q VMware vCenter TYPE DETAILS			
* VCENTER HOST 🔞	* USERNAME	* PASSWORD	
		SHOW	
		CANCEL	SAVE

VMware credentials have the following inputs that are required:

- vCenter Host: The vCenter hostname or IP address to connect to.
- Username: The username to use to connect to vCenter.
- **Password**: The password to use to connect to vCenter.

Note: If the VMware guest tools are not running on the instance, VMware inventory sync may not return an IP address for that instance.

CHAPTER

ELEVEN

CUSTOM CREDENTIAL TYPES

As a Tower administrator with superuser access, you can define a custom credential type in a standard format using a YAML/JSON-like definition, allowing the assignment of new credential types to jobs and inventory updates. This allows you to define a custom credential type that works in ways similar to existing credential types. For example, you could create a custom credential type that injects an API token for a third-party web service into an environment variable, which your playbook or custom inventory script could consume.

Custom credentials support the following ways of injecting their authentication information:

- Environment variables
- Ansible extra variables
- File-based templating (i.e., generating .ini or .conf files that contain credential values)

You can attach one SSH and multiple cloud credentials to a Job Template. Each cloud credential must be of a different type. In other words, only one AWS credential, one GCE credential, etc., are allowed. In Ansible Tower 3.2 and later, vault credentials and machine credentials are separate entities.

Note: When creating a new credential type, you are responsible for avoiding collisions in the extra_vars, env, and file namespaces. Also, avoid environment variable or extra variable names that start with ANSIBLE_because they are reserved. You must have Superuser permissions to be able to create and edit a credential type (CredentialType) and to be able to view the CredentialType.injection field.

11.1 Backwards-Compatible API Considerations

With Ansible Tower version 3.2, new support for version 2 of the API (V2) means:

- · One-to-many relationship for Job Templates to credentials (including multi-cloud support)
- Custom credentials will not be managed by the V1 API; if a user defines a custom credential type, its credentials will not show up in the V1 API
- POSTs to V1 credential API will transparently work with migrated CredentialTypes/Credentials

Credentials have the concept of "Kind" that dictates:

- How or *where* a credential can be used.
- You can attach one SSH and multiple cloud credentials to a Job Template. Each cloud credential must be of a different type. In other words, only one AWS credential, one GCE credential, etc.

In the V2 CredentialType model, the relationships are defined as follows:

Machine	SSH
Vault	Vault
Network	Sets environment variables (e.g., ANSIBLE_NET_AUTHORIZE)
SCM	Source Control
Cloud	EC2, AWS
	Lots of others
Insights	Insights

Custom type creation and modification are limited to cloud and network kinds.

11.2 Getting Started with Credential Types

Access the Credentials from clicking the Credential Types () icon from the left navigation bar. If no custom credential types have been created, the Credential Types view will not have any to display and will prompt you to add one:

CREDENTIAL TYPES	6
CREDENTIAL TYPES	
PLEASE ADD ITEMS TO THIS LIST	

If credential types have been created, this page displays a list of all existing and available Credential Types. It can be sorted and searched by **Name** and **Kind**.

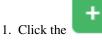
EDENTIAL TYPES		
EARCH	Q KEY	
NAME [*]	KIND 🗢	ACTION
New credential type	Cloud	Ø Ê
Another new credential type	Cloud	d l

To view more information about a credential type, click on its name or the Edit (\checkmark) button from the Actions column.

Each credential type displays its own unique configurations in the **Input Configuration** field and the **Injector Con-figuration** field, if applicable. Both YAML and JSON formats are supported in the configuration fields.

11.3 Create a New Credential Type

To create a new credential type:



button located in the upper right corner of the Credential Types screen.

CREDENTIAL TYPES / CREATE CREDENTIAL TYPE		0
NEW CREDENTIAL TYPE		0
* NAME	DESCRIPTION	
		CANCEL SAVE

2. Enter the appropriate details in the Name and Description field.

Note: When creating a new credential type, do not use reserved variable names that start with ANSIBLE_ for the **INPUT** and **INJECTOR** names and IDs, as they are invalid for custom credential types.

3. In the **Input Configuration** field, specify an input schema which defines a set of ordered fields for that type. The format can be in YAML or JSON, as shown:

YAML

```
fields:
    - type: string
    id: username
    label: Username
    type: string
    id: password
    label: Password
    secret: true
required:
    - username
    password
```

View more YAML examples at http://www.yaml.org/start.html.

JSON

{

```
"fields": [
{
"type": "string",
"id": "username",
```

(continues on next page)

(continued from previous page)

```
"label": "Username"
},
{
"secret": true,
"type": "string",
"id": "password",
"label": "Password"
}
],
"required": ["username", "password"]
}
```

View more JSON examples at www.json.org.

The configuration in JSON format below show each field and how they are used:

```
{
  "fields": [{
   "id": "api_token",
                                    # required - a unique name used to
                                    # reference the field value
   "label": "API Token",
                                   # required - a unique label for the
                                    # field
   "help_text": "User-facing short text describing the field.",
   "type": ("string" | "boolean")
                                   # defaults to 'string'
   "format": "ssh_private_key"
                                   # optional, can be used to enforce data
                                    # format validity for SSH private key
                                    # data (only applicable to_
"secret": true,
                                   # if true, the field value will be_
↔encrypted
   "multiline": false
                                    # if true, the field should be rendered
                                    # as multi-line for input entry
                                    # (only applicable to `type=string`)
},{
   # field 2...
},{
   # field 3...
}],
"required": ["api_token"]
                                   # optional; one or more fields can be_
→marked as required
},
```

When type=string, fields can optionally specify multiple choice options:

```
f
fields": [{
    "id": "api_token",    # required - a unique name used to_
    -reference the field value
    "label": "API Token",    # required - a unique label for the field
    "type": "string",
```

(continued from previous page)

```
"choices": ["A", "B", "C"]
}]
},
```

4. In the **Injector Configuration** field, enter environment variables or extra variables that specify the values a credential type can inject. The format can be in YAML or JSON (see examples in the previous step). The configuration in JSON format below show each field and how they are used:

```
{
   "env": {
    "THIRD_PARTY_CLOUD_API_TOKEN": "{{api_token}}"
   },
   "extra_vars": {
        "some_extra_var": "{{username}}:{{password}}"
   }
}
```

Credential Types can also generate temporary files to support .ini files or certificate/key data:

```
"file": {
    "template": "[mycloud] \ntoken={{api_token}}"
},
"env": {
    "MY_CLOUD_INI_FILE": "{{tower.filename}}"
}
```

In this example, Tower will write a temporary file that contains:

[mycloud] \ntoken=SOME_TOKEN_VALUE

The absolute file path to the generated file will be stored in an environment variable named MY_CLOUD_INI_FILE.

An example of referencing multiple files in a custom credential template is as follows:

Inputs

```
{
   "fields": [{
    "id": "cert",
    "label": "Certificate",
    "type": "string"
   },{
    "id": "key",
    "label": "Key",
    "type": "string"
   }]
}
```

Injectors

{

```
"file": {
    "template.cert": "[mycert]\n{{cert}}",
    "template.key": "[mykey]\n{{key}}"
},
```

(continues on next page)

(continued from previous page)

```
"env": {
    "MY_CERT_INI_FILE": "{{tower.filename.cert}}",
    "MY_KEY_INI_FILE": "{{tower.filename.key}}"
}
```

5. Click Save when done.

}

Click 🥔

6. Scroll down to the bottom of the screen and your newly created credential type appears on the list of credential types:

SEARCH	Q KEY	+
NAME [*]	KIND 🗢	ACTIONS
Another new credential type	Cloud	1947 m
New credential type	Cloud	Ø 🗎
new_cred_type	Cloud	a
		ITEMS 1

to modify or $\overline{\square}$ to remove the credential type options under the Actions column.

Note: If deleting a credential type that is being used by a credential, you must delete the credential type from all the credentials that use it before you can delete it. Below is an example of such a message:

CREDENTIAL TYPES			
CREDENTIAL TY	This credential	ORK TYPE type is currently being used by one or more credentials. Credent edential type must be deleted before the credential type can be	
NAME 🔶 Network Type		CA	CANCEL
			ITEMS 1 -

7. Verify that the newly created credential type can be selected from the **Credential Type** selection window when creating a new credential:

CREDENTIALS / CREATE CREDENTIAL			
NEW CREDENTIAL	SELECT CREDENTIAL TYPE	\$	
DETAILS PERMISSIONS	SEARCH	Q KEY	
* NAME @	NAME A		ORGANIZATION
new_credential	Network		Q Default
* CREDENTIAL TYPE	new_cred_type		
Q SELECT A CREDENTIAL TYPE	OpenStack		
	O Red Hat CloudForms		
	O Red Hat Satellite 6		
CREDENTIALS	< 1 2 3 > PAGE 2 OF 3	ITEMS 6 - 11 OF 14	
SEARCH		CANCEL SELECT	

For details on how to create a new credential, see Credentials.

CHAPTER

TWELVE

APPLICATIONS

Creating and configuring token-based authentication for external applications is available in Ansible Tower 3.3.

12.1 Getting Started with Applications

Access the Applications page by clicking the Applications (in the left navigation bar. The Applications page displays a search-able list of all available Applications currently managed by Tower and can be sorted by **Name**.

Schedules	APPLICATIONS	0
II My View		
RESOURCES	APPLICATIONS (3)	
🖉 Templates	SEARCH Q KEY	+
e Credentials	Application sample	
🗁 Projects	ORG Default	ŵ
inventories	LAST MODIFIED 7/23/2018 5:10:09 PM	
	my creds app	
ACCESS	ORG Default LAST MODIFIED 7/25/2018 11:59:42 AM	Û
. Organizations	New app	
🐣 Users	ORG Default	ŵ
😁 Teams	LAST MODIFIED 7/23/2018 5:10:52 PM	
ADMINISTRATION		ITEMS 1 - 3
Credential Types		

If no other applications exist, only a gray box with a message to add applications displays.

APPLICATIONS 0		
SEARCH	Q KEY	+
	PLEASE ADD ITEMS TO THIS LIST.	

12.2 Create a new application

Token-based authentication for users can be configured in the Applications window.

1. In the Ansible Tower User Interface, click the Applications (

The Applications window opens.

2. Click the **button** located in the upper right corner of the Applications window.

The New Application window opens.

NEW APPLICATION		8
DETAILS TOKENS		
* NAME	DESCRIPTION	* ORGANIZATION
		Q SELECT AN ORGANIZATION
* AUTHORIZATION GRANT TYPE 🕑	REDIRECT URIS 🔞	* CLIENT TYPE 🔞
•		· · · · ·
		CANCEL

- 3. Enter the following details in Create New Application window:
- Name (required): provide a name for the application you want to create
- Description: optionally provide a short description for your application
- Organization (required): provide an organization for which this application is associated
- Authorization Grant Type (required): Select from one of the grant types to use in order for the user to acquire tokens for this application. Refer to grant types in the Applications section of the *Ansible Tower Administration Guide*.
- **Redirect URIS**: Provide a list of allowed URIs, separated by spaces. This is required if you specified the grant type to be **Authorization code** or **Implicit**.
- Client Type (required): Select the level of security of the client device
- 4. When done, click Save or Cancel to abandon your changes

12.2.1 Applications - Tokens

Selecting the Tokens view displays a list of the users that have tokens to access the application.

Application sample		8
DETAILS TOKENS		
SEARCH	Q KEY	
admin		

Tokens can only access resources that its associated user can access, and can be limited further by specifying the scope of the token.

Add Tokens

Tokens are added through the Users screen before they can be associated with an application. Specifying an application can be performed directly in the User's token settings. You can create a token for your user in the Tokens configuration tab. To add a token:

- 1. Access the Users list view by clicking the Users () icon from the left navigation bar then click on your user to configure your OAuth 2 tokens.
- 2. Click the Tokens tab from your user's profile.

When no tokens are present, the Tokens screen prompts you to add them:

admin admin	0
DETAILS ORGANIZATIONS TEAMS PERMISSIONS TOKENS	
SEARCH Q KEY	+
PLEASE ADD ITEMS TO THIS LIST.	

3. Click the

- 4. Enter the following details in Create Token window:
- Application: enter the name of the application with which you want to associate your token. Alternatively, you

can search for it by clicking the button. This opens a separate window that allows you to choose from the available options. Use the Search bar to filter by name if the list is extensive. Leave this field blank if you want to create a Personal Access Token (PAT) that is not linked to any application.

- Description: optionally provide a short description for your token.
- Scope (required): specify the level of access you want this token to have.

button, which opens the Create Token window.

5. When done, click Save or Cancel to abandon your changes.

After the token is saved, the newly created token for the user displays with the token information and when it expires.

ſ	TOKEN INFORMAT	TION	8
EA	TOKEN REFRESH TOKEN EXPIRES	ufCk6HsQB5b89ALtXQHDYQbreR2BDt CeBZ5MUOj3AzgbBDVAF0TmsHgHLYh2 11/25/3017 5:27:34 PM	
			ОК

Note: This is the only time the token value and associated refresh token value will ever be shown.

In the user's profile, the application for which it is assigned to and its expiration displays in the token list view.

admin ADMIN		0
DETAILS ORGANIZATIONS TEAMS PERMISSION	TOKENS	
SEARCH	Q KEY	
my creds app Token		
APPLICATION my creds app		面
EXPIRATION 11/25/3017 11:00:22 AM		

To verify the application in the example above now shows the user with the appropriate token, go to the **Tokens** tab of the Applications window:

my creds app	0
DETAILS TOKENS	
SEARCH Q KEY	
admin	

CHAPTER

THIRTEEN

PROJECTS

A Project is a logical collection of Ansible playbooks, represented in Tower.

You can manage playbooks and playbook directories by either placing them manually under the Project Base Path on your Tower server, or by placing your playbooks into a source code management (SCM) system supported by Tower, including Git, Subversion, Mercurial, and Red Hat Insights. To create a Red Hat Insights project, refer to *Setting up an Insights Project*.

Note: By default, the Project Base Path is /var/lib/awx/projects, but this may have been modified by the Tower administrator. It is configured in /etc/tower/settings.py. Use caution when editing this file, as incorrect settings can disable your installation.

This menu displays a list of the projects that are currently available. The list of projects may be sorted and searched by any of the table headers displayed.

ROJECTS							
PROJECTS							
SEARCH		QKEY					+
NAME 🔶	TYPE 🗢	REVISION \$	LAST UPDATED 🗢			ACT	TIONS
O Demo Project	Git			Ø	C	4	Û
Project from Git	Git	b2cf1f0	10/5/2018 3:55:16 PM	Ø	C	2	Û
						1	ITEMS 1 - 2

Status indicates the state of the project and may be one of the following (note that you can also filter your view by specific status types):

- **Pending** The source control update has been created, but not queued or started yet. Any job (not just source control updates) will stay in pending until it's actually ready to be run by the system. Reasons for it not being ready because it has dependencies that are currently running so it has to wait until they are done, or there is not enough capacity to run in the locations it is configured to.
- Waiting The source control update is in the queue waiting to be executed.
- Running The source control update is currently in progress.
- Successful The last source control update for this project succeeded.
- Failed The last source control update for this project failed.
- Error The last source control update job failed to run at all. (To be deprecated.)
- Canceled The last source control update for the project was canceled.

- Never updated The project is configured for source control, but has never been updated.
- OK The project is not configured for source control, and is correctly in place. (To be deprecated.)
- Missing Projects are absent from the project base path of /var/lib/awx/projects (applicable for manual or source control managed projects).

For each project listed, you can edit () project properties, copy the project attributes (), get the latest SCM revision (), or delete () the project, using the respective icons under the **Actions** column.

Note: Projects of credential type Manual cannot update or schedule source control-based actions without being reconfigured as an SCM type credential.

Note: If deleting items that are used by other work items, a message opens listing the items are affected by the deletion and prompts you to confirm the deletion. Some screens will contain items that are invalid or previously deleted, so they will fail to run. Below is an example of such a message:

		💄 admin	0		<u>ا</u> ر
PROJECTS					•
PROJECTS 2	DELETE PROJECT FROM GIT				Ŧ
NAME 🔶				ACT	IONS
O Demo Project	CANCEL	(J ^{ab}	C	2	Ŵ
Project from Git	Git b2cflf0 🗋 10/5/2018 3:55:16 PM	Ø	Q	4	面 TEMS 1 - 2
					TEMS 1 - 2

13.1 Add a new project

To create a new project:



button, which launches the Create Project dialog.

NEW PROJECT			0
DETAILS PERMISSIONS NOTIFICATIONS JOB TEMPLATES	SCHEDULES		
* NAME	DESCRIPTION	* ORGANIZATION	
		Q Default	
ANSIBLE ENVIRONMENT 🚱	* SCM TYPE		
Select Ansible Environment	Choose an SCM Type 👻		
		CANCEL	

2. Enter the appropriate details into the following required fields:

- Name
- **Description** (optional)
- **Organization** A project must have at least one organization. Pick one organization now to create the project, and then after the project is created you can add additional organizations.
- Ansible Environment (optional) Select from the drop-down menu list a custom virtual environment on which to run this project.
- SCM Type Select from the drop-down menu list an SCM type associated with this project. Refer to *Manage* playbooks manually and *Manage* playbooks using Source Control in the subsequent sections for more detail.

Note: If adding a manual project, each project path inside of the project root folder can only be assigned to one project. If you receive the following message, ensure that you have not already assigned the project path to an existing project:

```
All of the project paths have been assigned to existing projects, or
there are no directories found in the base path. You will need to add
a project path before creating a new project.
```

3. Click Save when done.

13.1.1 Manage playbooks manually

- Create one or more directories to store playbooks under the Project Base Path (for example, /var/lib/awx/projects/)
- Create or copy playbook files into the playbook directory.
- Ensure that the playbook directory and files are owned by the same UNIX user and group that the Tower service runs as.
- Ensure that the permissions are appropriate for the playbook directories and files.

If you have trouble adding a project path, check the permissions and SELinux context settings for the project directory and files.

IEW PROJECT			
DETAILS PERMISSIONS NOTIFICATIONS JO	DB TEMPLATES SCHEDULES		
NAME	DESCRIPTION	* ORGANIZATION	
Example	Ansible example playbook	Q Honey Dog, Inc.	
NSIBLE ENVIRONMENT @	* SCM TYPE		
Select Ansible Environment	▼ Manual	•	
	var/lib/aws/projects. Either that directory is empty, or all of the contents a books from source control using the SCM Type option above.	are already assigned to other projects. Create a new directory there and make sure t	ne playbook files can be read by the
/var/lib/awx/projects			

Correct this issue by creating the appropriate playbook directories and checking out playbooks from your SCM or otherwise copying playbooks into the appropriate playbook directories.

13.1.2 Manage playbooks using Source Control

- 1. Select the appropriate option from the SCM Type drop-down menu list.
- 2. Enter the appropriate details into the following fields:
 - SCM URL See an example in the help 🕑 text.
 - SCM Branch Optionally enter the SCM branch for Mercurial, or the SCM branch, tag, or revision for Git
 - Revision # Optionally enter the Revision # for Subversion
 - SCM Credential If authentication is required, select the appropriate SCM credential
 - SCM Update Options:
 - Clean Remove any local modifications prior to performing an update.
 - **Delete on Update** Delete the local repository in its entirety prior to performing an update. Depending on the size of the repository this may significantly increase the amount of time required to complete an update.
 - Update on Launch Each time a job runs using this project, perform an update to the local repository prior to starting the job. To avoid job overflows if jobs are spawned faster than the project can sync, selecting this allows you to configure a Cache Timeout to cache prior project syncs for a certain number of seconds.

NEW PROJECT			0
DETAILS PERMISSIONS NOTIFICATIONS JOB TEMPLATES	SCHEDULES		
* NAME	DESCRIPTION	* ORGANIZATION	
Example	Ansible example playbook	Q Honey Dog, Inc.	
ANSIBLE ENVIRONMENT @	* SCM TYPE		
Select Ansible Environment *	Git •		
SOURCE DETAILS			
* SCM URL @	SCM BRANCH/TAG/COMMIT	SCM CREDENTIAL	
https://github.com/ansible/tower-example		Q	
SCM UPDATE OPTIONS			
Clean @			
Delete on Update Update Revision on Launch			
		CANCEL	AVE

3. Click **Save** to save your project.

Tip: Using a Github link offers an easy way to use a playbook. To help get you started, use the helloworld.yml file available at: https://github.com/ansible/tower-example.git

This link offers a very similar playbook to the one created manually in the instructions found in the Ansible Tower Quick Start Guide. Using it will not alter or harm your system in anyway.

Updating projects from source control

1. Update an existing SCM-based project by selecting the project and clicking the ⁴ button.

Note: Please note that immediately after adding a project setup to use source control, a "Sync" starts that fetches the project details from the configured source control.

ROJECTS							(
PROJECTS							
SEARCH		QKEY					+
NAME 🔶	TYPE 🗢	REVISION \$	LAST UPDATED 🗢			AC	TIONS
O Demo Project	Git			ø	0	4	Ē
Project from Git	Git	b2cf1f0	10/5/2018 3:55:16 PM	Ø	Q	2	Û
							ITEMS 1 - 2

2. Click on the dot under **Status** (far left, beside the name of the Project) to get further details about the update process.

JOBS / DEMO PROJECT		
RESULTS	A 🖻	STANDARD OUT 🔀 🛓
NAMEDemo ProjectSTATUS- SuccessfulSTARTED2/27/2017 1:43:26 PMFINISHED2/27/2017 1:43:30 PMELAPSED3.775 secondsLAUNCH TYPEDependencyPROJECTDemo Project		Using /etc/ansible/ansible.cfg as config file PLAY [all] ***********************************

13.2 Work with Permissions

The set of Permissions assigned to this project (role-based access controls) that provide the ability to read, modify, and administer projects, inventories, job templates, and other Tower elements are Privileges.

You can access the project permissions via the **Permissions** tab next to the **Details** tab. This screen displays a list of users that currently have permissions to this project. The list may be sorted and searched by **User**, **Role**, or **Team Role**.

Q KEY
TEAM ROLES
STRATOR

13.2.1 Add Permissions

The **Permissions** tab allows you to review, grant, edit, and remove associated permissions for users as well as team members. To assign permissions to a particular user for this resource:

1. Click the **Permissions** tab.

+

- 2. Click the
- button to open the Add Users/Teams window.

/ DEMO EXAMPLE / PE	ERMISSIONS			
MPLE	DEMO EXAMPLE ADD USERS	7 TEAMS	0	
PERMISSIONS	Please select Users / Teams USERS TEAMS	s from the lists below.		
	SEARCH		Q, KEY	
	USERNAME [▲]	FIRST NAME	LAST NAME	
	🗆 althea	Althea	Bully	
	austin78	Austin	Texas	
	□ gdoge	Gerry	Doge	
ES HOSTS	🗆 jdoge	Josie	Doge	
	🗆 jgarcia	Jerry	Garcia	
NAME 🕈	< 1 2 > PAGE 1 OF 2		ITEMS 1 - 5 OF 6	
Database Servers			CANCEL	
DEMO EXAMPLE				

- 3. Specify the users or teams that will have access then assign them specific roles:
 - a. Click to select one or multiple checkboxes beside the name(s) of the user(s) or team(s) to select them.

Note: You can select multiple users and teams at the same time by navigating between the **Users** and **Teams** tabs without saving.

After selections are made, the window expands to allow you to select a role from the drop-down menu list for each user or team you chose.

/ DEMO EXAMPLE / PE	RMISSIONS			
	DEMO EXAMPLE ADD USE	RS / TEAMS	0	
MPLE	1 Please select Users / Tea	ms from the lists below.		
PERMISSIONS	USERS			
	SEARCH		QKEY	
	USERNAME [^]	FIRST NAME	LAST NAME	
	🥑 althea	Althea	Bully	
	austin78	Austin	Texas	
	□ gdoge	Gerry	Doge	
ES HOSTS	🗆 jdoge	Josie	Doge	
	🗆 jgarcia	Jerry	Garcia	
NAME 🔦	< 1 2 > PAGE 1 OF 2		ITEMS 1 - 5 OF 6	
Database Servers	2 Please assign roles to the	e selected users/teams	KEY	
DEMO EXAMPLE	Althea Bully USER	SELECT ROLES		
Demo Inventory	Althea Bully USER	Admin	×	
King PLC		Update	SAVE	
		Ad Hoc		
		Use		
		Read		

The example above shows options associated with inventories. Different resources have different options available:

- Admin allows read, run, and edit privileges (applies to all resources)
- Use allows use of a resource in a job template (applies all resources except job templates)
- Update allows updating of project via the SCM Update (applies to projects and inventories)
- Ad Hoc allows use of Ad Hoc commands (applies to inventories)
- Execute allows launching of a job template (applies to job templates)
- Read allows view-only access (applies to all resources)

Tip: Use the Key button in the roles selection pane to display a description of each of the roles.

b. Select the role to apply to the selected user or team.

Note:

/ DEMO EXAMPLE / PERMISSIONS						
MPLE	DEMO EXAMPLE ADD USE			•		
PERMISSIONS	USERS TEAMS SEARCH		٩	KEY		
	USERNAME [▲]	FIRST NAME	LAST NAME 🗢			
	🥑 althea	Althea	Bully			
	austin78	Austin	Texas			
	□ gdoge	Gerry	Doge			
ES HOSTS	✓ jdoge	Josie	Doge	_		
	🗆 jgarcia	Jerry	Garcia			
NAME 🔺	< 1 2 > PAGE 1 OF 2		п	'EMS 1 - 5 OF 6		
Database Servers	2 Please assign roles to the	e selected users/teams		KEY		
Demo Inventory	Althea Bully USER	SELECT ROLES		×		
King PLC	Josie Doge USER	SELECT ROLES		×		
	Production Operatio TEAM	SELECT ROLES		×		
			CANCEL	SAVE		

You can assign roles to multiple users and teams by navigating between the **Users** and **Teams** tabs without saving.

4. Review your role assignments for each user and team.

/ DEMO EXAMPLE / PI	ERMISSIONS				
	DEMO EXAMPLE ADD USE	RS / TEAMS		•	
MPLE	1 Please select Users / Tear	ns from the lists below.			
PERMISSIONS	USERS				
	SEARCH		Q	KEY	
	USERNAME [▲]	FIRST NAME	LAST NAME		
	🗹 althea	Althea	Bully		
	austin78	Austin	Texas	_	
	□ gdoge	Gerry	Doge		
ES HOSTS	🗹 jdoge	Josie	Doge		
	🗆 jgarcia	Jerry	Garcia		
NAME 🔺	< 1 2 > PAGE 1 OF 2		ITEMS	1 - 5 OF 6	
Database Servers	2 Please assign roles to the	selected users/teams		KEY	
DEMO EXAMPLE	Althea Bully USER	× Update		×	
Demo Inventory	-				
King PLC	Josie Doge USER	× Use		×	
	Production Operatio TEAM	× Admin		×	
			CANCEL	SAVE	

5. Click **Save** when done, and the Add Users/Teams window closes to display the updated roles assigned for each user and team.

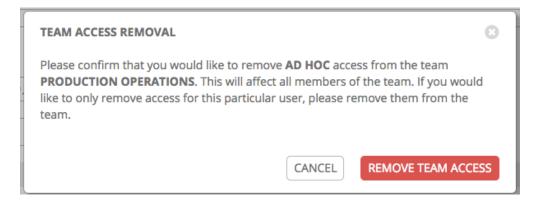
USER *	ROLE	TEAM ROLES
admin	SYSTEM ADMINISTRATOR	
althea	× AD HOC SYSTEM AUDITOR X USE	
jdoge	× UPDATE × USE	
mags3707	SYSTEM ADMINISTRATOR	× AD HOC 알 × ADMIN 알 × USE 알
yser	SYSTEM AUDITOR	

To remove Permissions for a particular user, click the Disassociate (x) button next to its resource.

USER ^	ROLE	TEAM ROLES
		TDAM ROLES
admin	SYSTEM ADMINISTRATOR	
althea	* AD HOC SYSTEM AUDITOR X USE	
jdoge	× UPDATE × USE	
mags3707	SYSTEM ADMINISTRATOR	X AD HOC W X ADMIN W X USE W
yser	SYSTEM AUDITOR	
		ITEMS 1-5

This launches a confirmation dialog, asking you to confirm the disassociation.

TEMS 1-5



13.3 Work with Notifications

gineering			
DETAILS PERMISSIONS NOTIFICATIO	NS JOB TEMPLATES SCHEDULES		
SEARCH	٩	KEY	GO TO NOTIFICATIONS ADD A NEW TEMPL
NAME 🔺	TYPE 🗢	SUCCESS	FAILURE
Amazing notification template	Email	(NO)	ON
Engineering Notification Bot	HipChat	ON)	OFF
Jenkins Slack Channel Notification Bot	Slack	OFF	(ON)
SNotification	SMS	OFF	OFF

Clicking on Notifications allows you to review any notification integrations you have setup.

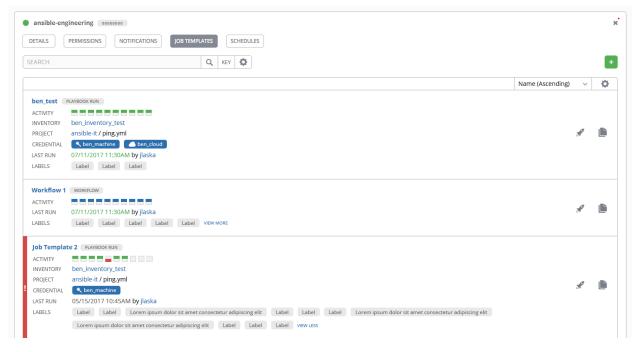
To create a new notification, click the **NOTIFICATIONS** link from the upper-right side of the notifications list view. If no notifications have been set up, click the **NOTIFICATIONS** link from above or inside the gray box to add a new notification to create a notification.

Refer to Notifications for more information.

emo Job Template			
DETAILS PERMISSIONS NOTIFICATIONS	COMPLETED JOBS SCHEDULES		
SEARCH	Q KEY		GO TO NOTIFICAT ADD A NEW TH
NAME 🔶	TYPE 🗢	SUCCESS	FAILURE
test-actions-notification test notification-template	Slack	OFF	OFF
e2e-ae53906d-notification-template	SMS	OFF	OFF
test-actions-notification-template	Slack	OFF	OFF
test-actions-notification-template@3:05:07 PM	EMAIL	OFF	OFF
			п

13.4 Work with Job Templates

Clicking on Job Templates allows you to review any job templates or workflow templates associated with this project.



From this view, you can launch or copy the template configuration.

13.5 Work with Schedules

Clicking on Schedules allows you to review any schedules set up for this project.

DETAILS	PERMISSIONS	JOB TEMPLATES SCHEDULES				
SEARCH		Q				+
	NAME [*]	FIRST RUN 🗢	NEXT RUN 🗢	FINAL RUN 🗢	AC	TIONS
ON	Schedule 1	9/13/2018 1:00:00 AM	9/13/2018 1:00:00 AM	9/13/2018 1:00:00 AM	an a	Û
ON	Schedule 2	10/23/2018 9:45:00 AM	10/23/2018 9:45:00 AM		dit.	Ŵ
ON	Schedule 3	9/13/2018 11:15:00 PM	9/13/2018 11:15:00 PM	9/27/2018 11:15:00 PM	den a	Ô
ON	Schedule 4	10/31/2018 11:30:00 PM	10/31/2018 11:30:00 PM	10/31/2023 11:30:00 PM	60	Ê
						ITEMS

From this view, you can select schedules to edit, turn on or off, or select multiple schedules to delete.

This screen displays a list of the schedules that are currently available for the selected **Project**. The schedule list may be sorted and searched by **Name**.

The list of schedules includes:

- Name: Clicking the schedule name opens the Edit Schedule dialog
- First Run: The first scheduled run of this task
- Next Run: The next scheduled run of this task
- Final Run: If the task has an end date, this is the last scheduled run of the task
- Last Modified: The last time this schedule was modified

13.5.1 Add a new schedule

To create a new schedule:

	+	
1. Click the		butto

button, which opens the Add Schedule dialog.

NEW SCHEDULE				×
* NAME	* START DATE (MM/DD/YYYY)		* START TIME (HH24:MM:SS)	
	11/19/2015	Ê	00 \$: 00 \$: 00 \$	UTC START TIME 11/19/2015 05:00:00 UTC
LOCAL TIME ZONE	REPEAT FREQUENCY			
US / New York v	None (Run Once)	~		
VARIABLES YAML JSON				
1				
SCHEDULE SUMMARY				
One time only.	-			
OCCURRENCES (Limited to first 10) DATE FORMAT LOCAL TIME U	inc.			
11/19/2015 00:00:00 EST				
				CANCEL

- 2. Enter the appropriate details into the following fields:
- Name (required)
- **Start Date** (required)
- Start Time (required)
- Local Time Zone The entered Start Time should be in this timezone
- UTC Start Time Calculated from Start Time + Local Time Zone
- Repeat Frequency Appropriate scheduling options are displayed depending on the frequency you select

The **SCHEDULE DESCRIPTION** allows you to review the set schedule and a list of the scheduled occurrences in the selected Local Time Zone.

Caution: Jobs are scheduled in UTC. Repeating jobs that run at a specific time of day may move relative to a local timezone when Daylight Savings Time shifts occur. Essentially, Tower resolves the local time zone based time to UTC when the schedule is saved. To ensure your schedules are correctly set, you should set your schedules in UTC time.

3. Once done, click **Save**.

You can use the **ON/OFF** toggle button to stop an active schedule or activate a stopped schedule.

The schedules overview screen for the project also shows you when the first, next, and final runs are scheduled.

DEMO PROJECT SCHEDULES				
SEARCH		Q KEY		+ ADD
NAME 🔶	FIRST RUN 🗢	NEXT RUN 🗘	FINAL RUN 🗢	ACTIONS
ON New schedule	3/2/2017 5:01:02 AM	3/2/2017 5:01:02 AM	3/2/2017 5:01:02 AM	a 🕅
				ITEMS 1 - 1

13.5.2 Ansible Galaxy Support

At the end of a Project update, Tower searches for a file called requirements.yml in the roles directory, located at``<project-top-level-directory>/roles/requirements.yml``. If this file is found, the following command automatically runs:

ansible-galaxy install -r roles/requirements.yml -p ./roles/ --force

This file allows you to reference Galaxy roles or roles within other repositories which can be checked out in conjunction with your own project. The addition of this Ansible Galaxy support eliminates the need to create git submodules for achieving this result.

For more information and examples on the syntax of the requirements.yml file, refer to Advanced Control Over Role Requirements in the Ansible documentation.

If there are any directories that should specifically be exposed, you can specify those in the Configure Tower screen in the **Paths to Expose to Isolated Jobs** or by updating the following entry in the settings file:

AWX_PROOT_SHOW_PATHS = ['/list/of/', '/paths']

Note: The primary file you may want to add to AWX_PROOT_SHOW_PATHS is /var/lib/awx/. ssh, if your playbooks need to use keys or settings defined there.

If you made changes in the settings file, be sure to restart services with the ansible-tower-service restart command after your changes have been saved.

CHAPTER

FOURTEEN

INVENTORIES

An Inventory is a collection of hosts against which jobs may be launched, the same as an Ansible inventory file. Inventories are divided into groups and these groups contain the actual hosts. Groups may be sourced manually, by entering host names into Tower, or from one of Ansible Tower's supported cloud providers.

Note: If you have a custom dynamic inventory script, or a cloud provider that is not yet supported natively in Tower, you can also import that into Tower. Refer to Inventory File Importing in the *Ansible Tower Administration Guide*.

This tab displays a list of the inventories that are currently available. The inventory list may be sorted and searched by **Name**, **Type**, or **Organization**.

INVENTORIES HOSTS			
SEARCH	Q KEY		
NAME 🔺	TYPE ≑	ORGANIZATION ≑	ACTIONS
Demo Inventory	Inventory	Default	<i>₽</i> 4 îi
Network Inventory Small	Inventory	Default	ø 🖄 🗎
			ITEMS

The list of Inventory details includes:

• Inventory Sync (): Green indicates successful syncs in the inventory, and red indicates failed syncs. Clicking this icon displays the sync status for the last five inventory source syncs and source information, if the inventory has sources that are able to sync.

	TORIES	HOSTS		
SEAR	СН			
	SYNC ST	ATUS		
		ATUS Last Sync	Source	

- Status Dot: This shows the status of recent jobs for this inventory.
- Name: The inventory name. Clicking the Inventory name navigates to the properties screen for the selected inventory, which shows the inventory's groups and hosts. (This view is also accessible from the icon.)
- **Type**: Identifies whether it is a standard inventory or a Smart Inventory.
- Organization: The organization to which the inventory belongs.
- Actions: The following actions are available for the selected inventory:
 - Edit: Edit the properties for the selected inventory
 - Copy: Makes a copy of an existing inventory as a template for creating a new one
 - Delete: Delete the selected inventory. This operation cannot be reversed!

Note: If deleting items that are used by other work items, a message opens listing the items are affected by the deletion and prompts you to confirm the deletion. Some screens will contain items that are invalid or previously deleted, so they will fail to run. Below is an example of such a message:

		🛔 admin	0	l		
PROJECTS						
PROJECTS 2 SEARCH NAME ^	DELETE PROJECT FROM GIT The project is currently being used by other resources. Are you sure you want to delete this project? Job Templates CANCEL DELETE		0	ACT 41	+ TIONS	
Demo Project Project from Git	Git b2cf1f0 10/5/2018 3:55:16 PM	J.	2	بط لک		
					ITEMS 1 -	2

14.1 Smart Inventories

A Smart Inventory is a collection of hosts defined by a stored search that can be viewed like a standard inventory and made to be easily used with job runs. Organization administrators have admin permission to inventories in their organization and can create Smart Inventories. A Smart Inventory is identified by KIND=smart. You can define a Smart Inventory using the same method being used with Tower Search. InventorySource is directly associated with an Inventory.

The Inventory model has the following new fields that are blank by default but are set accordingly for Smart Inventories:

- kind is set to smart for Smart Inventories
- host_filter is set AND kind is set to smart for Smart Inventories.

The host model has a new field, smart_inventories that uses a membership lookup table that identifies a set of all the Smart Inventory a host is associated with. The memberships are generated by a task. The task is launched when:

- a new host is added
- an existing host is modified (updated or deleted)
- a new Smart Inventory is added
- an existing Smart Inventory is modified (updated or deleted)

Note: The update_host_smart_inventory_memberships task is only run if the AWX_REBUILD_SMART_MEMBERSHIP is set to True (default is False).

You can view actual inventories without being editable:

- Names of Host and Group created as a result of an inventory source sync
- Group records cannot be edited or moved

You cannot create hosts from a Smart Inventory host endpoint (/inventories/N/hosts/) as with a normal inventory. The administrator of a Smart Inventory has permission to edit fields such as the name, description, variables, and the ability to delete, but does not have the permission to modify the host_filter, because that will affect which hosts (that have a primary membership inside another inventory) are included in the smart inventory. Note, host_filter only apply to hosts inside of inventories inside of the Smart Inventory's organization.

In order to modify the host_filter, you need to be the organization administrator of the inventory's organization. Organization admins already have implicit "admin" access to all inventories inside the organization, therefore, this does not convey any permissions they did not already possess.

Administrators of the Smart Inventory can grant other users (who are not also admins of your organization) permissions like "use" "adhoc" to the smart inventory, and these will allow the actions indicate by the role, just like other standard inventories. However, this will not give them any special permissions to hosts (which live in a different inventory). It will not allow them direct read permission to hosts, or permit them to see additional hosts under /#/hosts/, although they can still view the hosts under the smart inventory host list.

In some situations, you can modify the following:

- A new Host manually created on Inventory w/ inventory sources
- In Groups that were created as a result of inventory source syncs
- Variables on Host and Group are changeable

Hosts associated with the Smart Inventory are manifested at view time. If the results of a Smart Inventory contains more than one host with identical hostnames, only one of the matching hosts will be included as part of the Smart Inventory, ordered by Host ID.

14.1.1 host_filter Search

You can search host_filter by host name, group name, and Ansible facts.

The format for a group search is:

groups.name:groupA

The format for a fact search is:

```
ansible_facts.ansible_fips:false
```

You can also perform Smart Search searches, which consist a host name and host description.

```
host_filter=name=my_host
```

If a search term in host_filter is of string type, to make the value a number (e.g. 2.66), or a JSON keyword (e.g. null, true or false) valid, add double quotations around the value to prevent Tower from mistakenly parsing it as a non-string:

```
host_filter=ansible_facts_packages_dnsmasq[]_version="2.66"
```

14.2 Add a new inventory

To create a new inventory or Smart Inventory:



button, and select the type of inventory to create.

The type of inventory is identified by the labels and the row of tabs across the top of the create form.

	INVENTORIES / CREATE SMART INVENTORY		
INVENTORIES / CREATE INVENTORY	NEW SMART INVENTORY SMART INVENTORY DETAILS PERMISSIONS HOSTS COMPLETED JOBS		Ø
NEW INVENTORY			8
DETAILS PERMISSIONS GROU	UPS HOSTS SOURCES COMPLETED JOAS DESCRIPTION	* ORGANIZATION Q Default	
VARIABLES VAML JSON	Please solicit an organization before exting the host filter.		
		CAN	ACEL SAVE

- 2. Enter the appropriate details into the following fields:
- Name: Enter a name appropriate for this inventory.
- Description: Enter an arbitrary description as appropriate (optional).
- Organization: Required. Choose among the available organizations.
- Smart Host Filter: (Only applicable to Smart Inventories) Click the button to open a separate Dynamic Hosts window to filter hosts for this inventory. These options are based on the organization you chose.

Filters are similar to tags in that tags are used to filter certain hosts that contain those names. Therefore, to populate the **Smart Host Filter** field, you are specifying a tag that contains the hosts you want, not actually selecting the hosts themselves. Enter the tag in the **Search** field and press [Enter]. Filters are case-sensitive. Refer to the *Smart Host Filter* section for more information.

DYNAMIC HOSTS	2	DYNAMI	C HOSTS		0
local	Q KEY	SEARCH	1	Q	KEY
NAME *	INVENTORY \$	× local	CLEAR ALL		
.host-000001.group-00000.dummy	Inventory 1 Org 0	NAME	*	INVENTORY \$	
.host-000002.group-00000.dummy	Inventory 1 Org 0	lo localho	ost	Demo Inventory	
.host-000003.group-00000.dummy	Inventory 1 Org 0				
.host-000004.group-00000.dummy	Inventory 1 Org 0				ITEMS 1-1
.host-000005.group-00000.dummy	Inventory 1 Org 0			CANCEL	SAVE
< 1 2 3 4 5 6 7 8 > PAGE 1 OF 8	ITEMS 1 - 5 OF 38				
	CANCEL				

- **Insights Credential**: (Only applicable to standard inventories) Enter the appropriate Insights credential if the inventory is used with Insights.
- Instance Groups: Click the button to open a separate window. Choose the instance groups for this inventory to run on. If the list is extensive, use the search to narrow the options.
- Variables: Variable definitions and values to be applied to all hosts in this inventory. Enter variables using either JSON or YAML syntax. Use the radio button to toggle between the two.

INVENTORIES / CREATE INVENTORY			0
NEW INVENTORY DETAILS PERMISSIONS GROUPS HOSTS SOURCE	COMPLETED JOBS		0
NAME Database Servers	DESCRIPTION dbservers	ORGANIZATION Q Honey Dog. Inc.	
	INSTANCE GROUPS @		
VARIABLES @ YAML JSON			
		CA	NCEL

3. Click Save when done.

After Tower saves the new inventory, you can proceed with configuring permissions, groups, hosts, sources, and view completed jobs, if applicable to the type of inventory. For more instructions, refer to the subsequent sections.

14.2.1 Add Permissions

The **Permissions** tab allows you to review, grant, edit, and remove associated permissions for users as well as team members. To assign permissions to a particular user for this resource:

- 1. Click the **Permissions** tab.
- 2. Click the

button to open the Add Users/Teams window.

/ DEMO EXAMPLE / PE	ERMISSIONS			
	DEMO EXAMPLE ADD USER	RS / TEAMS	•	
MPLE	1 Please select Users / Tean	ns from the lists below.		
PERMISSIONS	USERS			
	SEARCH		Q KEY	
	USERNAME [▲]	FIRST NAME	LAST NAME	
	🗆 althea	Althea	Bully	
	austin78	Austin	Texas	
	□ gdoge	Gerry	Doge	
S HOSTS	🗆 jdoge	Josie	Doge	
	🗆 jgarcia	Jerry	Garcia	
NAME A	< 1 2 >> PAGE 1 OF 2		ITEMS 1 - 5 OF 6	
Database Servers			CANCEL	

3. Specify the users or teams that will have access then assign them specific roles:

a. Click to select one or multiple checkboxes beside the name(s) of the user(s) or team(s) to select them.

Note: You can select multiple users and teams at the same time by navigating between the **Users** and **Teams** tabs without saving.

After selections are made, the window expands to allow you to select a role from the drop-down menu list for each user or team you chose.

/ DEMO EXAMPLE / P	ERMISSIONS			
	DEMO EXAMPLE ADD U	SERS / TEAMS		0
MPLE	1 Please select Users / T	eams from the lists below.		
PERMISSIONS	USERS			
	SEARCH		Q	KEY
	USERNAME [▲]	FIRST NAME	LAST NAME 🗘	
	🥑 althea	Althea	Bully	
	austin78	Austin	Texas	
	gdoge	Gerry	Doge	
ES HOSTS	🗆 jdoge	Josie	Doge	
	🗆 jgarcia	Jerry	Garcia	
NAME *	< 1 2 > PAGE 1 OF 2		ITE	EMS 1 - 5 OF 6
Database Servers	2 Please assign roles to	the selected users/teams		KEY
DEMO EXAMPLE	Althea Bully USER	SELECT ROLES		×
Demo Inventory	Aithea Bully Osek	Admin		_ ^
King PLC		Update		SAVE
		Ad Hoc		
		Use		
		Read		

The example above shows options associated with inventories. Different resources have different options available:

- Admin allows read, run, and edit privileges (applies to all resources)
- Use allows use of a resource in a job template (applies all resources except job templates)
- Update allows updating of project via the SCM Update (applies to projects and inventories)
- Ad Hoc allows use of Ad Hoc commands (applies to inventories)
- **Execute** allows launching of a job template (applies to job templates)
- Read allows view-only access (applies to all resources)

Tip: Use the **Key** button in the roles selection pane to display a description of each of the roles.

b. Select the role to apply to the selected user or team.

Note:

/ DEMO EXAMPLE / PER	MISSIONS			
	DEMO EXAMPLE ADD USER	S / TEAMS		0
MPLE	1 Please select Users / Team	s from the lists below.		
PERMISSIONS	USERS			
	SEARCH		Q	KEY
	USERNAME	FIRST NAME	LAST NAME 🗘	
	✓ althea	Althea	Bully	
	austin78	Austin	Texas	
	□ gdoge	Gerry	Doge	
ES HOSTS	🗹 jdoge	Josie	Doge	
	🗋 jgarcia	Jerry	Garcia	
NAME *	< 1 2 > PAGE 1 OF 2		ITE	NS 1 - 5 OF 6
Database Servers	2 Please assign roles to the s	selected users/teams		KEY
Demo Inventory	Althea Bully USER	SELECT ROLES		×
King PLC	Josie Doge USER	SELECT ROLES		×
King PLC	Production Operatio TEAM	SELECT ROLES		×
			CANCEL	SAVE
				_

You can assign roles to multiple users and teams by navigating between the **Users** and **Teams** tabs without saving.

4. Review your role assignments for each user and team.

/ DEMO EXAMPLE / PI	ERMISSIONS				
	DEMO EXAMPLE ADD USE	RS / TEAMS		•	
MPLE	1 Please select Users / Tear	ns from the lists below.			
PERMISSIONS	USERS				
	SEARCH		Q	KEY	
	USERNAME [▲]	FIRST NAME	LAST NAME		
	🗹 althea	Althea	Bully		
	austin78	Austin	Texas	_	
	□ gdoge	Gerry	Doge		
ES HOSTS	🗹 jdoge	Josie	Doge		
	🗆 jgarcia	Jerry	Garcia		
NAME 🔺	< 1 2 > PAGE 1 OF 2		ITEMS	1 - 5 OF 6	
Database Servers	2 Please assign roles to the	selected users/teams		KEY	
DEMO EXAMPLE	Althea Bully USER	× Update		×	
Demo Inventory	-				
King PLC	Josie Doge USER	× Use		×	
	Production Operatio TEAM	× Admin		×	
			CANCEL	SAVE	

5. Click **Save** when done, and the Add Users/Teams window closes to display the updated roles assigned for each user and team.

USER *	ROLE	TEAM ROLES
admin	SYSTEM ADMINISTRATOR	
althea	× AD HOC SYSTEM AUDITOR X USE	
jdoge	× UPDATE × USE	
mags3707	SYSTEM ADMINISTRATOR	× AD HOC 알 × ADMIN 알 × USE 알
yser	SYSTEM AUDITOR	

To remove Permissions for a particular user, click the Disassociate (x) button next to its resource.

USER ^	ROLE	TEAM ROLES
		TDAM ROLES
admin	SYSTEM ADMINISTRATOR	
althea	* AD HOC SYSTEM AUDITOR X USE	
jdoge	× UPDATE × USE	
mags3707	SYSTEM ADMINISTRATOR	X AD HOC W X ADMIN W X USE W
yser	SYSTEM AUDITOR	
		ITEMS 1-5

This launches a confirmation dialog, asking you to confirm the disassociation.

TEMS 1-5

TEAM ACCESS REMOVAL			0
Please confirm that you would like to remove PRODUCTION OPERATIONS . This will affect a like to only remove access for this particular team.	all members o	f the team. If you would	d
	CANCEL	REMOVE TEAM ACCE	SS

14.2.2 Add Groups

Inventories are divided into groups, which may contain hosts and other groups, and hosts. Groups are only applicable to standard inventories and is not a configurable directly through a Smart Inventory. You can associate an existing group through host(s) that are used with standard inventories. There are several actions available for standard inventories:

- Create a new Group
- · Create a new Host
- Run a command on the selected Inventory
- Edit Inventory properties
- View activity streams for Groups and Hosts
- Obtain help building your Inventory

Note: Starting in Ansible Tower 3.2, inventory sources are no longer associated with groups. Prior versions, spawned groups and hosts would be children of our inventory source group. Now, spawned groups are top-level. These groups may still have child groups, and all of these spawned groups may have hosts.

To create a new group for an inventory:

1. Click the

button to open the **Create Group** window.

CREATE GROUP		8
DETAILS GROUPS HOSTS		
*NAME	DESCRIPTION	
VARIABLES @ YAML JSON		
1		
		CANCEL SAVE

- 2. Enter the appropriate details into the required and optional fields:
- Name: Required
- **Description**: Enter an arbitrary description as appropriate (optional)

- Variables: Enter definitions and values to be applied to all hosts in this group. Enter variables using either JSON or YAML syntax. Use the radio button to toggle between the two.
- 3. When done, click Save.

Add groups within groups

To add groups within groups:

1. Click the **Groups** tab.



- 2. Click the button, and select whether to add a group that already exists in your configuration or create a new group.
- 3. If creating a new group, enter the appropriate details into the required and optional fields:
- Name: Required
- **Description**: Enter an arbitrary description as appropriate (optional)
- Variables: Enter definitions and values to be applied to all hosts in this group. Enter variables using either JSON or YAML syntax. Use the radio button to toggle between the two.
- 4. When done, click Save.

The **Create Group** window closes and the newly created group displays as an entry in the list of groups associated with the group that it was created for.

CMS Web Group		0
DETAILS GROUPS HOSTS		
SEARCH	Q KEY	RUN COMMANDS +
GROUPS A		ACTIONS
		ð ×
		ITEMS 1 - 1

If you chose to add an existing group, available groups will appear in a separate selection window.

SELECT	GROUPS		0
SEAR	СН	Q	KEY
	GROUPS 🔶		
	Subgroup		
			ITEMS 1-1
		CANCEL	SAVE

Once a group is selected, it displays as an entry in the list of groups associated with the group.

5. To configure additional groups and hosts under the subgroup, click on the name of the subgroup from the list of groups and repeat the same steps described in this section.

CMS Web Group		8
DETAILS GROUPS HOSTS		
SEARCH	Q KEY	RUN COMMANDS +
GROUPS *		ACTIONS
Subgroup		ð ×
		ITEMS 1-1

14.2.3 Add hosts

You can configure hosts for the inventory as well as for groups and groups within groups. To configure hosts:

- 1. Click the **Hosts** tab.
- 2. Click the button, and select whether to add a host that already exists in your configuration or create a new host.
- 3. If creating a new host, select the button to specify whether or not to include this host while running jobs.
- 4. Enter the appropriate details into the required and optional fields:
- Host Name: Required
- Description: Enter an arbitrary description as appropriate (optional)
- Variables: Enter definitions and values to be applied to all hosts in this group. Enter variables using either JSON or YAML syntax. Use the radio button to toggle between the two.

5. When done, click Save.

The **Create Host** window closes and the newly created host displays as an entry in the list of hosts associated with the group that it was created for.

Subgroup		8
DETAILS GROUPS HOSTS		
SEARCH	QKEY	RUN COMMANDS +
HOSTS A		ACTIONS
🗆 💽 🔿 Web Host		d ^{ar} X
		ITEMS 1 - 1
Demo Inventory		8
DETAILS PERMISSIONS GROUPS	HOSTS SOURCES COMPLETED JOBS	
SEARCH	Q KEY	RUN COMMANDS +
GROUPS [▲]		ACTIONS
CMS Web Group		#* m
		ITEMS 1 - 1

If you chose to add an existing host, available hosts will appear in a separate selection window.

SELECT	HOSTS			8
SEAR	СН		Q	KEY
	HOSTS 🔺			
	Web Host			
				ITEMS 1-1
			CANCEL	SAVE

Once a host is selected, it displays as an entry in the list of hosts associated with the group.

6. To configure facts and additional groups for the host, click on the name of the host from the list of hosts.

New Inventory		0
DETAILS PERMISSIONS GROUPS HOST	S SOURCES COMPLETED JOBS	
SEARCH	Q KEY	RUN COMMANDS +
HOSTS A	RELATED GROUPS	ACTIONS
ON O Web Host	× CMS Web Group	▲ 6
		ITEMS 1-1

This opens the Details tab of the selected host.

Web Host ON		0
DETAILS FACTS GROUPS		
*HOST NAME @	DESCRIPTION	
Web Host	Host for CMS Web	
VARIABLES @ YAML JSON		
1		
		CANCEL

- 7. Click the **Facts** tab to input facts you want to gather. Refer to the *Fact Caching* section for more information about facts.
- 8. Click the **Groups** tab to configure groups for the host.



a. Click the **button** to associate the host with an existing group.

Available groups appear in a separate selection window.

SELEC	T GROUPS	8
SEAF	RCH	QKEY
	GROUPS 🔶	
	CMS Web Group	
	Subgroup	
		ITEMS 1-2
		CANCEL SAVE

b. Click to select the group(s) to associate with the host and click Save.

Once a group is associated, it displays as an entry in the list of groups associated with the host.

14.2.4 Add source

Inventory sources are no longer associated with groups. Prior to Ansible Tower 3.2, spawned groups and hosts would be children of our inventory source group. Now, spawned groups are top-level. These groups may still have child groups, and all of these spawned groups may have hosts.

Adding a source to an inventory only applies to standard inventories. Smart inventories inherit their source from the standard inventories they are associated with. To configure the source for the inventory:

1. In the inventory you want to add a source, click the Sources tab.



This opens the Create Source window.

INVENTORIES / Demo Inventory / SOURCES / CREATE INVENTORY SOURCE			0
CREATE SOURCE DETAILS			8
• NAME	DESCRIPTION	SOURCE Choose a source	CANCEL SAVE

- 3. Enter the appropriate details into the required and optional fields:
- Name: Required
- **Description**: Enter an arbitrary description as appropriate (optional)
- **Source**: Choose a source for your inventory. Refer to the *Inventory Sources* section for more information about each source and details for entering the appropriate information.

Note: Starting with Ansible Tower version 3.2, support for Rackspace Cloud Servers was discontinued.

- 4. You can configure the level of output on any inventory source's update jobs by selecting the appropriate option from the **Verbosity** drop-down menu.
- 5. All cloud inventory sources have the following update options:
- Overwrite: Refer to the on-screen tooltip () for information. In order to guarantee consistent behavior after 3.2 migration, do not set to True.
- Overwrite Variables: Refer to the on-screen tooltip () for information.
- Update on Launch: Each time a job runs using this inventory, refresh the inventory from the selected source before executing job tasks. To avoid job overflows if jobs are spawned faster than the inventory can sync, selecting this allows you to configure a Cache Timeout to cache prior inventory syncs for a certain number of seconds.

The "Update on Launch" setting refers to a dependency system for projects and inventory, and it will not specifically exclude two jobs from running at the same time. If a cache timeout is specified, then the dependencies for the second job is created and it uses the project and inventory update that the first job spawned. Both jobs then wait for that project and/or inventory update to finish before proceeding. If they are different job templates, they can then both start and run at the same time, if the system has the capacity to do so.

Note: If you intend to use Tower's provisioning callback feature with a dynamic inventory source, "Update on Launch" should be set for the inventory group.

6. Review your entries and selections and click Save when done.

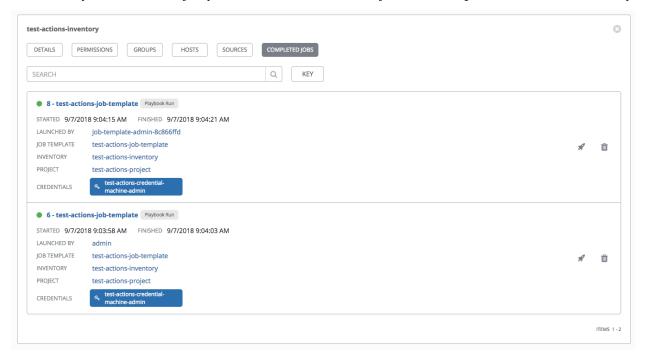
Once a source is defined, it displays as an entry in the list of sources associated with the inventory. From the **Sources** tab you can perform a sync on a single source, or sync all of them at once. You can also perform additional actions such as scheduling a sync process, and edit or delete the source.

external-org-inventory		0
DETAILS PERMISSIONS GROUPS HOSTS	SOURCES COMPLETED JOBS	
SEARCH	QKEY	SYNC ALL +
SOURCES [▲]	TYPE 🗢	ACTIONS
external-org-inventory-source-custom	Custom Script	Ø 2 î
		ITEMS 1 · 1

- 7. To configure notifications for the source, click the Notifications tab.
- a. If notifications are already set up, select a notification preference.
- b. if notifications have not been set up, refer to Notifications for more information.

14.2.5 View completed jobs

If an inventory was used to run a job, you can view details about those jobs in the **Completed Jobs** tab of the inventory.



Smart Host Filter

You can use a search filter to populate hosts for an inventory. This feature was introduced in Ansible Tower 3.2 utilizing the capability of the fact searching feature.

Facts generated by an Ansible playbook during a Job Template run are stored by Tower into the database whenever use_fact_cache=True is set per-Job Template. New facts are merged with existing facts and are per-host. These stored facts can be used to filter hosts via the /api/v2/hosts endpoint, using the GET query parameter host_filter For example: /api/v2/hosts? host_filter=ansible_facts_ansible_processor_vcpus=8

The host_filter parameter allows for:

- grouping via ()
- use of the boolean and operator:
 - _____ to reference related fields in relational fields
 - _____ is used on ansible_facts to separate keys in a JSON key path
 - [] is used to denote a json array in the path specification
 - "" can be used in the value when spaces are wanted in the value
- "classic" Django queries may be embedded in the host_filter

Examples:

```
/api/v2/hosts/?host_filter=name=localhost
/api/v2/hosts/?host_filter=ansible_facts__ansible_date_time__weekday_number="3"
/api/v2/hosts/?host_filter=ansible_facts__ansible_processor[]="GenuineIntel"
/api/v2/hosts/?host_filter=ansible_facts__ansible_lo__ipv6[]__scope="host"
/api/v2/hosts/?host_filter=ansible_facts__ansible_processor_vcpus=8
/api/v2/hosts/?host_filter=ansible_facts__ansible_env__PYTHONUNBUFFERED="true"
/api/v2/hosts/?host_filter=(name=localhost or name=database) and (groups__name=east__
or groups__name="west coast") and ansible_facts__an
```

Inventory Sources

Choose a source which matches the inventory type against which a host can be entered:

- Sourced from a Project
- Amazon Web Services EC2
- Google Compute Engine
- Microsoft Azure Classic (deprecated)
- Microsoft Azure Resource Manager
- VMware vCenter
- Red Hat Satellite 6
- Red Hat CloudForms
- OpenStack
- Red Hat Virtualization

- Ansible Tower
- Custom Script

Sourced from a Project

An inventory that is sourced from a project means that is uses the SCM type from the project it is tied to. For example, if the project's source is from GitHub, or a Red Hat Insights project, then the inventory will use the same source.

- 1. To configure a project-sourced inventory, select Sourced from a Project from the Source field.
- 2. The Create Source window expands with additional fields. Enter the following details:
 - Credential: Optionally specify the credential to use for this source.
 - **Project**: Required. Specify the project this inventory is using as its source. Click the button to choose from a list of projects. If the list is extensive, use the search to narrow the options.
 - **Inventory File**: Required. Select an inventory file associated with the sourced project. If not already populated, you can type it into the text field within the drop down menu to filter the extraneous file types. In addition to a flat file inventory, you can point to a directory or an inventory script.

1	INVENTORY FILE 🔞	
	Choose an inventory file	
	1	
	/ (project root)	

- 3. In addition to the update options available for cloud inventory sources, you can specify whether or not to update on project changes. Check the **Update on Project Change** option to refresh the inventory from the selected source after every project update where the SCM revision changes before executing job tasks.
- 4. In order to pass to the custom inventory script, you can optionally set environment variables in the **Environment Variables** field.

CREATE SOURCE DETAILS NOTIFICATIONS			8
* NAME Source from Project	DESCRIPTION	* SOURCE Sourced from a Project *	
SOURCE DETAILS CREDENTIAL Q VERBOSITY @ 1 (INFO) VAML_JSON	PROJECT Another Project UPDATE OPTIONS Overwrite @ Voerwrite Variables @ Update on Froject Change @	INVENTORY FILE Inventories/Inventory.ini	
1		CANCEL	SAVE

Amazon Web Services EC2

- 1. To configure an AWS EC2-sourced inventory, select Amazon EC2 from the Source field.
- 2. The Create Source window expands with additional fields. Enter the following details:
 - **Credential**: Optionally choose from an existing credential (for more information, refer to *Credentials*).

If Tower is running on an EC2 instance with an assigned IAM Role, the credential may be omitted, and the security credentials from the instance metadata will be used instead. For more information on using IAM Roles, refer to the IAM_Roles_for_Amazon_EC2_documentation_at_Amazon.

- **Regions**: Click on the regions field to see a list of regions for your cloud provider. You can select multiple regions, or choose "All" to include all regions. Tower will only be updated with Hosts associated with the selected regions.
- **Instance Filters**: Rather than importing your entire Amazon EC2 inventory, filter the instances returned by the inventory script based on a variety of metadata. Hosts are imported if they match any of the filters entered here.

Examples:

- To limit to hosts having the tag TowerManaged: Enter tag-key=TowerManaged
- To limit to hosts using either the key-name staging or production: Enter key-name=staging, key-name=production
- To limit to hosts where the Name tag begins with test: Enter tag:Name=test*

For more information on the filters that can be used here, refer to the Describe Instances documentation at Amazon.

- Only Group By: By default, Tower creates groups based on the following Amazon EC2 parameters:
 - Availability Zones
 - Image ID
 - Instance ID
 - Instance Type

- Key Name
- Region
- Security Group
- Tags (by name)
- VPC ID
- Tag None

If you do not want all these groups created, select from the dropdown the list of groups that you would like created by default. You can also select Instance ID to create groups based on the Instance ID of your instances.

3. Use the **Source Variables** field to override variables found in ec2.ini and used by the inventory update script. Enter variables using either JSON or YAML syntax. Use the radio button to toggle between the two. For a detailed description of these variables view ec2.ini in the Ansible GitHub repo.

CREATE SOURCE			Θ
DETAILS NOTIFICATIONS			
* NAME	DESCRIPTION	* SOURCE	
Sourced from AWS		Amazon EC2 🔹	
SOURCE DETAILS			
CREDENTIAL	REGIONS Ø	INSTANCE FILTERS	
٩	× US West (Northern California)		
ONLY GROUP BY 🔞	VERBOSITY @	UPDATE OPTIONS	
× Region	1 (INFO) *	Overwrite Overwrite Variables Overwrite Variable	
		Update on Launch @	
CACHE TIMEOUT (SECONDS)			
10 🗘			
SOURCE VARIABLES @ YAML JSON			
1			
		CANCEL SA	AVE

Google Compute Engine

- 1. To configure a Google-sourced inventory, select Google Compute Engine from the Source field.
- 2. The Create Source window expands with additional fields. Enter the following details:
- Credential: Required. Choose from an existing Credential. For more information, refer to Credentials.
- **Regions**: Click on the regions field to see a list of regions for your cloud provider. You can select multiple regions, or choose "All" to include all regions. Tower will only be updated with Hosts associated with the selected regions.

CREATE SOURCE DETAILS NOTIFICATIONS		٥
* NAME Source from GCE	DESCRIPTION	* SOURCE Google Compute Engine v
SOURCE DETAILS *CREDENTIAL Q Inventory Credential	REGIONS 🛛	VERBOSITY
UPDATE OPTIONS ○ Overwrite @ ○ Overwrite Variables @ ○ Update on Launch @		
		CANCEL SAVE

Microsoft Azure Classic (deprecated)

- 1. To configure a Azure-sourced inventory, select Microsoft Azure Classic (deprecated) from the Source field.
- 2. The Create Source window expands with additional fields. Enter the following details:
- Credential: Required. Choose from an existing Credential. For more information, refer to Credentials.
- **Regions**: Click on the regions field to see a list of regions for your cloud provider. You can select multiple regions, or choose "All" to include all regions. Tower will only be updated with Hosts associated with the selected regions.

CREATE SOURCE		e	8
DETAILS NOTIFICATIONS			
* NAME	DESCRIPTION	* SOURCE	
Source from Azure Classic		Microsoft Azure Classic (deprecated)	
SOURCE DETAILS			
* CREDENTIAL	REGIONS @	VERBOSITY 💿	
Q Inventory Credential	× US East	1 (INFO) *	
UPDATE OPTIONS Overwrite @ Overwrite Variables @ Update on Launch @		CANCEL SAVE	

Microsoft Azure Resource Manager

- To configure a Azure Resource Manager-sourced inventory, select Microsoft Azure Resource Manager from the Source field.
- 2. The Create Source window expands with additional fields. Enter the following details:
- Credential: Required. Choose from an existing Credential. For more information, refer to Credentials.
- **Regions**: Click on the regions field to see a list of regions for your cloud provider. You can select multiple regions, or choose "All" to include all regions. Tower will only be updated with Hosts associated with the selected regions.

CREATE SOURCE DETAILS NOTIFICATIONS			8
NAME Source from Azure RM		SOURCE Microsoft Azure Resource Manager	
CREDENTIAL Q Inventory Credential	REGIONS 🛛	VERBOSITY	
UPDATE OPTIONS Overwrite @ Overwrite Variables @ Update on Launch @			
		CANCEL	AVE

VMware vCenter

- 1. To configure a VMWare-sourced inventory, select VMware vCenter from the Source field.
- 2. The Create Source window expands with additional fields. Enter the following details:
 - **Credential**: Required. Choose from an existing credential (for more information, refer to *Credentials*).
 - **Instance Filters**: Rather than importing your entire VMWare inventory, filter the instances returned by the inventory script based on a variety of metadata. Hosts are imported if they match any of the filters entered here.

For more information on the filters that can be used here, refer to the Quick Filters Available for vSphere Objects documentation at VMware.

- Only Group By: By default, Tower creates groups based on user-specified VMWare parameters. For example, enter Instance ID to create groups based on the Instance ID of your instances.
- 3. Use the **Source Variables** field to override variables found in vmware.ini and used by the inventory update script. Enter variables using either JSON or YAML syntax. Use the radio button to toggle between the two. For a detailed description of these variables view vmware_inventory.ini in the Ansible GitHub repo <vmware_inventory.ini inventory script.

Note: The inventory script for VMware was updated in Ansible Tower 3.1.2 to allow configuration of the host_filters or groupby_patterns parameter. Specify those values in the **Source Variables** text field of the Create Group screen or Edit Group screen. For example:

: "{{ config.guestid == 'rhel7_64Guest' }}"	
erns: "{{ guest.guestid }},{{ 'templates' if config.template else '	quests'}}"
erns: "{{ guest.guestid }},{{ 'templates' if config.template else '	guests'}

CREATE SOURCE DETAILS NOTIFICATIONS			8
NAME Source from VMware	DESCRIPTION	*SOURCE VMware vCenter *	
SOURCE DETAILS * CREDENTIAL Q Inventory Credential		ONLY GROUP BY	
VERBOSITY	UPDATE OPTIONS Overwrite @ Overwrite Variables @ 2 Update on Launch @	CACHE TIMEOUT (SECONDS)	
SOURCE VARIABLES YAML JSON			
		CANCEL SAVE	2

Red Hat Satellite 6

- 1. To configure a Red Hat Satellite-sourced inventory, select Red Hat Satellite from the Source field.
- 2. The Create Source window expands with additional fields.
- Credential: Required. Choose from an existing credential (for more information, refer to Credentials).
- Use the Source Variables field to override variables found in foreman.ini and used by the inventory update script.

Note: The variable want_facts from foreman.ini is hard-coded to True and cannot be overridden at this time. If you want to set the group_patterns, group_prefix, or want_hostcollections variables, prefix them with satellite6, e.g.: satellite6_group_prefix: myprefix

Enter variables using either JSON or YAML syntax. Use the radio button to toggle between the two. For a detailed description of these variables view foreman.ini in the Ansible GitHub repo.

CREATE SOURCE			Θ
DETAILS NOTIFICATIONS			
*NAME	DESCRIPTION	* SOURCE	
Source from RH Satellite 6		Red Hat Satellite 6	
SOURCE DETAILS			
* CREDENTIAL	VERBOSITY @	UPDATE OPTIONS	
Q Inventory Credential	1 (INFO) *	Overwrite Overwrite Variables Overwrite Variables	
		Update on Launch @	
SOURCE VARIABLES @ YAML JSON			
1			
		CANCEL	E

Red Hat CloudForms

- 1. To configure a Red Hat CloudForms-sourced inventory, select Red Hat CloudForms from the Source field.
- 2. The Create Source window expands with additional fields. Enter the following details:
- Credential: Required. Choose from an existing credential (for more information, refer to Credentials).
- Use the **Source Variables** field to override variables found in cloudforms.ini and used by the inventory update script. Enter variables using either JSON or YAML syntax. Use the radio button to toggle between the two. For a detailed description of these variables view cloudforms.ini in the Ansible GitHub repo.

CREATE SOURCE			8
DETAILS NOTIFICATIONS			
*NAME	DESCRIPTION	*SOURCE	
Source from RH CloudForms		Red Hat CloudForms 🔹	
SOURCE DETAILS			
* CREDENTIAL	VERBOSITY @	UPDATE OPTIONS	
Q Inventory Credential	1 (INFO) *	Overwrite Overwrite Variables Overwrite Variable	
		Update on Launch @	
SOURCE VARIABLES @ YAML JSON			
1			
		CANCEL	AVE

OpenStack

- 1. To configure an OpenStack-sourced inventory, select OpenStack from the Source field.
- 2. The Create Source window expands with additional fields. Enter the following details:
- Credential: Required. Choose from an existing credential (for more information, refer to Credentials).
- Use the **Source Variables** field to override variables found in openstack.yml and used by the inventory update script. Enter variables using either JSON or YAML syntax. Use the radio button to toggle between the two. For a detailed description of these variables view openstack.yml in the Ansible GitHub repo.

CREATE SOURCE			0
DETAILS NOTIFICATIONS			
* NAME	DESCRIPTION	*SOURCE	
Source from OpenStack		OpenStack 👻	
SOURCE DETAILS			
* CREDENTIAL	VERBOSITY @	UPDATE OPTIONS	
Q Inventory Credential	1 (INFO) *	Overwrite @ Overwrite Variables @	
		Update on Launch @	
SOURCE VARIABLES @ YAML JSON			
1			
		CANCEL	AVE

Red Hat Virtualization

- 1. To configure a Red Hat Virtualization-sourced inventory, select Red Hat Virtualization from the Source field.
- 2. The Create Source window expands with additional fields. The **Credential** is required. Choose from an existing credential (for more information, refer to *Credentials*).

CREATE SOURCE			Θ
DETAILS NOTIFICATIONS			
*NAME	DESCRIPTION	*SOURCE	
Source from RH Virtualization		Red Hat Virtualization *	
SOURCE DETAILS			
* CREDENTIAL	VERBOSITY 🔞	UPDATE OPTIONS	
Q RHV Credential	1 (INFO) *	Overwrite Overwrite Variables Overwrite Variable	
		Update on Launch 😡	
SOURCE VARIABLES @ YAML JSON			
1			
			_
		CANCEL	SAVE

Ansible Tower

- 1. To configure a Ansible Tower-sourced inventory, select Ansible Tower from the Source field.
- 2. The Create Source window expands with additional fields. Enter the following details:
 - **Credential**: Required. Choose from an existing credential (for more information, refer to *Credentials*).
 - **Instance Filters**: Rather than importing your entire Tower inventory, filter the instances by an inventory ID/name; then the inventory script would return that inventory from the other Tower instance.

Tower Inventory Source			0
DETAILS NOTIFICATIONS			
* NAME	DESCRIPTION	* SOURCE	
Tower Inventory Source		Ansible Tower +	
SOURCE DETAILS			
* CREDENTIAL	INSTANCE FILTERS 🚱	VERBOSITY @	
Q TowerCred	1011	1 (INFO) *	
UPDATE OPTIONS			
Overwrite @ Overwrite Variables @ Update on Launch @			
		CANCEL	SAVE

Custom Script

Tower allows you to use a custom dynamic inventory script, if your administrator has added one.

- 1. To configure a Custom Script-sourced inventory, select Custom Script from the Source field.
- 2. The Create Source window expands with additional fields. Enter the following details:
- **Credential**: You can optionally provide a credential for custom sources. The kind of credential is limited to cloud and network. Refer to *Custom Credential Types* for more information.
- **Custom Inventory Script**: Required. Choose from an existing Inventory Script (for more information, refer to Custom Inventory Scripts).
- Environment Variables: Set variables in the environment to be used by the inventory update script. The variables would be specific to the script that you have written. Enter variables using either JSON or YAML syntax. Use the radio button to toggle between the two.

CREATE SOURCE			0
DETAILS NOTIFICATIONS			
* NAME	DESCRIPTION	* SOURCE	
Source from Custom Script		Custom Script	•
SOURCE DETAILS			
CREDENTIAL	* CUSTOM INVENTORY SCRIPT	VERBOSITY 🚱	
Q Cloud credential	Q Large Inventory Script	1 (INFO)	•
UPDATE OPTIONS	CACHE TIMEOUT (SECONDS) 🕖		
Overwrite Overwrite Variables Overwrite Variable	30 🗘		
Update on Launch @			
ENVIRONMENT VARIABLES @ YAML JSON			
1			
-			CANCEL

For more information on syncing or using custom inventory scripts, refer to Custom Inventory Scripts in the Ansible Tower Administration Guide.

14.3 Running Ad Hoc Commands

To run an ad hoc command:

1. Select an inventory source from the list of hosts or groups. The inventory source can be a single group or host, a selection of multiple hosts, or a selection of multiple groups.

lbgroup		6
DETAILS GROUPS HOSTS		
SEARCH	Q	RUN COMMANDS +
HOSTS A		ACTIONS
🗆 💽 🔿 Web Host 🖌		Ø X
		ITEMS 1
emo Inventory		
DETAILS PERMISSIONS GROUPS	HOSTS SOURCES COMPLETED JOBS	
	HOSTS SOURCES COMPLETED JOBS	
DETAILS PERMISSIONS GROUPS		RUN COMMANDS
DETAILS PERMISSIONS GROUPS		RUN COMMANDS

2. Click the RUN COMMANDS

The Execute Command window opens.

EXECUTE COMMAND			Θ
*MODULE @	ARGUMENTS @	LIMIT @	
Choose a module *		Web Host	
*MACHINE CREDENTIAL @	*VERBOSITY @	FORKS @	
Q Demo Credential	0 (Normal) -	DEFAULT	
SHOW CHANGES 🔞			
OFF	ENABLE PRIVILEGE ESCALATION		
EXTRA VARIABLES @ YAML JSON			
1			
		RESET	

- 3. Enter the details for the following fields:
- Module: Select one of the modules that Tower supports running commands against.

command	apt_repository	mount	win_service
shell	apt_rpm	ping	win_updates
yum	service	selinux	win_group
apt	group	setup	win_user
apt_key	user	win_ping	

• Arguments: Provide arguments to be used with the module you selected.

button.

• Limit: Enter the limit used to target hosts in the inventory. To target all hosts in the inventory enter all or *, or leave the field blank. This is automatically populated with whatever was selected in the previous view prior to clicking the launch button.

- Machine Credential: Select the credential to use when accessing the remote hosts to run the command. Choose the credential containing the username and SSH key or password that Ansbile needs to log into the remote hosts.
- Verbosity: Select a verbosity level for the standard output.
- Forks: If needed, select the number of parallel or simultaneous processes to use while executing the command.
- Show Changes: Select to enable the display of Ansible changes in the standard output. The default is OFF.
- Enable Privilege Escalation: If enabled, the playbook is run with administrator privileges. This is the equivalent of passing the --become option to the ansible command.
- Extra Variables: Provide extra command line variables to be applied when running this inventory. Enter variables using either JSON or YAML syntax. Use the radio button to toggle between the two.

RUN COMMAND			0
*MODULE @	ARGUMENTS 🛛	LIMIT @ Web Host	
*MACHINE CREDENTIAL @ Q Demo Credential		FORKS @	
SHOW CHANGES O ON EXTRA VARIABLES O YAML JSON	□ ENABLE PRIVILEGE ESCALATION ●		
			н



button to run this ad hoc command.

The results display in the Job Results and Standard Out window.

RESULTS		A	ŵ
NAME	ping		
STATUS	Successful		
STARTED	11/14/2017 10:00:44 AM		
FINISHED	11/14/2017 10:00:47 AM		
ELAPSED	2.991 seconds		
INVENTORY	Demo Inventory		
CREDENTIAL	Demo Credential		
LAUNCHED BY	admin		
FORKS	0		
LIMIT	localhost		
VERBOSITY	0		
EXTRA VARIABLES @	1		

CHAPTER

FIFTEEN

JOB TEMPLATES

A job template is a definition and set of parameters for running an Ansible job. Job templates are useful to execute the same job many times. Job templates also encourage the reuse of Ansible playbook content and collaboration between teams. While the REST API allows for the execution of jobs directly, Tower requires that you first create a job template.



The () menu opens a list of the job templates that are currently available. The job template list is sorted alphabetically by name but you can search by various fields and attributes of the job template. The job template list also enables you to launch, copy, and remove a job template. Before deleting a job template, be sure it is not used in a workflow job template.

TEMPLATES				
EMPLATES				
TEMPLATES				
SEARCH	Q KEY			
Demo Job Tem	plate job Template			
INVENTORY	Demo Inventory			
PROJECT	Demo Project	A	2	
CREDENTIALS	🔍 Demo Credential			
LAST MODIFIED	10/3/2018 2:17:40 PM by admin			
Example Job To	nçiate			
ACTIVITY				
INVENTORY	New inventory detail			
PROJECT	Project from Git	4	2	
CREDENTIALS	4. Demo Credential	39	42	
LAST MODIFIED	10/10/2018 3:29:14 PM by admin			
LAST RAN	10/10/2018 9:41:03 AM			
LABELS	Test			
New Template	with Dependencies Job Template			
ACTIVITY				
INVENTORY	New inventory detail			
PROJECT	Project from Git	A	4	
CREDENTIALS	R Demo Credential			
LAST MODIFIED	10/10/2018 11:11:12 AM by admin			
LAST RAN	10/10/2018 11:11:12 AM			
WF in WF World	low Template	4	42	
LAST MODIFIED	10/8/2018 9:31:30 AM by admin	<i></i>	421	
Workflow usin	g JT Workflow Template	4	2	
LAST MODIFIED	10/5/2018 7:35:21 PM by admin		42	

Note: If deleting items that are used by other work items, a message opens listing the items are affected by the deletion and prompts you to confirm the deletion. Some screens will contain items that are invalid or previously deleted, so they will fail to run. Below is an example of such a message:

		💄 admin	0		
OJECTS					
PROJECTS 2	DELETE PROJECT FROM GIT The project is currently being used by other resources. Are you sure you want to delete this project? Job Templates Job Templates				ŧ
O Demo Project	CANCEL	J	Q	AC [*]	TIONS
Project from Git	Git b2cf1f0 🗋 10/5/2018 3:55:16 PM	Ø	C	4	Ŵ
					ITEMS 1 - 2

Note: Job templates can be used to build a workflow template. Many parameters in a job template allow you to enable **Prompt on Launch** that can be modified at the workflow level, and do not affect the values assigned at the job template level. For instructions, see the *Workflow Visualizer* section.

15.1 Create a Job Template

To create a new job template:



button then select Job Template from the menu list.

TEMPLATES / CREATE JOB TEMPLATE			0
NEW JOB TEMPLATE DETAILS PERMISSIONS NOTIFICATIONS COMPLETED JOBS SI			6
* NAME	DESCRIPTION	JOB TYPE PROMPT ON LAUNCH Run T	
INVENTORY PROMPT ON LAUNCH	◆ PROJECT ●	 ▶ PLAYBOOK ● Choose a playbook 	
	Forks •		
* VERBOSITY	JOB TAGS 🛛 📄 PROMPT ON LAUNCH	SKIP TAGS 🛛 📄 PROMPT ON LAUNCH	
		JOB SLICING O	
SHOW CHANGES O	OPTIONS Enable Privilege Escalation @ Allow Provisioning Callback: @ Enable Concurrent Jobs @ Use Fact Cache @		
EXTRA VARIABLES 😧 YAML JSON		PROMPT ON	I LAUNCH
1			
		CANCEL	

- 2. Enter the appropriate details into the following fields:
- Name: Enter a name for the job.

- Description: Enter an arbitrary description as appropriate (optional).
- Job Type:
 - Run: Execute the playbook when launched, running Ansible tasks on the selected hosts.
 - **Check**: Perform a "dry run" of the playbook and report changes that would be made without actually making them. Tasks that do not support check mode will be skipped and will not report potential changes.
 - **Prompt on Launch** If selected, even if a default value is supplied, you will be prompted upon launch to choose a job type of run or check.

Note: More information on job types can be found in the Playbooks: Special Topics section of the Ansible documentation.

- **Inventory**: Choose the inventory to be used with this job template from the inventories available to the currently logged in Tower user.
- **Prompt on Launch**: If selected, even if a default value is supplied, you will be prompted upon launch to choose an inventory to run this job template against.
- **Project**: Choose the project to be used with this job template from the projects available to the currently logged in Tower user.
- **Playbook**: Choose the playbook to be launched with this job template from the available playbooks. This menu is automatically populated with the names of the playbooks found in the project base path for the selected project. For example, a playbook named "jboss.yml" in the project path appears in the menu as "jboss".
- **Credential**: Click the button to open a separate window. Choose the credential from the available options to be used with this job template. Use the drop-down menu list to filter by credential type if the list is extensive.

CREATE JOB TEMPLATE			
TEMPLATE	CREDENTIALS	0	
PERMISSIONS	CREDENTIAL TYPE:	Machine 🔺	
PERMISSIONS	CEADCH	Amazon Web Services	
	SEARCH	Ansible Tower	* JOB TYPE 🕜
	NAME 🔶	Google Compute Engine	Run
RY 😧	 Demo Credential 	Microsoft Azure Resource Manager	* PLAYBOOK 😧
no Inventory		OpenStack	Choose a playbook
. 0		Red Hat CloudForms CANCEL SELECT	
YØ			JOB TAGS 😡

• **Prompt on Launch**: If selected, upon launching a job template that has a default machine credential, you will not be able to remove the default machine credential in the Prompt dialog without replacing it with another machine credential before it can launch. Alternatively, you can add more credentials as you see fit. Below is an example of such a message:

INVENTORY	Demo Inventory				
PROJECT	Demo Project	PROMPT	R	4	Û
CREDENTIALS	৭ Demo Creden				
LAST MODIFIED	10/3/2018 2:17:	CREDENTIAL PREVIEW			
Example Job Te		SELECTED No credentials selected REVERT			
ACTIVITY					
INVENTORY	New inventory c	A This job template has a default Machine credential which must be included or replaced before proceeding.	st.	2	Û
PROJECT	Project from Git	proceeding.			
LAST MODIFIED	10/10/2018 9:41	Credential Type: Machine 👻			
LAST RAN	10/10/2018 9:41				
New Template	with Dependenci	SEARCH Q. KEY			
INVENTORY	New inventory c	NAME *			
PROJECT	Project from Git	O Demo Credential	A.	4	Û
CREDENTIALS	ି ୧ Demo Creden				
LAST MODIFIED	10/10/2018 10:4	O Machine credential			
WF in WF Work	flow Template	ITEMS 1 - 2			
LAST MODIFIED	10/8/2018 9:31:	CANCEL	A	æ	Û
Workflow usin	g JT Workflow Template				
			R	2	Û
LAST MODIFIED	10/5/2018 7:35:2	1 PM by admin			

- Forks: The number of parallel or simultaneous processes to use while executing the playbook. A value of zero uses the Ansible default setting, which is 5 parallel processes unless overridden in /etc/ansible/ansible.cfg.
- Limit: A host pattern to further constrain the list of hosts managed or affected by the playbook. Multiple patterns can be separated by colons (":"). As with core Ansible, "a:b" means "in group a or b", "a:b:&c" means "in a or b but must be in c", and "a:!b" means "in a, and definitely not in b".
- **Prompt on Launch**: If selected, even if a default value is supplied, you will be prompted upon launch to choose a limit.

Note: For more information and examples refer to Patterns in the Ansible documentation.

• Verbosity: Control the level of output Ansible produces as the playbook executes. Set the verbosity to any of Default, Verbose, or Debug. This only appears in the "details" report view. Verbose logging includes the output of all commands. Debug logging is exceedingly verbose and includes information on SSH operations that can be useful in certain support instances. Most users do not need to see debug mode output.

Warning: Verbosity 5 causes Tower to block heavily when jobs are running, which could delay reporting that the job has finished (even though it has) and can cause the browser tab to lock up.

- **Prompt on Launch**: If selected, even if a default value is supplied, you will be prompted upon launch to choose a verbosity.
- **Job Tags**: Provide a comma-separated list of playbook tags to specify what parts of the playbooks should be executed. For more information and examples refer to Tags in the Ansible documentation.
- **Prompt on Launch**: If selected, even if a default value is supplied, you will be prompted upon launch to choose a job tag.
- **Skip Tags**: Provide a comma-separated list of playbook tags to skip certain tasks or parts of the playbooks to be executed. For more information and examples refer to Tags in the Ansible documentation.
- **Prompt on Launch**: If selected, even if a default value is supplied, you will be prompted upon launch to choose tag(s) to skip.
- Labels: Supply optional labels that describe this job template, such as "dev" or "test". Labels can be used to group and filter job templates and completed jobs in the Tower display.
- Labels are created when they are added to the Job Template. Labels are associated to a single Organization using the Project that is provided in the Job Template. Members of the Organization can create labels on a Job

Template if they have edit permissions (such as admin role).

- Once the Job Template is saved, the labels appear in the Job Templates overview.
- Click on the "x" beside a label to remove it. When a label is removed, and is no longer associated with a Job or Job Template, the label is permanently deleted from the list of Organization labels.
- Jobs inherit labels from the Job Template at the time of launch. If a label is deleted from a Job Template, it is also deleted from the Job.

LABELS 🕜				
× test × scan × run				
Example JOB TEMPLATE				
INVENTORY Demo Inventory				
PROJECT Demo Example	3	**	අත	ŵ
CREDENTIALS 4. DEMO CREDENTIAL	27		421	ш
LAST MODIFIED 2/9/2018 4:13:41 PM by admin				
LABELS run scan test				

- Instance Groups: Click the button to open a separate window. Choose the instance groups on which you want to run this job template. If the list is extensive, use the search to narrow the options.
- Job Slicing: Specify the number of slices you want this job template to run. Each slice will run the same tasks against a portion of the inventory. For more information about job slices, see *Job Slicing*.
- Show Changes: Allows you to see the changes made by Ansible tasks.
- **Prompt on Launch**: If selected, even if a default value is supplied, you will be prompted upon launch to choose whether or not to show changes.
- **Options**: Supply optional labels that describe this job template, such as "dev" or "test". Labels can be used to group and filter job templates and completed jobs in the Tower display.
- Enable Privilege Escalation: If enabled, run this playbook as an administrator. This is the equivalent of passing the --become option to the ansible-playbook command.
- Allow Provisioning Callbacks: Enable a host to call back to Tower via the Tower API and invoke the launch of a job from this job template. Refer to *Provisioning Callbacks* for additional information.
- Enable Concurrent Jobs: Allow jobs in the queue to run simultaneously if not dependent on one another. Check this box if you want to run job slices simultaneously. Refer to *Ansible Tower Capacity Determination and Job Impact* for additional information.
- Use Fact Cache: When enabled, Tower will activate an Ansible fact cache plugin for all hosts in an inventory related to the job running.
- Extra Variables:
 - Pass extra command line variables to the playbook. This is the "-e" or "-extra-vars" command line parameter for ansible-playbook that is documented in the Ansible documentation at Passing Variables on the Command Line.
 - Provide key/value pairs using either YAML or JSON. These variables have a maximum value of
 precedence and overrides other variables specified elsewhere. An example value might be:

```
git_branch: production
release_version: 1.5
```

For more information about extra variables, refer to Extra Variables.

• **Prompt on Launch**: If selected, even if a default value is supplied, you will be prompted upon launch to choose command line variables.

Note: If you want to be able to specify extra_vars on a schedule, you must select **Prompt on Launch** for **EXTRA VARIABLES** on the job template, or a enable a survey on the job template, then those answered survey questions become extra_vars.

3. When you have completed configuring the details of the job template, select Save.

Saving the template does not exit the job template page but remains on the Job Template Details view for further editing, if necessary. After saving the template, you can now proceed with adding more attributes about the template, such as permissions, notifications, view completed jobs, and add a survey (if the job type is not a scan).

Job template with slicing				G
DETAILS PERMISSIONS NOTIFICATIONS	COMPLETED JOBS SCH	ADD SURVEY		
* NAME		DESCRIPTION	* JOB TYPE 🛛	PROMPT ON LAUNCH
Job template with slicing			Run	•
* INVENTORY @	PROMPT ON LAUNCH	* PROJECT 🚱	* PLAYBOOK 🕖	
Q New inventory detail		Q Project from Git	free_waiter.yml	•
CREDENTIAL	PROMPT ON LAUNCH	FORKS 🚱	LIMIT 🕑	PROMPT ON LAUNCH
Q		DEFAULT		
* VERBOSITY 🕐	PROMPT ON LAUNCH	JOB TAGS 🚱 📄 PROMPT ON LAUNCH	SKIP TAGS 🕜	PROMPT ON LAUNCH
0 (Normal)	-			
LABELS Ø		INSTANCE GROUPS 🚱	JOB SLICING Ø	
		٩	2	* *
SHOW CHANGES 🕖	PROMPT ON LAUNCH	OPTIONS		
OFF		Enable Privilege Escalation Allow Provisioning Callbacks Yenable Concurrent Jobs Enable Concurrent Jobs Use Fact Cache		
EXTRA VARIABLES Ø YAML JSON				PROMPT ON LAUNCE
1 2 ansible_ssh_user: ubuntu 3 ansible_connection: local				
				CANCEL

You can verify the template is saved when the newly created template appears on the list of templates at the bottom of the screen.

ARCH	Q KEY			
)emo job Temj	JOB TEMPLATE			
NVENTORY	Demo Inventory			
ROJECT	Demo Project	a 🏥	2	Û
REDENTIALS	% DEMO CREDENTIAL			
AST MODIFIED	2/9/2018 10:58:18 AM by admin			
Example JOB	TEMPLATE			
INVENTORY	Demo Inventory			
PROJECT	Demo Example	A 66	<i>a</i> L	_
CREDENTIALS	4 DEMO CREDENTIAL	ar 🏥	4	Ú
LAST MODIFIED	2/9/2018 4:10:35 PM by admin			
	test			

15.2 Add Permissions

The **Permissions** tab allows you to review, grant, edit, and remove associated permissions for users as well as team members. To assign permissions to a particular user for this resource:

- 1. Click the **Permissions** tab.
- 2. Click the button to open the Add Users/Teams window.

/ DEMO EXAMPLE / PE	ERMISSIONS			
	DEMO EXAMPLE ADD USER	S / TEAMS	3	
MPLE	1 Please select Users / Team	is from the lists below.		
PERMISSIONS	USERS		_	
	SEARCH		Q KEY	
	USERNAME [▲]	FIRST NAME	LAST NAME 🗘	
	🗆 althea	Althea	Bully	
	austin78	Austin	Texas	
	□ gdoge	Gerry	Doge	
HOSTS	🗆 jdoge	Josie	Doge	
	🗆 jgarcia	Jerry	Garcia	
NAME 🔶	< 1 2 > PAGE 1 OF 2		ITEMS 1 - 5 OF 6	
Database Servers			CANCEL	
DEMO EXAMPLE			CANCEL SAVE	

- 3. Specify the users or teams that will have access then assign them specific roles:
 - a. Click to select one or multiple checkboxes beside the name(s) of the user(s) or team(s) to select them.

Note: You can select multiple users and teams at the same time by navigating between the **Users** and **Teams** tabs without saving.

After selections are made, the window expands to allow you to select a role from the drop-down menu list for each user or team you chose.

/ DEMO EXAMPLE / PE	ERMISSIONS		
	DEMO EXAMPLE ADD US	ERS / TEAMS	0
MPLE	1 Please select Users / Tea	ams from the lists below.	
PERMISSIONS	USERS TEAMS		
	SEARCH		Q KEY
	USERNAME [▲]	FIRST NAME 🗢	LAST NAME 🗢
	🗹 althea	Althea	Bully
	austin78	Austin	Texas
	gdoge	Gerry	Doge
ES HOSTS	🗆 jdoge	Josie	Doge
	🗆 jgarcia	Jerry	Garcia
NAME 📤	< 1 2 > PAGE 1 OF 2		ITEMS 1 - 5 OF 6
Database Servers			
DEMO EXAMPLE	2 Please assign roles to th	e selected users/teams	KEY
	Althea Bully USER	SELECT ROLES	×
Demo Inventory		Admin	
King PLC		Update	SAVE
		Ad Hoc	
		Use	
		Read	

The example above shows options associated with inventories. Different resources have different options available:

- Admin allows read, run, and edit privileges (applies to all resources)
- Use allows use of a resource in a job template (applies all resources except job templates)
- Update allows updating of project via the SCM Update (applies to projects and inventories)
- Ad Hoc allows use of Ad Hoc commands (applies to inventories)
- **Execute** allows launching of a job template (applies to job templates)
- **Read** allows view-only access (applies to all resources)

Tip: Use the Key button in the roles selection pane to display a description of each of the roles.

b. Select the role to apply to the selected user or team.

Note:

You can assign roles to multiple users and teams by navigating between the **Users** and **Teams** tabs without saving.

/ DEMO EXAMPLE / PE	RMISSIONS			
MPLE	DEMO EXAMPLE ADD USEI			•
PERMISSIONS	USERS TEAMS		Q	КЕҮ
	USERNAME [^]	FIRST NAME	LAST NAME 🗢	
	althea	Althea	Bully	
	austin78	Austin	Texas	
	gdoge	Gerry	Doge	
S HOSTS	🗹 jdoge	Josie	Doge	
	🗆 jgarcia	Jerry	Garcia	
JAME 🕈	< 1 2 > PAGE 1 OF 2			TEMS 1 - 5 OF 6
Database Servers	2 Please assign roles to the	selected users/teams		KEY
	Althea Bully USER	SELECT ROLES		×
Demo Inventory	Josie Doge USER	SELECT ROLES		×
	Production Operatio TEAM	SELECT ROLES		×
			CANCEL	SAVE

4. Review your role assignments for each user and team.

/ DEMO EXAMPLE / PI	ERMISSIONS				
	DEMO EXAMPLE ADD USE	RS / TEAMS		•	
MPLE	1 Please select Users / Tear	ns from the lists below.			
PERMISSIONS	USERS				
	SEARCH		Q	KEY	
	USERNAME [▲]	FIRST NAME	LAST NAME		
	🗹 althea	Althea	Bully		
	austin78	Austin	Texas	_	
	□ gdoge	Gerry	Doge		
ES HOSTS	🗹 jdoge	Josie	Doge		
	🗆 jgarcia	Jerry	Garcia		
NAME 🔺	< 1 2 > PAGE 1 OF 2		ITEMS	1 - 5 OF 6	
Database Servers	2 Please assign roles to the	selected users/teams		KEY	
DEMO EXAMPLE	Althea Bully USER	× Update		×	
Demo Inventory	-				
King PLC	Josie Doge USER	× Use		×	
	Production Operatio TEAM	× Admin		×	
			CANCEL	SAVE	

5. Click **Save** when done, and the Add Users/Teams window closes to display the updated roles assigned for each user and team.

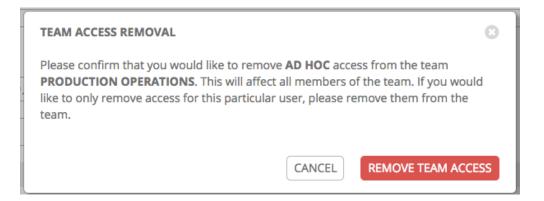
USER *	ROLE	TEAM ROLES
admin	SYSTEM ADMINISTRATOR	
althea	× AD HOC SYSTEM AUDITOR × USE	
jdoge	× UPDATE X USE	
mags3707	SYSTEM ADMINISTRATOR	× AD HOC 앞 × ADMIN 섬 × USE 삼
yser	SYSTEM AUDITOR	

To remove Permissions for a particular user, click the Disassociate (x) button next to its resource.

USER ^	ROLE	TEAM ROLES
		TDAM ROLES
admin	SYSTEM ADMINISTRATOR	
althea	* AD HOC SYSTEM AUDITOR X USE	
jdoge	× UPDATE × USE	
mags3707	SYSTEM ADMINISTRATOR	X AD HOC W X ADMIN W X USE W
yser	SYSTEM AUDITOR	
		ITEMS 1-5

This launches a confirmation dialog, asking you to confirm the disassociation.

TEMS 1-5



15.3 Work with Notifications

Clicking the **Notifications** tab allows you to review any notification integrations you have setup. If none are setup, the following screen displays with links to create one:

TEMPLATES / Example / NOTIFICATIONS	•
Example DETAILS PERMISSIONS NOTIFICATIONS COMPLETED JOBS SCHEDULES	0
	GO TO NOTIFICATIONS TO ADD A NEW TEMPLATE
THIS LIST IS POPULATED BY NOTIFICATION TEMPLATES ADDED FROM THE NOTIFICATIONS SECTION	

Follow the on-screen links to create a notification template. Refer to Notifications for more information.

DETAILS PERMISSIONS NOTIFICATION	VS COMPLETED JOBS SCHEDULES		
EARCH	Q KEY		GO TO NOTIFICATI ADD A NEW TEP
NAME 🗖	TYPE 🗢	SUCCESS	FAILURE
test-actions-notification test notification-template	Slack	OFF	OFF
e2e-ae53906d-notification-template	SMS	OFF	OFF
test-actions-notification-template	Slack	OFF	OFF
test-actions-notification-template@3:05:07 PM	EMAIL	OFF	OFF

15.4 View Completed Jobs

The **Completed Jobs** tab provides details of how this job template has been run. It provides you with the ID, Name, Job Type, when it completed, and allows you to relaunch or delete the job. You can filter the list of completed jobs using the job ID, Name, Type, or if the Job Failed.

ETAILS	ERMISSIONS NOTIFICATIONS COMPLETED JOBS SCHEDULES		
EARCH	Q		
17 - Demo	Iob Template Playbook Run		
STARTED 9/11/	201811:33:38 PM FINISHED 9/11/201811:33:43 PM		
LAUNCHED BY	admin		
OB TEMPLATE	Demo Job Template	al de	1
INVENTORY	Demo Inventory		
PROJECT	Demo Project		
CREDENTIALS	9 Demo Credential		
	2018 11:33:10 PM EINISHED 9/11/2018 11:33:16 PM		
AUNCHED BY OB TEMPLATE NVENTORY PROJECT	2018 11:33:10 PM FINISHED 9/11/2018 11:33:16 PM admin Demo Job Template Demo Inventory Demo Project <pre> Credential </pre>	A	
LAUNCHED BY OB TEMPLATE INVENTORY PROJECT CREDENTIALS	admin Demo Job Template Demo Inventory Demo Project	đ	
LAUNCHED BY JOB TEMPLATE INVENTORY PROJECT CREDENTIALS • 11 - Demo	admin Demo Job Template Demo Inventory Demo Project https://www.credential	37	
AUNCHED BY OB TEMPLATE INVENTORY PROJECT CREDENTIALS 11 - Demo STARTED 9/11/ LAUNCHED BY	admin Demo Job Template Demo Inventory Demo Project (*) Demo Credential Job Template Playbook Run 2018 11:31:51 PM FINISHED 9/11/2018 11:31:56 PM	3°	
AUNCHED BY OB TEMPLATE INVENTORY PROJECT CREDENTIALS 11 - Demo STARTED 9/11/ LAUNCHED BY OB TEMPLATE	admin Demo Job Template Demo Inventory Demo Credential Iob Template Playcook Run 2018 11:31:51 PM FINISHED 9/11/2018 11:31:56 PM admin	SI SI	
LAUNCHED BY IOB TEMPLATE INVENTORY PROJECT CREDENTIALS 11 - Demo STARTED 9/11/	admin Demo Job Template Demo Inventory Demo Project Demo Credential Job Template PlaySeook Run 2018 11:31:51 PM FINISHED 9/11/2018 11:31:56 PM admin Demo Job Template	S S	

Sliced jobs that display on this list are labeled accordingly, with the number of sliced jobs that have run:

126 - SJT Playbook Run	Slice Job 1/2		
STARTED 12/10/2018 4:44:	21 PM FINISHED 12/	/10/2018 4:44:38 PM	
WORKFLOW JOB SJT			
JOB TEMPLATE SJT			
INVENTORY INV			
PROJECT LOTR			

15.5 Scheduling

Access the schedules for a particular job template from the Schedules tab. Otherwise, you can launch the scheduled

iobs list via the

button. Scheduling from the job template page opens the **Schedules** page.

TE	EMPLATES / New Template with Dependencies / SCHEDULES					0		
	New Temp	ate with Dependencies						ø
	DETAILS	PERMISSIONS NOTIFICATIONS COMPLETED JOBS	SCHEDULES					
	SEARCH			Q	KEY			•
		NAME [•]	FIRST RUN 🗢		NEXT RUN 🗢	FINAL RUN 🗢		ACTIONS
	ON	Schedule 1	10/10/2018 11:00:00 PM		10/10/2018 11:00:00 PM	10/10/2018 11:00:00 PM	đ	۶ û
								ITEMS 1 - 1

This page displays a list of the schedules that are currently available for the selected **Job Template**. The schedule list may be sorted and searched by any of the following:

- Name: Clicking the schedule name opens the Edit Schedule dialog
- First Run: The first scheduled run of this task
- Next Run: The next scheduled run of this task
- Final Run: If the task has an end date, this is the last run of the task

Buttons located in the upper right corner of the Schedules screen provide the following actions:

- · View Activity Stream
- · Add a new schedule

15.5.1 Schedule a Job Template

To create a new schedule:

- 1. From the Schedules screen, click the button.
- 2. Enter the appropriate details into the following fields:
- Name
- Start Date
- Start Time
- Local Time Zone: the entered Start Time should be in this timezone
- Repeat Frequency: the appropriate options display as the update frequency is modified

Note: Jobs are scheduled in UTC. Repeating jobs that runs at a specific time of day may move relative to a local timezone when Daylight Saving Time shifts occur.

The Schedule Description below displays the specifics of the schedule and a list of the scheduled occurrences in the selected Local Time Zone.

ball *STATE TOTALE *STATE TOTALE *STATE TOTALE (PER/CAMMASS) 1:0.0.1. THE ZONE *BEPAIT TREQUENCY *BEPAIT TREQUENCY American New York *BEPAIT TREQUENCY *BEPAIT TREQUENCY *Reversame York *BEPAIT TREQUENCY *BEPAIT TREQUENCY *BEPAIT TREQUENCY *BEPAIT TREQUENCY *BEPAIT TREQUENCY	LATES / Demo Job Template / SCHEDULES / CREATE S							
Daily Scan 4/03/2017 4/2017 4/2017	ily Scan							
• LOCAL TIME ZONE • REPEAT FREQUENCY AmericanNew_York • Day • Repear To FREQUENCY • END • Repear To FREQUENCY • END • EVERY • END •	IAME		* START DATE		* START TIME (H	IH24:MM:SS)		
America/New_York Day Day Contention Contention Day Contention Con	Daily Scan		4/03/2017		22	्रेः 45	Ĵ: 00	0
REQUENCY DETAILS EVERY *END 2 O Date END TIME (H-024MM-SS) 23 O : 45 SCHEDULE DESCRIPTION every 2 days until July 17, 2017 OCCURRENCES (Limited to first 10) DATT 224-500 EDT 4/3/2017 224-500 EDT 4/3/2	OCAL TIME ZONE		* REPEAT FREQUENCY					
EVERY • END 2 O DAYS • END TIME (+H:24.MM.455) 23 : 45 : 0 : 0 : 0 : 0 : 0 : 0 : 0 : 0 : 0 :	merica/New_York	•	Day	Ŧ				
Image:	EQUENCY DETAILS							
END TIME (+H24:MM:SS) 23	EVERY		* END		* END DATE			
23 0: 45 0: 00 0 SCHEDULE DESCRIPTION every 2 days until July 17, 2017 OCCURRENCES (untited to first 10) DATE FORMAT @ LOCAL TIME O UTC 4/3/2017 22:45:00 EDT 4/9/2017 22:45:00 EDT 4/13/2017 22		DAYS	On Date	•	7/17/2	017		
23 23: 45 25: 00 20 SCHEDULE DESCRIPTION every 2 days until July 17, 2017 OCCURRENCES (Junited to first 10) DATE FORMAT (*) LOCAL TIME (*) UTC 4/3/2017 22:45:00 EDT 4/3/2017 22:45:00 EDT 4/3/2017 22:45:00 EDT 4/11/2017 22:45:00 EDT 4/13/2017 22:45:00 EDT								
SCHEDULE DESCRIPTION every 2 days until July 17, 2017 OCCURRENCES (Limited to first 10) DATE FORMAT (*) LOCAL TIME (*) UTC 4/3/2017 22:45:00 EDT 4/7/2017 22:45:00 EDT 4/17/2017 22:45:00 EDT		^						
every 2 days until july 17, 2017 OCCURRENCES (Limited to first 10 DATE FORMAT @ LOCAL TIME O UTC 4/3/2017 22:45:00 EDT 4/3/2017 22:45:00 EDT 4/13/2017 22:45:00 EDT		×						
	every 2 days until July 17, 2017 OCCURRENCES (Limited to first 10) DATE FORMAT	AL TIME 🔿 UTC						
1	every 2 days until July 17, 2017 OCCURRENCES (Limited to first 10) DATE FORMAT ® LOCA 4/3/2017 22:45:00 EDT 4/5/2017 22:45:00 EDT 4/7/2017 22:45:00 EDT 4/11/2017 22:45:00 EDT 4/13/2017 22:45:00 EDT 4/15/2017 22:45:00 EDT 4/15/2017 22:45:00 EDT 4/17/2017 22:45:00 EDT	al time –) utc						
	every 2 days until July 17, 2017 OCCURRENCES (Limited to first 10) DATE FORMAT (a) LOCA 4/3/2017 22:45:00 EDT 4/5/2017 22:45:00 EDT 4/17/2017 22:45:00 EDT 4/21/2017 22:45:00 EDT	al time 🔿 utc						
	every 2 days until July 17, 2017 OCCURRENCES (Limited to first 10) DATE FORMAT ® LOCA 4/3/2017 22:45:00 EDT 4/3/2017 22:45:00 EDT 4/17/2017 22:45:00 EDT 4/17/2017 22:45:00 EDT 4/17/2017 22:45:00 EDT 4/17/2017 22:45:00 EDT 4/17/2017 22:45:00 EDT 4/17/2017 22:45:00 EDT 4/12/2017 22:45:00 EDT 4/21/2017 22:45:00 EDT 4/21/2017 22:45:00 EDT 4/21/2017 22:45:00 EDT	al time 🔿 utc						
	every 2 days until July 17, 2017 OCCURRENCES (Limited to first 10) DATE FORMAT ® LOCA 4/3/2017 22:45:00 EDT 4/3/2017 22:45:00 EDT 4/17/2017 22:45:00 EDT 4/17/2017 22:45:00 EDT 4/17/2017 22:45:00 EDT 4/17/2017 22:45:00 EDT 4/17/2017 22:45:00 EDT 4/17/2017 22:45:00 EDT 4/12/2017 22:45:00 EDT 4/21/2017 22:45:00 EDT 4/21/2017 22:45:00 EDT 4/21/2017 22:45:00 EDT	ALTIME O UTC						
	every 2 days until July 17, 2017 OCCURRENCES (Limited to first 10) DATE FORMAT ® LOCA 4/3/2017 22:45:00 EDT 4/3/2017 22:45:00 EDT 4/17/2017 22:45:00 EDT 4/17/2017 22:45:00 EDT 4/17/2017 22:45:00 EDT 4/17/2017 22:45:00 EDT 4/17/2017 22:45:00 EDT 4/17/2017 22:45:00 EDT 4/12/2017 22:45:00 EDT 4/21/2017 22:45:00 EDT 4/21/2017 22:45:00 EDT 4/21/2017 22:45:00 EDT	altime () utc						
	every 2 days until July 17, 2017 OCCURRENCES (Limited to first 10) DATE FORMAT ® LOCA 4/3/2017 22:45:00 EDT 4/3/2017 22:45:00 EDT 4/17/2017 22:45:00 EDT 4/17/2017 22:45:00 EDT 4/17/2017 22:45:00 EDT 4/17/2017 22:45:00 EDT 4/17/2017 22:45:00 EDT 4/17/2017 22:45:00 EDT 4/12/2017 22:45:00 EDT 4/21/2017 22:45:00 EDT 4/21/2017 22:45:00 EDT 4/21/2017 22:45:00 EDT	altime () utc						

Note: If **Prompt on Launch** was selected for the **Credentials** field, and you create or edit scheduling information for your job template, a **Prompt** button displays at the bottom of the Schedules form. You will not be able to remove the default machine credential in the Prompt dialog without replacing it with another machine credential before you can save it. Below is an example of such a message:

	PROMPT O	
Schedule 1		(
* NAME	CREDENTIAL PREVIEW * START TIME (HH24:MM:SS)	
Schedule 1	SELECTED No credentials selected REVERT 23 0:0 0:0	Ŷ
* LOCAL TIME ZONE	▲ This job template has a default Machine credential which must be included or replaced before	
America/Denver	proceeding.	
FREQUENCY DETAILS	Credential Type: Machine 🔹	
* EVERY	SEARCH Q KEY * OCCURRENCES	
1	NAME *	~
SCHEDULE DESCRIPTION	O Demo Credential	
every day for 1 time	O Machine credential	
OCCURRENCES (Limited to first 10)	D	
10-10-2018 23:00:00		
	CANCEL	
	PROMPT	NCEL SAVE

Note: To able to set extra_vars on schedules, you must select **Prompt on Launch** for **EXTRA VARI-ABLES** on the job template, or a enable a survey on the job template, then those answered survey questions become

extra_vars.

3. When satisfied with the schedule specifics, click Save.

Once the schedule is saved, the list of schedules display for the associated job template.

Use the **ON/OFF** toggle button to quickly activate or deactivate this schedule.

Other actions for schedules are available under the Actions column:

- Edit Schedule
- Delete schedule

15.6 Surveys

Job types of Run or Check will provide a way to set up surveys in the Job Template creation or editing screens. Surveys set extra variables for the playbook similar to 'Prompt for Extra Variables' does, but in a user-friendly question and

answer way. Surveys also allow for validation of user input. Click the

ADD SURVEY

button to create a survey.

Use cases for surveys are numerous. An example might be if operations wanted to give developers a "push to stage" button they could run without advanced Ansible knowledge. When launched, this task could prompt for answers to questions such as, "What tag should we release?"

Many types of questions can be asked, including multiple-choice questions.

Note: Surveys are only available to those with Enterprise-level licenses.

15.6.1 Create a Survey

To create a survey:

1. Click on the

ADD SURVEY

button to bring up the **Add Survey** window.

Use the ON/OFF toggle button at the top of the screen to quickly activate or deactivate this survey prompt.

- 2. A survey can consist of any number of questions. For each question, enter the following information:
- Name: The question to ask the user
- Description: (optional) A description of what's being asked of the user.
- Answer Variable Name: The Ansible variable name to store the user's response in. This is the variable to be used by the playbook. Variable names cannot contain spaces.
- Answer Type: Choose from the following question types.
 - Text: A single line of text. You can set the minimum and maximum length (in characters) for this answer.
 - *Textarea*: A multi-line text field. You can set the minimum and maximum length (in characters) for this answer.
 - *Password*: Responses are treated as sensitive information, much like an actual password is treated. You can set the minimum and maximum length (in characters) for this answer.

	6	Э
ADD SURVEY PROMPT	PREVIEW	
*PROMPT	PLEASE ADD A SURVEY PROMPT ON THE LEFT.	
Which group(s) should include this user?		
DESCRIPTION		
Enter groups, one per line.		
*ANSWER VARIABLE NAME @		
group_name		
*ANSWER TYPE		
Text 💌		
MINIMUM LENGTH MAXIMUM LENGTH		
0		
DEFAULT ANSWER		
REQUIRED		
CANCEL	CANCEL SAVE	

- *Multiple Choice (single select)*: A list of options, of which only one can be selected at a time. Enter the options, one per line, in the **Multiple Choice Options** box.
- *Multiple Choice (multiple select)*: A list of options, any number of which can be selected at a time. Enter the options, one per line, in the **Multiple Choice Options** box.
- Integer: An integer number. You can set the minimum and maximum length (in characters) for this answer.
- Float: A decimal number. You can set the minimum and maximum length (in characters) for this answer.
- **Default Answer**: The default answer to the question. This value is pre-filled in the interface and is used if the answer is not provided by the user.
- **Required**: Whether or not an answer to this question is required from the user.
- 3. Once you have entered the question information, click the

A stylized version of the survey is presented in the Preview pane. For any question, you can click on the **Edit** button to edit the question, the **Delete** button to delete the question, and click and drag on the grid icon to rearrange the order of the questions.

- 4. Return to the left pane to add additional questions.
- 5. When done, click **Save** to save the survey.



button to add the question.

NEW JOB TEMPLATE SURVEY ON			Θ
ADD SURVEY PROMPT	PREVIEW		
*PROMPT	*WHICH GROUP(S) SHOULD INCLUDE THIS USER? Enter groups, one per line.	Salt.	圓
DESCRIPTION	Click and hold down to drag	5	
*ANSWER VARIABLE NAME @	the question to reorder it.		
*ANSWER TYPE			
Choose an answer type *			
REQUIRED			
CANCEL			
	CANCEL	S	AVE

15.6.2 Optional Survey Questions

The **Required** setting on a survey question determines whether the answer is optional or not for the user interacting with it.

Behind the scenes, optional survey variables can be passed to the playbook in extra_vars, even when they aren't filled in.

- If a non-text variable (input type) is marked as optional, and is not filled in, no survey extra_var is passed to the playbook.
- If a text input or text area input is marked as optional, is not filled in, and has a minimum length > 0, no survey extra_var is passed to the playbook.
- If a text input or text area input is marked as optional, is not filled in, and has a minimum length === 0, that survey extra_var is passed to the playbook, with the value set to an empty string ("").

15.7 Launch a Job Template

A major benefit of Ansible Tower is the push-button deployment of Ansible playbooks. You can easily configure a template within Tower to store all parameters you would normally pass to the ansible-playbook on the command line–not just the playbooks, but the inventory, credentials, extra variables, and all options and settings you can specify on the command line.

Easier deployments drive consistency, by running your playbooks the same way each time, and allow you to delegate responsibilities–even users who aren't Ansible experts can run Tower playbooks written by others.

To launch a job template:

1. Access the job template from the **Templates** navigational link or while in the Job Template Details view, scroll to the bottom to access it from a list of templates.

ARCH	Q, KEY		
Demo Job Tem	plate Job Template		
ACTIVITY			
NVENTORY	Demo Inventory		
PROJECT	Demo Project	A C	2 1
REDENTIALS	Remo Credential		
AST MODIFIED	9/11/2018 11:33:43 PM by admin		
AST RAN	9/11/2018 11:33:43 PM		
xample Job Te	mplate Job Template		
VVENTORY	Network Inventory Small	The second s	h ti
ROJECT	Demo Project	01 U	
AST MODIFIED	9/12/2018 5:09:48 AM by admin		

2. Click the button.

A job may require additional information to run. The following data may be requested at launch:

- · Credentials that were setup
- Passwords or passphrases that have been set to Ask
- A survey, if one has been configured for the job templates
- Extra variables, if requested by the job template

Note: If a job has user-provided values, then those are respected upon relaunch. If the user did not specify a value, then the job uses the default value from the job template. Jobs are not relaunched as-is. They are relaunched with the user prompts re-applied to the job template.

Below is an example job launch that prompts for Job Tags, and runs the example survey created in Surveys.

LAUNCH JOB HELI	O WORLD		8
OTHER PROMPTS	SURVEY		
JOB TAGS			
INVENTORY Demo Inventory	CREDENTIAL Demo Credential	CANCEL	NEXT

LAUNCH JOB HE	LLO WORLD		8
OTHER PROMPTS	SURVEY		
*WHICH GROUP(S) S Enter groups, one p	HOULD INCLUDE THIS USER? Der line.		
INVENTORY Demo Inventory	CREDENTIAL Demo Credential	CANCEL	LAUNCH

Along with any extra variables set in the job template and survey, Tower automatically adds the following variables to the job environment:

- tower_job_id: The Job ID for this job run
- tower_job_launch_type: The description to indicate how the job was started:
 - manual: Job was started manually by a user.
 - relaunch: Job was started via relaunch.
 - callback: Job was started via host callback.
 - scheduled: Job was started from a schedule.
 - dependency: Job was started as a dependency of another job.
 - workflow: Job was started from a workflow job.
 - sync: Job was started from a project sync.
 - scm: Job was created as an Inventory SCM sync.
- tower_job_template_id: The Job Template ID that this job run uses
- tower_job_template_name: The Job Template name that this job uses
- tower_project_revision: The revision identifier for the source tree that this particular job uses (it is also the same as the job's field scm_revision)
- tower_user_email: The user email of the Tower user that started this job. This is not available for callback or scheduled jobs.
- tower_user_first_name: The user's first name of the Tower user that started this job. This is not available for callback or scheduled jobs.
- tower_user_id: The user ID of the Tower user that started this job. This is not available for callback or scheduled jobs.
- tower_user_last_name: The user's last name of the Tower user that started this job. This is not available for callback or scheduled jobs.

- tower_user_name: The user name of the Tower user that started this job. This is not available for callback or scheduled jobs.
- tower_schedule_id: If applicable, the ID of the schedule that launched this job
- tower_schedule_name: If applicable, the name of the schedule that launched this job
- tower_workflow_job_id: If applicable, the ID of the workflow job that launched this job
- tower_workflow_job_name: If applicable, the name of the workflow job that launched this job. Note this is also the same as the workflow job template.

All variables are also given an "awx" prefix, for example, awx_job_id.

Upon launch, Tower automatically redirects the web browser to the Job Status page for this job under the Jobs tab.

Note: You can re-launch the most recent job from the list view to re-run on all hosts or just failed hosts in the specified inventory. Refer to *Jobs* in the *Ansible Tower User Guide* for more detail.

When slice jobs are running, job lists display the workflow and job slices, as well as a link to view their details individually.

• 126 - SJT	Playbook Run	Slice Job 1/2		
STARTED 12/	10/2018 4:44:2 ⁻	1 PM FIN	NISHED	12/10/2018 4:44:38 PM
WORKFLOW JO	b SJT			
JOB TEMPLATE	SJT			
INVENTORY	INV			
PROJECT	LOTR			

15.8 Copy a Job Template

Ansible Tower 3.0 introduced the ability to copy a Job Template. If you choose to copy Job Template, it **does not** copy any associated schedule, notifications, or permissions. Schedules and notifications must be recreated by the user or admin creating the copy of the Job Template. The user copying the Job Template will be granted the admin permission, but no permissions are assigned (copied) to the Job Template.

1. Access the job template that you want to copy from the **Templates** navigational link or while in the Job Template Details view, scroll to the bottom to access it from a list of templates.

TEMPLATES 2					
SEARCH	Q KEY			+	
Demo Job Tem	Job Template				
ACTIVITY					
INVENTORY	Demo Inventory				
PROJECT	Demo Project	3°	2	Û	
CREDENTIALS	4. Demo Credential				
LAST MODIFIED	9/11/2018 11:33:43 PM by admin				
LAST RAN	9/11/2018 11:33:43 PM				
Example Job Te	mplate Job Template				
INVENTORY	Network Inventory Small	T.	671	ŵ	
PROJECT	Demo Project	24	40	L.	
LAST MODIFIED	9/12/2018 5:09:48 AM by admin				
				ITEMS 1-2	

2. Click the button.

A new template opens with the name of the template from which you copied and a timestamp.

- 3. Replace the contents of the **Name** field with a new name, and provide or modify the entries in the other fields to complete this page.
- 4. Click Save when done.

15.9 Scan Job Templates

Scan jobs are no longer supported starting with Ansible Tower 3.2. This system tracking feature was used as a way to capture and store facts as historical data. Facts are now stored in Tower via fact caching. For more information, see *Fact Caching*.

If you have Job Template Scan Jobs in your system prior to Ansible Tower 3.2, they have been converted to type run (like normal job templates) and retained their associated resources (i.e. inventory, credential). Job Template Scan Jobs that do not have a related project are assigned a special playbook by default, or you can specify a project with your own scan playbook. A project was created for each organization that points to https://github.com/ansible/tower-fact-modules and the Job Template was set to the playbook, https://github.com/ansible/tower-fact-modules/blob/master/scan_facts.yml.

15.9.1 Fact Scan Playbooks

The scan job playbook, scan_facts.yml, contains invocations of three fact scan modules - packages, services, and files, along with Ansible's standard fact gathering. The scan_facts.yml playbook file looks like the following:

```
- hosts: all
vars:
    scan_use_checksum: false
    scan_use_recursive: false
tasks:
    - scan_packages:
    - scan_services:
    - scan_files:
        paths: '{{ scan_file_paths }}'
        get_checksum: '{{ scan_use_checksum }}'
```

(continues on next page)

(continued from previous page)

```
recursive: '{{ scan_use_recursive }}'
when: scan_file_paths is defined
```

The scan_files fact module is the only module that accepts parameters, passed via extra_vars on the scan job template.

```
scan_file_paths: '/tmp/'
scan_use_checksum: true
scan_use_recursive: true
```

- The scan_file_paths parameter may have multiple settings (such as /tmp/ or /var/log).
- The scan_use_checksum and scan_use_recursive parameters may also be set to false or omitted. An omission is the same as a false setting.

Scan job templates should enable become and use credentials for which become is a possibility. You can enable become by checking the **Enable Privilege Escalation** from the Options menu:

OPTIONS
 Enable Privilege Escalation @ Allow Provisioning Callbacks @
 Enable Concurrent Jobs 😨
Use Fact Cache Ø

Note: If you maintained scan job templates in Ansible Tower 3.1.x and then upgrade to Ansible Tower 3.2, a new "Tower Fact Scan - Default" project is automatically created for you. This project contains the old scan playbook previously used in earlier versions of Ansible Tower.

15.9.2 Supported OSes for scan_facts.yml

If you use the scan_facts.yml playbook with use fact cache, ensure that your OS is supported:

- Red Hat Enterprise Linux 5, 6, & 7
- CentOS 5, 6, & 7
- Ubuntu 12.04, 14.04, 16.04
- OEL 6 & 7
- SLES 11 & 12
- Debian 6, 7, 8
- Fedora 22, 23, 24
- Amazon Linux 2016.03
- · Windows Server 2008 and later

Note that some of these operating systems may require initial configuration in order to be able to run python and/or have access to the python packages (such as python-apt) that the scan modules depend on.

15.9.3 Pre-scan Setup

The following are examples of playbooks that configure certain distributions so that scan jobs can be run against them.

Bootstrap Ubuntu (16.04)

```
name: Get Ubuntu 15, 16, and on ready
hosts: all
sudo: yes
gather_facts: no
tasks:
name: install python-simplejson
raw: sudo apt-get -y update
raw: sudo apt-get -y install python-simplejson
raw: sudo apt-get install python-apt
```

Bootstrap Fedora (23, 24)

```
---
- name: Get Fedora ready
hosts: all
sudo: yes
gather_facts: no
tasks:
- name: install python-simplejson
raw: sudo dnf -y update
raw: sudo dnf -y install python-simplejson
raw: sudo dnf -y install rpm-python
```

CentOS 5 or Red Hat Enterprise Linux 5 may also need the simplejson package installed.

15.9.4 Custom Fact Scans

A playbook for a custom fact scan is similar to the example of the Fact Scan Playbook above. As an example, a playbook that only uses a custom scan_foo Ansible fact module would look like this:

scan_custom.yml:

```
- hosts: all
gather_facts: false
tasks:
    - scan_foo:
```

scan_foo.py:

```
def main():
    module = AnsibleModule(
        argument_spec = dict())

    foo = [
        {
            "hello": "world"
        },
        {
            "foo": "bar"
        }
        ]
        results = dict(ansible_facts=dict(foo=foo))
        module.exit_json(**results)

main()
```

To use a custom fact module, ensure that it lives in the /library/ subdirectory of the Ansible project used in the scan job template. This fact scan module is very simple, returning a hard-coded set of facts:

```
{
    "hello": "world"
    },
    {
        "foo": "bar"
    }
]
```

[

Refer to the Module Provided 'Facts' section of the Ansible documentation for more information.

15.10 Fact Caching

Tower can store and retrieve facts on a per-host basis through an Ansible Fact Cache plugin. This behavior is configurable on a per-job template basis. Fact caching is turned off by default but can be enabled to serve fact requests for all hosts in an inventory related to the job running. This allows you to use job templates with --limit while still having access to the entire inventory of host facts. A global timeout setting that the plugin enforces per-host, can be specified (in seconds) through the Configure Tower interface under the Jobs tab:

ETTINGS / EDIT CONFIGURATION					
CONFIGURE Tower					
AUTHENTICATION JOBS SYSTEM USER INTERFACE					
ANSIBLE MODULES ALLOWED FOR AD HOC JOBS	REVERT	*JOB EXECUTION PATH	REVERT	* MAXIMUM SCHEDULED JOBS	REVERT
<pre>x command * shell * yum * apt * apt.key x apt.repository * apt.prm * service * group * user mount * ping * selinux * setup * win_ping * win_service * win_updates * win_group * win_user</pre>		/tmp		10	
PATHS TO EXPOSE TO ISOLATED JOBS O	REVERT	ANSIBLE CALLBACK PLUGINS	REVERT	PATHS TO HIDE FROM ISOLATED JOBS D	REVERT
*ENABLE JOB ISOLATION @		DEFAULT JOB TIMEOUT @	REVERT	DEFAULT INVENTORY UPDATE TIMEOUT	REVERT
DEFAULT PROJECT UPDATE TIMEOUT 🔞	REVERT	PER-HOST ANSIBLE FACT CACHE TIMEOUT	REVERT		
0		0			
EXTRA ENVIRONMENT VARIABLES					
1 { 2 "HOME": "/var/lib/awx", 3 "USER": "awx" 4 }					
REVERT ALL TO DEFAULT					CANCEL
					Copyright © 2017 Red F

Upon launching a job that uses fact cache (use_fact_cache=True), Tower will store all ansible_facts associated with each host in the inventory associated with the job. The Ansible Fact Cache plugin that ships with Ansible Tower will only be enabled on jobs with fact cache enabled (use_fact_cache=True).

When a job that has fact cache enabled (use_fact_cache=True) finishes running, Tower will restore all records for the hosts in the inventory. Any records with update times *newer* than the currently stored facts per-host will be updated in the database.

New and changed facts will be logged via Tower's logging facility. Specifically, to the system_tracking namespace or logger. The logging payload will include the fields:

- host_name
- inventory_id
- ansible_facts

where ansible_facts is a dictionary of all Ansible facts for host_name in Tower inventory, inventory_id.

Note: If a hostname includes a forward slash (/), fact cache will not work for that host. If you have an inventory with 100 hosts and one host has a / in the name, 99 of those hosts will still collect facts.

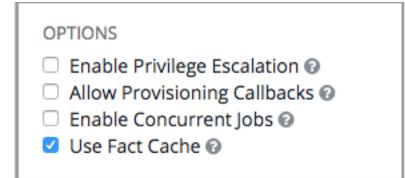
15.10.1 Benefits of Fact Caching

Fact caching saves a significant amount of time over running fact gathering. If you have a playbook in a job that runs against a thousand hosts and forks, you could easily spend 10 minutes gathering facts across all of those hosts. But if you run a job on a regular basis, the first run of it caches these facts and the next run will just pull them from the database. This cuts the runtime of jobs against large inventories, including Smart Inventories, by an enormous magnitude.

Note: Do not modify the tower.cfg file to apply fact caching. Custom fact caching could conflict with Tower's fact caching feature. It is recommended to use the fact caching module that comes with Ansible Tower. Fact caching

is not supported for isolated nodes.

You can choose to use cached facts in your job by enabling it in the **Options** field of the Job Templates window.



To clear facts, you need to run the Ansible clear_facts meta task. Below is an example playbook that uses the Ansible clear_facts meta task.

```
hosts: all
gather_facts: false
tasks:
name: Clear gathered facts from all currently targeted hosts
meta: clear_facts
```

The API endpoint for fact caching can be found at: http://<Tower server name>/api/v2/hosts/x/ ansible_facts.

15.11 Utilizing Cloud Credentials

Cloud Credentials can be used when syncing a respective cloud inventory. Cloud Credentials may also be associated with a Job Template and included in the runtime environment for use by a playbook. Cloud Credentials were introduced in Ansible Tower version 2.4.0 and these are currently supported:

- OpenStack
- Amazon Web Services
- Rackspace
- Google
- Azure
- VMware

15.11.1 OpenStack

The sample playbook below invokes the nova_compute Ansible OpenStack cloud module and requires credentials to do anything meaningful, and specifically requires the following information: auth_url, username, password, and project_name. These fields are made available to the playbook via the environmental variable OS_CLIENT_CONFIG_FILE, which points to a YAML file written by Tower based on the contents of the cloud credential. This sample playbook loads the YAML file into the Ansible variable space.

OS_CLIENT_CONFIG_FILE example:

```
clouds:
    devstack:
    auth:
        auth_url: http://devstack.yoursite.com:5000/v2.0/
        username: admin
        password: your_password_here
        project_name: demo
```

Playbook example:

```
- hosts: all
 gather_facts: false
 vars:
   config_file: "{{ lookup('env', 'OS_CLIENT_CONFIG_FILE') }}"
   nova_tenant_name: demo
   nova_image_name: "cirros-0.3.2-x86_64-uec"
   nova_instance_name: autobot
   nova_instance_state: 'present'
   nova_flavor_name: m1.nano
   nova_group:
     group_name: antarctica
     instance_name: deceptacon
     instance_count: 3
 tasks:
   - debug: msg="{{ config_file }}"
   - stat: path="{{ config_file }}"
     register: st
   - include_vars: "{{ config_file }}"
     when: st.stat.exists and st.stat.isreg
   - name: "Print out clouds variable"
     debug: msg="{{ clouds|default('No clouds found') }}"
   - name: "Setting nova instance state to: {{ nova_instance_state }}"
     local_action:
       module: nova_compute
       login_username: "{{ clouds.devstack.auth.username }}"
       login_password: "{{ clouds.devstack.auth.password }}"
```

15.11.2 Amazon Web Services

Amazon Web Services cloud credentials are exposed as the following environment variables during playbook execution (in the job template, choose the cloud credential needed for your setup):

- AWS_ACCESS_KEY_ID
- AWS_SECRET_ACCESS_KEY

All of the AWS modules will implicitly use these credentials when run via Tower without having to set the aws_access_key_id or aws_secret_access_key module options.

15.11.3 Rackspace

Rackspace cloud credentials are exposed as the following environment variables during playbook execution (in the job template, choose the cloud credential needed for your setup):

- RAX_USERNAME
- RAX_API_KEY

All of the Rackspace modules will implicitly use these credentials when run via Tower without having to set the username or api_key module options.

15.11.4 Google

Google cloud credentials are exposed as the following environment variables during playbook execution (in the job template, choose the cloud credential needed for your setup):

- GCE_EMAIL
- GCE_PROJECT
- GCE_CREDENTIALS_FILE_PATH

All of the Google modules will implicitly use these credentials when run via Tower without having to set the service_account_email, project_id, or pem_file module options.

15.11.5 Azure

Azure cloud credentials are exposed as the following environment variables during playbook execution (in the job template, choose the cloud credential needed for your setup):

- AZURE_SUBSCRIPTION_ID
- AZURE_CERT_PATH

All of the Azure modules implicitly use these credentials when run via Tower without having to set the subscription_id or management_cert_path module options.

15.11.6 VMware

VMware cloud credentials are exposed as the following environment variables during playbook execution (in the job template, choose the cloud credential needed for your setup):

- VMWARE_USER
- VMWARE_PASSWORD
- VMWARE_HOST

The sample playbook below demonstrates usage of these credentials:

```
- vsphere_guest:
    vcenter_hostname: "{{ lookup('env', 'VMWARE_HOST') }}"
    username: "{{ lookup('env', 'VMWARE_USER') }}"
    password: "{{ lookup('env', 'VMWARE_PASSWORD') }}"
    guest: newvm001
    from_template: yes
    template_src: centosTemplate
    cluster: MainCluster
    resource_pool: "/Resources"
    vm_extra_config:
        folder: MyFolder
```

15.12 Provisioning Callbacks

Provisioning callbacks are a feature of Tower that allow a host to initiate a playbook run against itself, rather than waiting for a user to launch a job to manage the host from the tower console. Please note that provisioning callbacks are *only* used to run playbooks on the calling host. Provisioning callbacks are meant for cloud bursting, ie: new instances with a need for client to server communication for configuration (such as transmitting an authorization key), not to run a job against another host. This provides for automatically configuring a system after it has been provisioned by another system (such as AWS auto-scaling, or a OS provisioning system like kickstart or preseed) or for launching a job programmatically without invoking the Tower API directly. The Job Template launched only runs against the host requesting the provisioning.

Frequently this would be accessed via a firstboot type script, or from cron.

To enable callbacks, check the *Allow Provisioning Callbacks* checkbox in the Job Template. This displays the **Provisioning Callback URL** for this job template.

Note: If you intend to use Tower's provisioning callback feature with a dynamic inventory, Update on Launch should be set for the inventory group used in the Job Template.

```
    OPTIONS
    PROVISIONING CALLBACK URL @
    HOST CONFIG KEY @

    Enable Privilege Escalation @
    https://10.42.0.42:443/api/v1/job_templates/5/callb.
    #

    Image: Allow Provisioning Callbacks @
    https://10.42.0.42:443/api/v1/job_templates/5/callb.
    #
```

Callbacks also require a Host Config Key, to ensure that foreign hosts with the URL cannot request configuration.

Click the button to create a unique host key for this callback, or enter your own key. The host key may be reused across multiple hosts to apply this job template against multiple hosts. Should you wish to control what hosts are able to request configuration, the key may be changed at any time.

To callback manually via REST, look at the callback URL in the UI, which is of the form:

http://<TOWER_SERVER_NAME>/api/v2/job_templates/1/callback/

The '1' in this sample URL is the job template ID in Tower.

The request from the host must be a POST. Here is an example using curl (all on a single line):

```
root@localhost:~$ curl -k -f -i -H 'Content-Type:application/json' -XPOST -d '{"host_

→config_key": "cfbaae23-81c0-47f8-9a40-44493b82f06a"}'

https://<TOWER_SERVER_NAME>/api/v2/job_templates/1/callback/
```

The requesting host must be defined in your inventory for the callback to succeed. If Tower fails to locate the host either by name or IP address in one of your defined inventories, the request is denied. When running a Job Template in this way, the host initiating the playbook run against itself must be in the inventory. If the host is missing from the inventory, the Job Template will fail with a "No Hosts Matched" type error message.

Note: If your host is not in inventory and Update on Launch is set for the inventory group, Tower attempts to update cloud based inventory source before running the callback.

Successful requests result in an entry on the Jobs tab, where the results and history can be viewed.

While the callback can be accessed via REST, the suggested method of using the callback is to use one of the example scripts that ships with Tower - /usr/share/awx/request_tower_configuration.sh (Linux/UNIX) or / usr/share/awx/request_tower_configuration.ps1 (Windows). Usage is described in the source code of the file by passing the -h flag, as shown below:

```
./request_tower_configuration.sh -h
Usage: ./request_tower_configuration.sh <options>
Request server configuration from Ansible Tower.
OPTIONS:
   -h
           Show this message
          Tower server (e.g. https://tower.example.com) (required)
   -s
   -k
          Allow insecure SSL connections and transfers
          Host config key (required)
   -c
           Job template ID (required)
   -t
   -e
           Extra variables
           Number of seconds between retries (default: 60)
   -s
```

This script is intelligent in that it knows how to retry commands and is therefore a more robust way to use callbacks than a simple curl request. As written, the script retries once per minute for up to ten minutes.

Note: Please note that this is an example script. You should edit this script if you need more dynamic behavior when detecting failure scenarios, as any non-200 error code may not be a transient error requiring retry.

Most likely you will use callbacks with dynamic inventory in Tower, such as pulling cloud inventory from one of the supported cloud providers. In these cases, along with setting *Update On Launch*, be sure to configure an inventory cache timeout for the inventory source, to avoid hammering of your Cloud's API endpoints. Since the request_tower_configuration.sh script polls once per minute for up to ten minutes, a suggested cache invalidation time for inventory (configured on the inventory source itself) would be one or two minutes.

While we recommend against running the request_tower_configuration.sh script from a cron job, a suggested cron interval would be perhaps every 30 minutes. Repeated configuration can be easily handled by scheduling in Tower, so the primary use of callbacks by most users is to enable a base image that is bootstrapped into the latest configuration upon coming online. To do so, running at first boot is a better practice. First boot scripts are just simple init scripts that typically self-delete, so you would set up an init script that called a copy of the request_tower_configuration.sh script and make that into an autoscaling image.

15.12.1 Passing Extra Variables to Provisioning Callbacks

Just as you can pass extra_vars in a regular Job Template, you can also pass them to provisioning callbacks. To pass extra_vars, the data sent must be part of the body of the POST request as application/json (as the content type). Use the following JSON format as an example when adding your own extra_vars to be passed:

'{"extra_vars": {"variable1":"value1","variable2":"value2",...}}'

(Added in Ansible Tower version 2.2.0.)

You can also pass extra variables to the Job Template call using curl, such as is shown in the following example:

For more information, refer to Launching Jobs with Curl.

15.12.2 Provisioning Callback through tower-cli

As an alternative to running the request_tower_configuration.sh script or a custom script, you can use tower-cli to make a provisioning callback, as in the following example:

15.13 Extra Variables

Note: Additional strict extra_vars validation was added in Ansible Tower 3.0.0. extra_vars passed to the job launch API are only honored if one of the following is true:

- They correspond to variables in an enabled survey
- ask_variables_on_launch is set to True

When you pass survey variables, they are passed as extra variables (extra_vars) within Tower. This can be tricky, as passing extra variables to a job template (as you would do with a survey) can override other variables being passed from the inventory and project.

For example, say that you have a defined variable for an inventory for debug = true. It is entirely possible that this variable, debug = true, can be overridden in a job template survey.

To ensure that the variables you need to pass are not overridden, ensure they are included by redefining them in the survey. Keep in mind that extra variables can be defined at the inventory, group, and host levels.

Note: Beginning with Ansible Tower version 2.4, the behavior for Job Template extra variables and Survey variables has changed. Previously, variables set using a Survey overrode any extra variables specified in the Job Template. In

2.4 and later, the Job Template extra variables dictionary is merged with the Survey variables. This may result in a change of behavior upon upgrading to 2.4.

Here are some simplified examples of extra_vars in YAML and JSON formats:

The configuration in YAML format:

```
launch_to_orbit: true
satellites:
    - sputnik
    - explorer
    - satcom
```

The configuration in JSON format:

```
{
    "launch_to_orbit": true,
    "satellites": ["sputnik", "explorer", "satcom"]
}
```

The following table notes the behavior (hierarchy) of variable precedence in Ansible Tower as it compares to variable precedence in Ansible.

Ansible Tower Variable Precedence Hierarchy (last listed wins)

Ansible	Tower
role de	efaults
dynamic inven	tory variables
inventory variables	Tower inventory variables
inventory group_vars	Tower group variables
inventory host_vars	Tower host variables
playbook g	roup_vars
playbook l	host_vars
host	facts
registered	variables
set_f	acts
play va	riables
play vars_prompt	(not supported in Tower)
play va	rs_files
role and inclu	ide variables
block va	ariables
task va	riables
extra variables	Job Template extra variables Job Template Survey (defaults) Job Launch extra variables

15.13.1 Relaunching Job Templates

Another change for Ansible Tower version 2.4 introduced a launch_type setting for your jobs. Instead of manually relaunching a job, a relaunch is denoted by setting launch_type to relaunch. The relaunch behavior deviates from the launch behavior in that it **does not** inherit extra_vars.

Job relaunching does not go through the inherit logic. It uses the same extra_vars that were calculated for the job being relaunched.

For example, say that you launch a Job Template with no extra_vars which results in the creation of a Job called **j1**. Next, say that you edit the Job Template and add in some extra_vars (such as adding "{ "hello": "world" }").

Relaunching j1 results in the creation of j2, but because there is no inherit logic and j1 had no $extra_vars$, j2 will not have any $extra_vars$.

To continue upon this example, if you launched the Job Template with the $extra_vars$ you added after the creation of **j1**, the relaunch job created (**j3**) will include the $extra_vars$. And relaunching **j3** results in the creation of **j4**, which would also include $extra_vars$.

CHAPTER

SIXTEEN

JOB SLICING

A sliced job refers to the concept of a distributed job. Distributed jobs are used for running a job across a very large number of hosts, allowing you to run multiple ansible-playbooks, each on a subset of an inventory, that can be scheduled in parallel across a cluster.

By default, Ansible runs jobs from a single control instance. Prior to Ansible Tower 3.4, a single Tower job would only be run as a single ansible-playbook run, which would not fully take advantage of Tower's ability to distribute work to multiple nodes in a cluster.

For jobs that do not require cross-host orchestration, job slicing solves this. Job slicing works by adding a Job Template field job_slice_count, which specifies the number of jobs into which to slice the Ansible run. When this number is greater than 1, Tower will generate a workflow from a job template instead of a job. The inventory will be distributed evenly amongst the slice jobs. The workflow job is then started, and proceeds as though it were a normal workflow. When launching a job, the API will return either a job resource (if job_slice_count = 1) or a workflow job resource. The corresponding Tower User Interface will redirect to the appropriate screen to display the status of the run.

16.1 Job slice considerations

Consider the following when setting up job slices:

- A sliced job creates a workflow job, and then that creates jobs.
- A job slice consists of a job template, an inventory, and a slice count.
- When executed, a sliced job splits each inventory into a number of "slice size" chunks. It then queues jobs of ansible-playbook runs on each chunk of the appropriate inventory. The inventory fed into ansible-playbook is a pared-down version of the original inventory that only contains the hosts in that particular slice. The completed sliced job that displays on the Jobs list are labeled accordingly, with the number of sliced jobs that have run:

• 126 - SJT	Playbook Run	Slice Job 1/2		
STARTED 12/	10/2018 4:44:2	1 PM FI	NISHED	12/10/2018 4:44:38 PM
WORKFLOW JO	B SJT			
JOB TEMPLATE	SJT			
INVENTORY	INV			
PROJECT	LOTR			

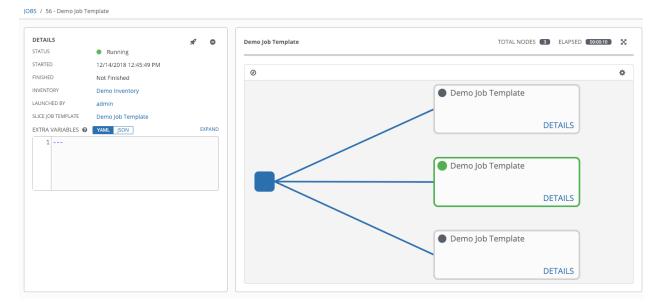
• These sliced jobs follow normal scheduling behavior (number of forks, queuing due to capacity, assignation to instance groups based on inventory mapping).

- Sliced job templates with prompts and/or extra variables behave the same as standard job templates, applying all variables and limits to the entire set of slice jobs in the resulting workflow job. However, when passing a limit to a Sliced Job, if the limit causes slices to have no hosts assigned, those slices will fail, causing the overall job to fail.
- A job slice job status of a distributed job is calculated in the same manner as workflow jobs; failure if there are any unhandled failures in its sub-jobs.

Warning: Any job that intends to orchestrate across hosts (rather than just applying changes to individual hosts) should not be configured as a slice job. Any job that does, may fail, and Tower will not attempt to discover or account for playbooks that fail when run as slice jobs.

16.2 Job slice execution behavior

When jobs are sliced, they can run on any Tower node and some may not run at the same time (insufficient capacity in the system, for example). When slice jobs are running, job details display the workflow and job slice(s) currently running, as well as a link to view their details individually.



By default, job templates are not normally configured to execute simultaneously (allow_simultaneous must be checked in the API or **Enable Concurrent Jobs** in the UI). Slicing overrides this behavior and implies allow_simultaneous even if that setting is unchecked. See *Job Templates* for information on how to specify this, as well as the number of job slices on your job template configuration.

The *Job Templates* section provides additional detail on performing the following operations in the Tower User Interface:

- Launch workflow jobs with a job template that has a slice number greater than one
- Cancel the whole workflow or individual jobs after launching a slice job template
- Relaunch the whole workflow or individual jobs after slice jobs finish running
- View the details about the workflow and slice jobs after a launching a job template
- Search slice jobs specifically after you create them (see subsequent section, Search job slices)

16.3 Search job slices

To make it easier to find slice jobs, use the Search functionality to apply a search filter to:

- job lists to show only slice jobs
- job lists to show only parent workflow jobs of job slices
- job templates lists to only show job templates that produce slice jobs

To show only slice jobs in job lists, as with most cases, you can filter either on the type (jobs here) or unified_jobs:

/api/v2/jobs/?job_slice_count__gt=1

To show only parent workflow jobs of job slices:

/api/v2/workflow_jobs/?job_template__isnull=false

To show only job templates that produce slice jobs:

/api/v2/job_templates/?job_slice_count__gt=1

CHAPTER

SEVENTEEN

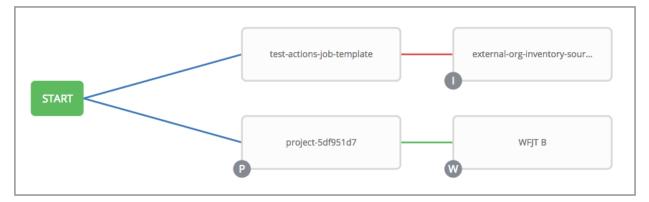
WORKFLOWS

Workflows allow you to configure a sequence of disparate job templates (or workflow templates) that may or may not share inventory, playbooks, or permissions. However, workflows have 'admin' and 'execute' permissions, similar to job templates. A workflow accomplishes the task of tracking the full set of jobs that were part of the release process as a single unit.

Note: Workflows are only available to those with Enterprise-level licenses.

Job or workflow templates are linked together using a graph-like structure called nodes. These nodes can be jobs, project syncs, or inventory syncs. A template can be part of different workflows or used multiple times in the same workflow. A copy of the graph structure is saved to a workflow job when you launch the workflow.

The example below shows a workflow that contains all three, as well as a workflow job template:



As the workflow runs, jobs are spawned from the node's linked template. Nodes linking to a job template which has prompt-driven fields (job_type, job_tags, skip_tags, limit) can contain those fields, and will not be prompted on launch. Job templates with promptable credential and/or inventory, WITHOUT defaults, will not be available for inclusion in a workflow.

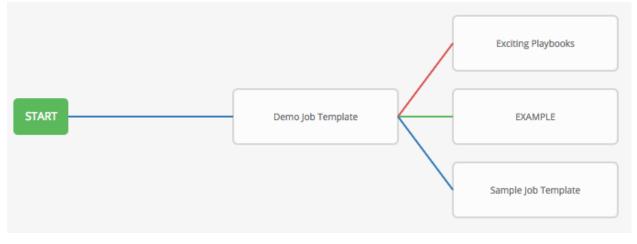
17.1 Workflow scenarios and considerations

Consider the following scenarios for building workflows:

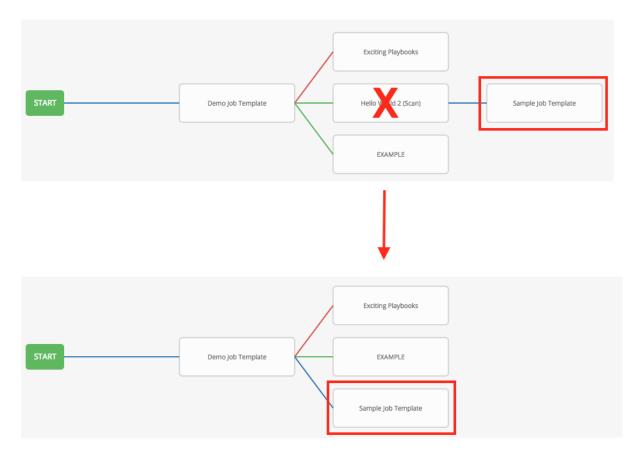
• A root node is set to ALWAYS by default and it not editable.

START	Demo Job Template

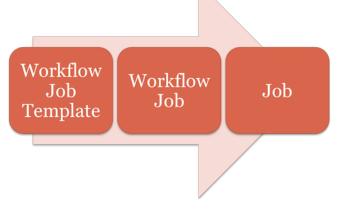
• A node can have multiple parents and children may be linked to any of the states of success, failure, or always. If always, then the state is neither success or failure. States apply at the node level, not at the workflow job template level. A workflow job will be marked as successful unless it is canceled or encounters an error.



• If you remove a job or workflow template within the workflow, the node(s) previously connected to those deleted, automatically get connected upstream and retains its edge type as in the example below:



- Prompts for inventory and surveys will apply to workflow nodes in workflow job templates.
- If you launch from the API, running a get command displays a list of warnings and highlights missing components. The basic workflow for a workflow job template is illustrated below.



- It is possible to launch several workflows simultaneously, and set a schedule for when to launch them. You can set notifications on workflows, such as when a job completes, similar to that of job templates.
- You can build a recursive workflow, but if Tower detects an error, it will stop at the time the nested workflow attempts to run.
- Artifacts gathered in jobs in the sub-workflow will not be passed to downstream nodes.
- An inventory can be set at the workflow level, or prompt for inventory on launch.

- When launched, all job templates in the workflow that have ask_inventory_on_launch=true will use the workflow level inventory.
- Job templates that do not prompt for inventory will ignore the workflow inventory and run against their own inventory.
- If a workflow prompts for inventory, schedules and other workflow nodes may provide the inventory.
- In a workflow convergence scenario, set_stats data will be merged in an undefined way, so it is recommended that you set unique keys.

17.2 Extra Variables

Also similar to job templates, workflows use surveys to specify variables to be used in the playbooks in the workflow, called extra_vars. Survey variables are combined with extra_vars defined on the workflow job template, and saved to the workflow job extra_vars. extra_vars in the workflow job are combined with job template variables when spawning jobs within the workflow.

Workflows utilize the same behavior (hierarchy) of variable precedence as Job Templates with the exception of three additional variables. Refer to the Ansible Tower Variable Precedence Hierarchy in the *Extra Variables* section of the Job Templates chapter of this guide. The three additional variables include:

Ansible	Tower
se	et_stats (i.e. artifacts)
custom facts	Job Artifacts Workflow Job Template extra variables Workflow Job Template Survey (defaults) Workflow Job Launch extra variables

Workflows included in a workflow will follow the same variable precedence - they will only inherit variables if they are specifically prompted for, or defined as part of a survey.

In addition to the workflow extra_vars, jobs and workflows ran as part of a workflow can inherit variables in the artifacts dictionary of a parent job in the workflow (also combining with ancestors further upstream in its branch). These can be defined by the set_stats Ansible module, version 2.2.2 or later.

If you use the set_stats module in your playbook, you can produce results that can be consumed downstream by another job, for example, notify users as to the success or failure of an integration run. In this example, there are two playbooks that can be combined in a workflow to exercise artifact passing:

• invoke_set_stats.yml: first playbook in the workflow:

```
- hosts: localhost
tasks:
    - name: "Artifact integration test results to the web"
    local_action: 'shell curl -F "file=@integration_results.txt" https://file.io'
    register: result
    - name: "Artifact URL of test results to Tower Workflows"
    set_stats:
        data:
            integration_results_url: "{{ (result.stdout|from_json).link }}"
```

• use_set_stats.yml: second playbook in the workflow

```
- hosts: localhost
tasks:
  - name: "Get test results from the web"
    uri:
        url: "{{ integration_results_url }}"
        return_content: true
        register: results
    - name: "Output test results"
    debug:
        msg: "{{ results.content }}"
```

The set_stats module processes this workflow as follows:

1. The contents of an integration results (example: integration_results.txt below) is first uploaded to the web.

the tests are passing!

- 2. Through the **invoke_set_stats** playbook, set_stats is then invoked to artifact the URL of the uploaded integration_results.txt into the Ansible variable "integration_results_url".
- 3. The second playbook in the workflow consumes the Ansible extra variable "integration_results_url". It calls out to the web using the uri module to get the contents of the file uploaded by the previous Job Template Job. Then, it simply prints out the contents of the gotten file.

MPLATES 3									
SEARCH			Q, KEY					+	ADD -
labels:workflow CLEAR ALL									
NAME ^	TYPE 🗢	DESCRIPTION \$	ACTIVITY	LABELS				AC	TIONS
set_stats Example Consumption	Job Template		•	× set_stats × workflow	R	m	2	(M ^A	Ē
set_stats Example Invocation	Job Template		•••	× set_stats × workflow	R	m	2	den .	Û
set_stats Workflow	Workflow Template		••	× set_stats × workflow	R	m	2	ø	Û
								ITEMS	1-30

Note: For artifacts to work, keep the default setting, per_host = False in the set_stats module.

17.3 Workflow States

The workflow job can have the following states (no Failed state):

- Waiting
- Running
- Success (finished)
- Cancel
- Error
- Failed

In the workflow scheme, canceling a job cancels the branch, while canceling the workflow job cancels the entire workflow.

17.4 Role-Based Access Controls

To edit and delete a workflow job template, you must have the admin role. To create a workflow job template, you must be an organization admin or a system admin. However, you can run a workflow job template that contains job templates you don't have permissions for. Similar to projects, organization admins can create a blank workflow and then grant an 'admin_role' to a low-level user, after which they can go about delegating more access and building the graph. You must have execute access to a job template to add it to a workflow job template.

Other tasks such as the ability to make a duplicate copy and re-launch a workflow can also be performed, depending on what kinds of permissions are granted to a particular user. Generally, you should have permissions to all the resources used in a workflow (like job templates) before relaunching or making a copy.

For more information on performing the tasks described in this section, refer to the Ansible Tower Administration Guide.

CHAPTER

EIGHTEEN

WORKFLOW JOB TEMPLATES

A workflow job template links together a sequence of disparate resources that accomplishes the task of tracking the full set of jobs that were part of the release process as a single unit. These resources may include:

- job templates
- · workflow templates
- project syncs
- inventory source syncs



The () menu opens a list of the workflow and job templates that are currently available. The workflow/job template list is sorted alphabetically by name but you can search by various fields and attributes of the workflow/job template. The workflow/job template list also enables you to launch, copy, and remove a job template.

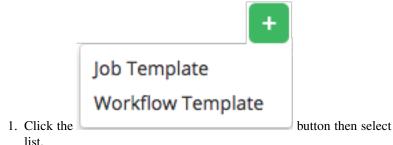
Only workflow templates have the Workflow Visualizer icon (

ARCH	Q KEY			
Demo Job Temp	Job Template			
ACTIVITY				
NVENTORY	Demo Inventory			
PROJECT	Demo Project	5,	r 4	Û
CREDENTIALS	🔩 Demo Credential			
AST MODIFIED	9/11/2018 11:33:43 PM by admin			
AST RAN	9/11/2018 11:33:43 PM			
xample Job Te	mplate Job Template			
VENTORY	Network Inventory Small	4	r Ph	÷
ROJECT	Demo Project	07	40	
AST MODIFIED	9/12/2018 5:09:48 AM by admin			
uper workflow	Workflow Template			
AST MODIFIED	9/12/2018 5:21:01 AM by admin	A G		Û
ABELS	run			

Note: Workflow templates can be used as building blocks for another workflow template. Many parameters in a workflow template allow you to enable **Prompt on Launch** that can be modified at the workflow job template level, and do not affect the values assigned at the individual workflow template level. For instructions, see the *Workflow Visualizer* section.

18.1 Create a Workflow Template

To create a new workflow job template:



button then select Workflow Job Template from the menu

NEW WORKFLOW JOB TEMP	LATE	0
DETAILS	NOTIFICATIONS COMPLETED JOBS SCHEDULES ADD SURVE	Y WORKFLOW VISUALIZER
* NAME	DESCRIPTION	
	PROMPT ON LAUNCH LABELS	OPTIONS Enable Concurrent Jobs
EXTRA VARIABLES ② YAML	JSON	
1		
		CANCEL SAVE

- 2. Enter the appropriate details into the following fields:
- Name: Enter a name for the workflow template.
- Description: Enter an arbitrary description as appropriate (optional).
- Organization: Optionally enter or search for an organization to associate the workflow.
- **Inventory**: Optionally enter or search for an inventory to be used with this workflow template from the inventories available to the currently logged in Tower user.
- **Prompt on Launch**: If selected, you can provide an inventory when this workflow template is launched, or when this workflow template is used within another workflow template.
- Labels: Supply optional labels that describe this workflow template, such as "dev" or "test". Labels can be used to group and filter workflow templates and completed jobs in the Tower display.
 - Labels are created when they are added to the Workflow Template. Labels are associated to a single
 Organization using the Project that is provided in the Workflow Template. Members of the Organization
 can create labels on a Workflow Template if they have edit permissions (such as an admin role).
 - Once the Workflow Template is saved, the labels appear in the Templates overview.
 - Click on the "x" beside a label to remove it. When a label is removed, and is no longer associated with a Workflow or Workflow Template, the label is permanently deleted from the list of Organization labels.
 - Jobs inherit labels from the Workflow Template at the time of launch. If a label is deleted from a Workflow Template, it is also deleted from the Job.

LABEL	S @		
×	test × scan × run		
			 _
Example јов	TEMPLATE		
	ITEMPLATE Demo Inventory		
INVENTORY			
Example Job INVENTORY PROJECT CREDENTIALS	Demo Inventory	39°	⁶ 2
INVENTORY	Demo Inventory Demo Example	A	¢۲

- Options: Check Enable Concurrent Jobs to allow simultaneous runs of this workflow.
- Extra Variables:
 - Pass extra command line variables to the playbook. This is the "-e" or "-extra-vars" command line parameter for ansible-playbook that is documented in the Ansible documentation at Passing Variables on the Command Line.
 - Provide key/value pairs using either YAML or JSON. These variables have a maximum value of precedence and overrides other variables specified elsewhere. An example value might be:

```
git_branch: production
release_version: 1.5
```

For more information about extra variables, refer to Extra Variables.

3. When you have completed configuring the workflow template, select Save.

Saving the template exits the Workflow Template page and the Workflow Visualizer opens to allow you to build a workflow. See the *Workflow Visualizer* section for further instructions. Otherwise, you may close the Workflow Visualizer to return to the Details tab of the newly saved template in order to review, edit, add permissions, notifications, schedules, and surveys, or view completed jobs and build a workflow template at a later time.

NEW WORKFLOW JOB TEMPLATE			0
DETAILS PERMISSIONS NOTIFICATION	ONS COMPLETED JOBS SCHEDULES ADD SURVEY	WORKFLOW VISUALIZER	
* NAME	DESCRIPTION	ORGANIZATION	
New Workflow job Template		Q Default	
INVENTORY @		OPTIONS	
Q	× run	Enable Concurrent Jobs @	
EXTRA VARIABLES 🛛 YAML JSON			
1			
		CANCEL SAVE	

You can verify the template is saved when the newly created workflow template appears on the list of templates at the bottom of the screen.

TEMPLATES					
SEARCH	Q, KEY				+
Demo Job Tem	late job Template				
INVENTORY	Demo Inventory				
PROJECT	Demo Project		3P	ළු	Û
CREDENTIALS	Demo Credential				
LAST MODIFIED	11/5/2018 10:15:50 AM by admin				
New Workflow	lob Template Workflow Template				
LAST MODIFIED	11/5/2018 10:44:57 PM by admin	A	421	#	ŵ
LABELS	nun				
template-b4844	31e job Template				
ACTIVITY					
INVENTORY	Inventory-b48431e				
PROJECT	project-b484431e		st.	2	Û
CREDENTIALS	q_ credential-machine- b484431e				
LAST MODIFIED	11/5/2018 10:25:42 AM by admin				
LAST RAN	11/5/2018 10:25:42 AM				
WF in WF Work	ow Template		0		
LAST MODIFIED	11/5/2018 12:22:05 PM by admin	Ħ	ළු	*	Ť
					ITEMS 1-4

Note: If a default inventory was specified on the workflow template, the inventory displays in the Templates list view.

Workflow 1	WORKFLOW
ACTIVITY	
INVENTORY	ben_inventory_test
LAST RUN	07/11/2017 11:30AM by jlaska
LABELS	Label Label Label Label VIEW MORE

18.2 Work with Permissions

Clicking on **Permissions** allows you to review, grant, edit, and remove associated permissions for users as well as team members.

DETAILS PERMISSIONS NOTI	FICATIONS COMPLETED JOBS SCHEDULES		
EARCH	Q KEY		
JSER 🔺	ROLE	TEAM ROLES	
admin	SYSTEM ADMINISTRATOR		
gdoge	× ADMIN		
doge	× EXECUTE		

Click the

k the **button** to create new permissions for this workflow template.

In this example, two users and one team have been selected and each have been granted permissions for this Workflow Template.

TEMPLATES / New Workflow Job Template / PER	MISSIONS				
New Workflow Job Template	NEW WORKFLOW JOB TEMPLAT			8	0
DETAILS PERMISSIONS NOTIFICATION	USERS				
SEARCH	SEARCH			Q KEY	•
USER 🔶	NAME 🔶		ORGANIZATION 🗢		TEAM ROLES
admin	Production Operations		Honey Dog, Inc.		
				ITEMS 1 - 1	ITEMS 1-1
TEMPLATES	2 Please assign roles to the se	elected users/teams		KEY	
SEARCH	Production Operatio TEAM	SELECT ROLES		×	•
		Admin			
Demo Job Template Job Template		Execute		AVE	
INVENTORY Demo Inventory		Read			
PROJECT Demo Project					A 4 🛍

Note that you do not have to choose between teams or users, and that you can assign permissions to both at the same time.

18.3 Work with Notifications

Clicking on Notifications allows you to review any notification integrations you have setup.

New Workflow Job Template					0
DETAILS PERMISSIONS SEARCH	NOTIFICATIONS COMPLETED JOBS SCHEE	Q KEY			GO TO NOTIFICATIONS TO ADD A NEW TEMPLATE
NAME *	TYPE 🗢		SUCCESS	FAILURE	
Email Notification	Email		OFF	OFF	
					ITEMS 1 - 1

If no notifications have been set up, click the **NOTIFICATIONS** link from above or inside the gray box to add or create a new notification.

New Workflow Job Template DETAILS PERMISSIONS NOTIFICATIONS	COMPLETED JOBS SCHEDULES	۵
		GOTO NOTIFICATIONS TO ADD A NEW TEMPOATE
	THIS LIST IS POPULATED BY NOTIFICATION TEMPLATES ADDED FROM THE NOTIFICATIONS BECTION	

Refer to Notifications for additional details on configuring various notification types.

18.4 View Completed Jobs

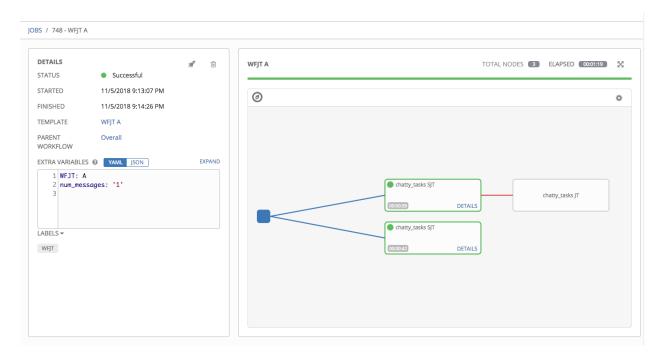
Clicking the **Completed Jobs** tab displays a list of all jobs that have run the workflow selected and various details about the job itself.

TEMPLATES / WFJT A / COMPLETED JOBS		Ø
WFJT A DETAILS PERMISSIONS NOTIFICATIONS COMPLETED JOBS SCHEDULES		Θ
SEARCH Q KEY		
• 748 - WFJT A Workflow Job STARTED 11/5/2018 9:13:07 PM FINISHED 11/5/2018 9:14:26 PM WORKFLOW JOB Overall LABELS Writ Vitility Vitility	đ	Ŵ
• 731 - WFJT A Workflowjob STARTED 11/5/2018 9:11:59 PM FINISHED 11/5/2018 9:13:06 PM WORKFLOW JOB Overall LABELS Wrjt Vertice Vertice	Ŕ	ش
673 - WFJT A Workflow Job STARTED 11/5/2018 9:10:27 PM FINISHED 11/5/2018 9:11:56 PM WORKFLOW JOB Overall LABELS Wrjt Vertice Vertice	Ŕ	Ŵ
		ITEMS 1-3

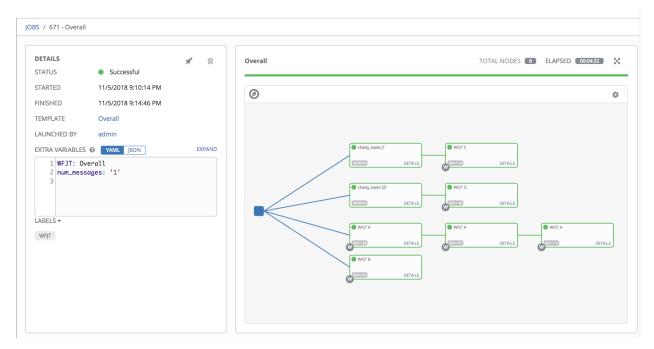
Note: If a workflow-level inventory was specified at run-time, the inventory name displays in the workflow job in the jobs list:



From this view, you can click the job ID - name of the workflow job and see its graphical representation. The example below shows the job details of the **748-WFJT A** workflow job.



Similarly, you can click the workflow template in which this workflow is used. In the example above, clicking the parent workflow template, **Overall**, takes you to its Job Details page and the graphical details of the nodes and statuses of each as they were launched.



The nodes noted with W are workflow templates while the ones not marked are job templates. Each node shows status and the duration it took for it to complete.

18.5 Work with Schedules

Clicking on Schedules allows you to review any schedules set up for this template.

DETAILS	PERMISSIONS NOTIFICATIONS COMPLETE	D JOBS SCHEDULES				
DETAILS		SCHEDULES				
EARCH		Q, K	EY			
N	IAME *	FIRST RUN \$	NEXT RUN \$	FINAL RUN 🗢	AC	TIONS
ON M	lonthly monitoring	10/15/2018 11:00:00 PM	10/15/2018 11:00:00 PM	1/15/2019 11:00:00 PM	æ	ŵ

From this view, you can select schedules to edit, turn on or off, or select multiple schedules to delete.

This screen displays a list of the schedules that are currently available for the selected workflow template. The Schedules list may be sorted and searched by any of the following criteria:

- Name: Clicking the schedule name opens the Edit Schedule dialog
- First Run: The first scheduled run of this task
- Next Run: The next scheduled run of this task
- Final Run: If the task has an end date, this is the last scheduled run of the task

Use the ON/OFF toggle next to the schedule name to enable/disable that schedule. Each schedule has a corresponding

Actions column that has options to allow editing (\checkmark) or deleting (\square) the schedule.

18.5.1 Schedule a Workflow Template

To create a new schedule:

- 1. From the Schedules screen, click the **button**.
- 2. Enter the appropriate details into the following fields:
- Name
- Start Date
- Start Time
- Local Time Zone: the entered Start Time should be in this timezone
- Repeat Frequency: the appropriate options display as the update frequency is modified

Note: Jobs are scheduled in UTC. Repeating jobs that runs at a specific time of day may move relative to a local timezone when Daylight Saving Time shifts occur.

The Schedule Description below displays the specifics of the schedule and a list of the scheduled occurrences in the selected Local Time Zone.

* START DATE * START TIME (HH24:MM:SS)	* TART DATE * START DATE *	aily workflow run						
Daly workflow run	bally workflow run OCAL TIME ZONE OCAL TIME ZONE PREPEAT FREQUENCY Day PROVINCY PROVINCY PROVI							
LOCAL TIME ZONE America/Denver America/Denve	OCAL TIME ZONE • REPEAT FREQUENCY Immerica/Denver Day equency Detrails • END Immerica/Denver • END Immerica/Denver<						<u></u>	0
America/Denver Day exequency perails EVERY *END 2 Days On Date 0 Date SCHEDULE DESCRIPTION every 2 days until July 17, 2017 OCCURRENCES (Limited to first 10) DATE FORMAT © LOCAL TIME O UTC 3/27/2017 20:45:00 MDT 3/27/2017 20:45:00 MDT 4/202017 20:45:00 MDT	Imperica/Derwer	Daily worknow run		□ 03/2//2017		20 . 45	 	~
REQUENCY DETAILS EVERY * END * END DATE 2 O DAYS On Date Image: Contract of Contract	Equency perails Every *END On Date *END DATE Inter (H+24-MM-SS) Inter (H+24-MM-SS) 3 • 45 • 0 Schedule description *every 2 days until july 17, 2017 Occurreences (Limited to first 10) DATE FORMAT © LOCAL TIME () UTC 3/27/2017 20/45:00 MDT 331/2017 20/45:00 MDT 3/37/2017 20/45:00 MDT 4/4/2017 20/45:00 MDT 4/4/2017 20/45:00 MDT 4/4/2017 20/45:00 MDT 4/12/2017 20/45:00 MDT *every 2/4/2017 20/45:00 MDT *every 2/4/2017 20/45:00 MDT *every 2/4/2017 20/45:00 MDT *every 2/4/2017 20/45:00 MDT *every 2/4/2017 20/45:00 MDT *every 2/4/2017 20/45:00 MDT *every 2	LOCAL TIME ZONE		* REPEAT FREQUENCY				
EVERY * END * END DATE 2 O DAVS On Date 07/17/2017 END TIME (HH24:MM:SS)	VRY • ND IND TIME (HH24:MM:SS) a a (Indate) CONDATE a (Indate)	America/Denver	•	Day	•			
Image:	Image: Davis Image: Davis <td< td=""><td>EQUENCY DETAILS</td><td></td><td></td><td></td><td></td><td></td><td></td></td<>	EQUENCY DETAILS						
END TIME (HH24:MM:SS) 3 3 3 45 3 5 6 5 5 5 5 5 5 5 5 5 5 5 5 5	END TIME (HH24:4MM:SS) 3): 45): 0 SCHEDULE DESCRIPTION every 2 days until July 17, 2017 OCCURRENCES (Limited to first 10) DATE FORMAT @ LOCAL TIME O UTC 3/27/2017 20:45:00 MDT 3/29/2017 20:45:00 MDT 4/2017 20:45:00 MDT 4/10/2017 20:45:0	EVERY		* END		* END DATE		
Bar Stepsel Bar Stepsel schedule bescription every 2 days until July 17, 2017 OCCURRENCES (Limited to first 10) DATE FORMAT	3 (): 45 (): 0 () SCHEDULE DESCRIPTION every 2 days until July 17, 2017 OCCURRENCES (Limited to first 10) DATE FORMAT () LOCAL TIME () UTC 3/27/2017 20:45:00 MDT 3/27/2017 20:45:00 MDT 4/22017 20:45:00 MDT 4/22017 20:45:00 MDT 4/22017 20:45:00 MDT 4/12/2017 20:	1	DAYS	On Date	Ψ.	07/17/2017		
		SCHEDULE DESCRIPTION every 2 days until July 17, 2017 DCCURRENCES (Limited to first 10) DATE FORMAT ()		rc				

3. When satisfied with the schedule specifics, click Save.

Once the schedule is saved, the list of schedules display for the associated workflow template.

DETAILS		COMPLETED JOBS SCHEDULES				¢
SEARCH		Q	KEY			+
	NAME *	FIRST RUN 🗢	NEXT RUN 🗢	FINAL RUN 🗢	ACT	TIONS
ON	Monthly monitoring	10/15/2018 11:00:00 PM	10/15/2018 11:00:00 PM	1/15/2019 11:00:00 PM	den .	ŵ
ON	Repeating Everyday	9/13/2018 3:00:00 PM	9/13/2018 3:00:00 PM	9/16/2018 3:00:00 PM	I	Û
					ſ	ITEMS

Use the **ON/OFF** toggle button to quickly activate or deactivate this schedule.

Note: If a workflow template used in a nested workflow has a survey, or the **Prompt on Launch** selected for the inventory option, the **PROMPT** button displays next to the **SAVE** and **CANCEL** buttons on the schedule form. Clicking the **PROMPT** button shows an optional INVENTORY step where you can provide or remove an inventory or skip this step without any changes.

18.6 Surveys

Workflows containing job types of Run or Check provide a way to set up surveys in the Workflow Job Template creation or editing screens. Surveys set extra variables for the playbook similar to 'Prompt for Extra Variables' does, but in a user-friendly question and answer way. Surveys also allow for validation of user input. Click the

ADD SURVEY

button to create a survey.

Use cases for surveys are numerous. An example might be if operations wanted to give developers a "push to stage" button they could run without advanced Ansible knowledge. When launched, this task could prompt for answers to questions such as, "What tag should we release?"

Many types of questions can be asked, including multiple-choice questions.

Note: Surveys are only available to those with Enterprise-level licenses.

ADD SURVEY

18.6.1 Create a Survey

To create a survey:

1. Click on the

button to bring up the Add Survey window.

TEMPLATES / New W	/orkflow Job Template		٩
New Workflow Jo	New Workflow Job Template SURVEY ON	G	
DETAILS P.	ADD SURVEY PROMPT	PREVIEW	
	* PROMPT	PLEASE ADD A SURVEY PROMPT ON THE LEFT.	
* NAME	Which group(s) should use this template?		
New Workflow J	DESCRIPTION		
LABELS @	Enter groups, one per line.		
× run	*ANSWER VARIABLE NAME @		
EXTRA VARIABLES	group_name		
1	*ANSWER TYPE		
	Text *		
	MINIMUM LENGTH MAXIMUM LENGTH		
	0 0		
	DEFAULT ANSWER		CANCEL SAVE
TEMPLATES	REQUIRED		
SEARCH	CANCEL + ADD	CANCEL	+ ADD -
NAME 🔶	TYPE C DESCRIPTION C	ACTIVITY LABELS	ACTIONS

Use the ON/OFF toggle button at the top of the screen to quickly activate or deactivate this survey prompt.

- 2. A survey can consist of any number of questions. For each question, enter the following information:
- Name: The question to ask the user.
- Description: (optional) A description of what's being asked of the user.

button to add the question.

- Answer Variable Name: The Ansible variable name to store the user's response in. This is the variable to be used by the playbook. Variable names cannot contain spaces.
- Answer Type: Choose from the following question types.
 - Text: A single line of text. You can set the minimum and maximum length (in characters) for this answer.
 - *Textarea*: A multi-line text field. You can set the minimum and maximum length (in characters) for this answer.
 - *Password*: Responses are treated as sensitive information, much like an actual password is treated. You can set the minimum and maximum length (in characters) for this answer.
 - *Multiple Choice (single select)*: A list of options, of which only one can be selected at a time. Enter the options, one per line, in the **Multiple Choice Options** box.
 - *Multiple Choice (multiple select)*: A list of options, any number of which can be selected at a time. Enter the options, one per line, in the **Multiple Choice Options** box.
 - Integer: An integer number. You can set the minimum and maximum length (in characters) for this answer.
 - Float: A decimal number. You can set the minimum and maximum length (in characters) for this answer.
- **Default Answer**: Depending on which type chosen, you can supply the default answer to the question. This value is pre-filled in the interface and is used if the answer is not provided by the user.
- Required: Whether or not an answer to this question is required from the user.



3. Once you have entered the question information, click the

A stylized version of the survey is presented in the Preview pane. For any question, you can click on the **Edit** button to edit the question, the **Delete** button to delete the question, and click and drag on the grid icon to rearrange the order of the questions.

- 4. Return to the left pane to add additional questions.
- 5. When done, click **Save** to save the survey.

	New Workflow Job Template			0
New Work	New Workflow Job Template SURVEY ON			0
DETAILS * NAME	ADD SURVEY PROMPT	PREVIEW		
New Wor	* PROMPT	*WHICH GROUP(S) SHOULD USE THIS TEMPLATE? Enter groups, one per line.		
LABELS @	DESCRIPTION		Ø	
EXTRA VARIA	*ANSWER VARIABLE NAME @	Click and hold down to drag the question to reorder it.		
	* ANSWER TYPE Choose an answer type			
	CANCEL + ADD			SAVE
TEMPLATE				
SEARCH			CANCEL	¥ ADD →
NAME *	TYPE \$ DESCRIPTION \$	ACTIVITY LABELS		ACTIONS

18.6.2 Optional Survey Questions

The **Required** setting on a survey question determines whether the answer is optional or not for the user interacting with it.

Behind the scenes, optional survey variables can be passed to the playbook in extra_vars, even when they aren't filled in.

- If a non-text variable (input type) is marked as optional, and is not filled in, no survey extra_var is passed to the playbook.
- If a text input or text area input is marked as optional, is not filled in, and has a minimum length > 0, no survey extra_var is passed to the playbook.
- If a text input or text area input is marked as optional, is not filled in, and has a minimum length === 0, that survey extra_var is passed to the playbook, with the value set to an empty string ("").

18.7 Workflow Visualizer

Ansible Tower 3.1 introduced the Workflow Visualizer (formerly *Workflow Editor*), which provides a graphical way of linking together job templates, workflow templates, project syncs, and inventory syncs to build a workflow template. Before building a workflow template, refer to the *Workflows* section for considerations associated with various scenarios on parent, child, and sibling nodes.

18.7.1 Build a Workflow

Make sure you have any combination of two of the following templates to build a workflow: jobs, project sync, or inventory sync. Each node is represented by a rectangle while the relationships and their associated edge types are represented by the line (or link) that connects them.

New Workflow Job Template	6
Ø	TOTAL NODES 🔟 🏟 ADD A TEMPLATE
	JOBS PROJECT SYNC INVENTORY SYNC
START	SEARCH Q KEY
	NAME 🔺
	O Demo Job Template
	ITEMS 1 • 1
	CANCEL SELECT
	CLOSE SAVE

3. On the right pane, select a template from the list of templates to add. To switch between jobs, project syncs, and inventory syncs, click the appropriate button above. Each template added represents a node.

Note: You will not be able to select job templates that don't have a default inventory when populating a workflow graph. Though credential is not required in a job template, you will not be able to choose a job template for your workflow if it has a credential that requires a password, unless the credential is replaced by a prompted credential.

4. Once a template is selected, the workflow begins to build, and you must specify the type of action to be taken for the selected template. This action is also referred to as *edge type*.

New Workflow Job Template	8
O TOTAL NODES D 🌣	ADD A TEMPLATE
	JOBS PROJECT SYNC INVENTORY SYNC
	SEARCH Q KEY
START	NAME 🔶
	Demo Job Template INFO
	ITEMS 1 - 1
	* RUN
	Always
	PROMPT CANCEL SELECT
	CLOSE SAVE

- 5. If the node is a root node, the edge type defaults to **Always** and is non-editable. For subsequent nodes, select one of the following scenarios (edge type) to apply to each:
- On Success: Upon successful completion, execute the next template.
- On Failure: Upon failure, execute a different template.
- Always: Continue to execute regardless of success or failure.
- 6. If a job template used in the workflow has **Prompt on Launch** selected for any of its parameters, a **Prompt** button appears, allowing you to change those values at the node level. Use the wizard to change the value(s) and click **Confirm**.

PROMPT	0
OTHER PROMPTS PREVIEW	
LIMIT	
15	
* VERBOSITY	
3 (Debug)	*
JOB TAGS	
× 3037	
CANCEL	кт

Likewise, if a workflow template used in the workflow has **Prompt on Launch** selected for the inventory option, use the wizard to supply the inventory at the prompt. If the parent workflow has its own inventory, it will override any inventory that is supplied here.

PROMPT INVENTORY PREVIEW		8
INVENTORY PREVIEW A This inventory is applied to all job template nodes that prompt for	an inventory.	
SEARCH	Q	KEY
NAME 🔺		
 Demo Inventory 		
<pre> <div class="xss">t</div>-inventory </pre>		
o <div class="xss" id="xss">test</div> -inventory		
O <div class="xss" id="xss">test</div> -smart-inventory		
e2e-17bbe884-inventory		
< 1 2 3 4 > PAGE 1 OF 4		ITEMS 1 - 5 OF 17
	CANCEL	NEXT

Note: For job templates with promptable fields that are required, but don't have a default, you must provide those values when creating a node before the **Select** button becomes enabled. The two cases that disable the **Select** button until a value is provided via the **Prompt** button: 1) when you select the **Prompt on Launch** checkbox in a job template, but do not provide a default, or 2) when you create a survey question that is required but don't provide a default answer. However, this is **NOT** the case with credentials. Credentials that require a password on launch are **not permitted** when creating a workflow node, since everything needed to launch the node must be provided when the node is created. So, if a job template prompts for credentials, Tower prevents you from being able to select a credential that requires a password.

You must also click **Select** when the prompt wizard closes in order to apply the changes at that node. Otherwise, any changes you make will revert back to the values set in the actual job template.

* RUN

Always		•
	PROMPT	SELECT

A template that is associated with each workflow node will run based on the selected run scenario as it proceeds. Click

the compass (**V**) icon to display the legend for each run scenario and their job types.

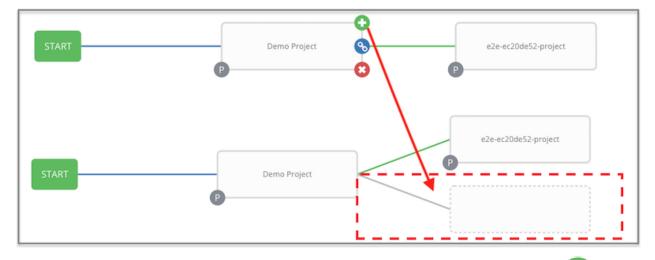


7. When done adding/editing a node, click **Select** to save any modifications and render it on the graphical view.

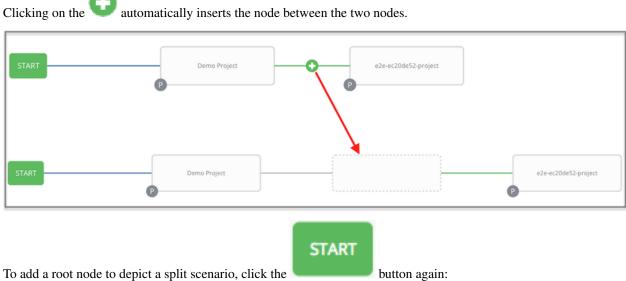
Hovering over a node allows you to add 🖸 another node, link to another node 📎, or delete 😳 the selected node.

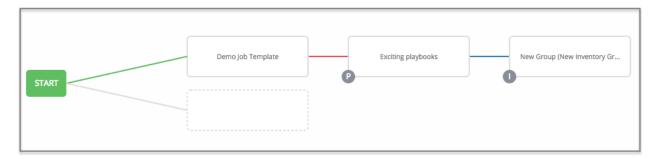
WORKFLOW VISUALIZER New Workflow Job Template	8
🙆 TOTAL NODES 💷 🔅	
START Demo job Template	
	CLOSE SAVE

You can add a sibling node by clicking the on the parent node:

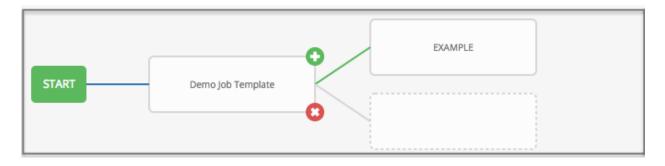


You can insert another node in between nodes by hovering over the line that connects the two until the appears.





At any node where you want to create a split scenario, hover over the node from which the split scenario begins and click the \bigcirc . This essentially adds multiple nodes from the same parent node, creating sibling nodes:



Note: When adding a new node, the **PROMPT** button applies to workflow templates as well. Workflow templates will prompt for inventory and surveys.

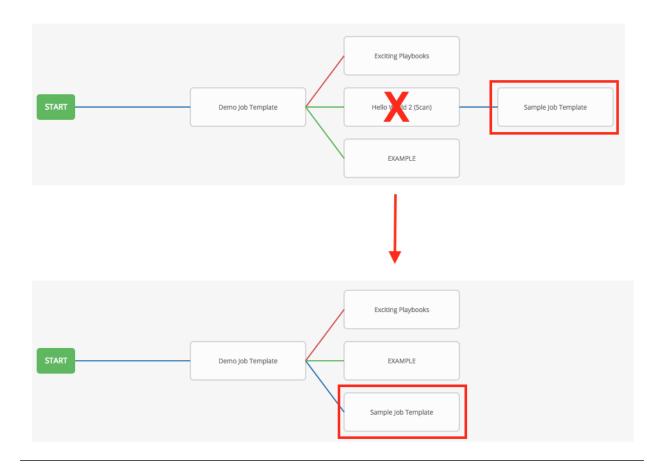
If you want to undo the last inserted node, click on another node without making a selection from the right pane. Or, click **Cancel** from the right pane.

Below is an example of a workflow that contains all three types of jobs that is initiated by a job template that if it fails to run, proceed to the project sync job, and regardless of whether that fails or succeeds, proceed to the inventory sync job.



Remember to refer to the Key at the top of the window to identify the meaning of the symbols and colors associated with the graphical depiction.

Note: In a workflow with a set of sibling nodes having varying edge types, and you remove a node that has a follow-on node attached to it, the attached node automatically joins the set of sibling nodes and retains its edge type:



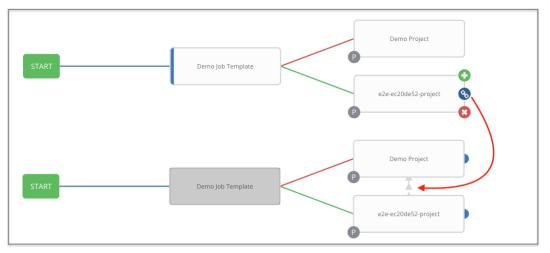
The following ways you can modify your nodes:

- If you want to edit a node, click on the node you want to edit. The right pane displays the current selections. Make your changes and click **Select** to apply them to the graphical view.
- To edit the edge type for an existing link (success/failure/always), click on the link. The right pane displays the current selection. Make your changes and click **Select** to apply them to the graphical view.

EDIT LINK to		
* RUN		
On Success		•
	CANCEL	SAVE

• To add a new link from one node to another, click the link icon that appears on each node. Doing this highlights the nodes that are possible to link to. These feasible options are indicated by the dotted lines. Invalid

options are indicated by grayed out boxes (nodes) that would otherwise produce an invalid link. The example below shows the **Demo Project** as a possible option for the **e2e-ec20de52-project** to link to, as indicated by the arrows:



• To remove a link, click the link and click the **Unlink** button.

EDIT LINK t	0		
* RUN			
On Failure			•
	UNLINK	CANCEL	SAVE

This button only appears in the right hand panel if the target or child node has more than one parent. All nodes must be linked to at least one other node at all times so you must create a new link before removing an old one.

Click the settings icon () to zoom, pan, or reposition the view. Alternatively, you can drag the workflow diagram to reposition it on the screen or use the scroll on your mouse to zoom.

8. When done with building your workflow template, click **Save** to save your entire workflow template and return to the new Workflow Template details page.

Important: Clicking **Close** on this pane will not save your work, but instead, closes the entire Workflow Visualizer and you will have to start over.

18.8 Launch a Workflow Template

To launch the workflow template:

-	-
	- X
	r 4 / 🗖

1. Access the workflow template from the **Templates** navigational link () or while in the Workflow Job Template Details view, scroll to the bottom to access it from a list of templates.

EARCH	Q KEY		
Demo Job Temj	Diate Job Template		
ACTIVITY			
INVENTORY	Demo Inventory		
PROJECT	Demo Project	37	2 🛍
CREDENTIALS	🔍 Demo Credential		
LAST MODIFIED	9/11/2018 11:33:43 PM by admin		
LAST RAN	9/11/2018 11:33:43 PM		
Example Job Te	mplate Job Template		
INVENTORY	Network Inventory Small	4	2 0
PROJECT	Demo Project	89	-0 Ш
LAST MODIFIED	9/12/2018 5:09:48 AM by admin		
Super workflov	V Workflow Template		
LAST MODIFIED	9/12/2018 5:21:01 AM by admin	A CL	њ û
LABELS	run		_

2. Click the launch () icon next to the workflow you want to launch.

Along with any extra variables set in the job template and survey, Tower automatically adds the same variables as those added for a job template upon launch. Additionally, Tower automatically redirects the web browser to the Jobs Details page for this job, displaying the progress and the results.

JOBS / 11 - Workflow						
DETAILS	Successful	A 🗈	Workflow	TOTAL NODES 🔳	ELAPSED 00:00:29	×
STARTED FINISHED	11/9/2018 12:13:58 PM		0			٥
TEMPLATE LAUNCHED BY EXTRA VARIABLES	Workflow admin	EXPAND	KEY — On Success — On Failure			
	YAML JSON	EAPAND	 Always Project Sync Inventory Sync Workflow 	Demo Job Template Demo Job Template DETAILS		

18.9 Copy a Workflow Template

Ansible Tower allows you the ability to copy a workflow template. If you choose to copy a workflow template, it **does not** copy any associated schedule, notifications, or permissions. Schedules and notifications must be recreated by the user or admin creating the copy of the workflow template. The user copying the workflow template will be granted the admin permission, but no permissions are assigned (copied) to the workflow template.

- 1. Access the workflow template that you want to copy from the **Templates** navigational link () or while in the Workflow Job Template Details view, scroll to the bottom to access it from a list of templates.

EARCH	Q KEY			
Demo Job Templ	ate Job Template			
ACTIVITY				
INVENTORY	Demo Inventory			
PROJECT	Demo Project	A	4	Û
CREDENTIALS	🔩 Demo Credential			
LAST MODIFIED	9/11/2018 11:33:43 PM by admin			
LAST RAN	9/11/2018 11:33:43 PM			
Example Job Ten	plate Job Template			
INVENTORY	Network Inventory Small	A	(2n	ŵ
PROJECT	Demo Project	<i></i>	40	W
LAST MODIFIED	9/12/2018 5:09:48 AM by admin			
Super workflow	Workflow Template			/
LAST MODIFIED	9/12/2018 5:21:01 AM by admin	1 4		Û
LABELS	run			

2. Click the button.

A new template opens with the name of the template from which you copied and a timestamp.

Replace the contents of the Name field with a new name, and provide or modify the entries in the other fields to complete this page.

3. Click **Save** when done.

Note: If a resource has a related resource that you don't have the right level of permission to, you cannot copy the resource, such as in the case where a project uses a credential that a current user only has *Read* access. However, for a workflow template, if any of its nodes uses an unauthorized job template, inventory, or credential, the workflow template can still be copied. But in the copied workflow template, the corresponding fields in the workflow template node will be absent.

18.10 Extra Variables

Note: Additional strict extra_vars validation was added in Ansible Tower 3.0.0. extra_vars passed to the job launch API are only honored if one of the following is true:

- They correspond to variables in an enabled survey
- ask_variables_on_launch is set to True

When you pass survey variables, they are passed as extra variables (extra_vars) within Tower. This can be tricky, as passing extra variables to a workflow template (as you would do with a survey) can override other variables being passed from the inventory and project.

For example, say that you have a defined variable for an inventory for debug = true. It is entirely possible that this variable, debug = true, can be overridden in a workflow template survey.

To ensure that the variables you need to pass are not overridden, ensure they are included by redefining them in the survey. Keep in mind that extra variables can be defined at the inventory, group, and host levels.

The following table notes the behavior (hierarchy) of variable precedence in Ansible Tower as it compares to variable precedence in Ansible.

Ansible Tower Variable Precedence Hierarchy (last listed wins)

Ansible	Tower
set_stats (i	.e. artifacts)
custom facts	Job Artifacts Workflow Job Template extra variables Workflow Job Template Survey (defaults) Workflow Job Launch extra variables

CHAPTER

NINETEEN

INSTANCE GROUPS

An Instance Group provides the ability to group instances in a clustered environment. Additionally, policies dictate how instance groups behave and how jobs are executed. The following view displays the capacity levels based on policy algorithms:

ARCH	Q KEY	
Group Eleven NSTANCES 1 RUNNING JOBS 0 TOTAL JOBS	USED CAPACITY COM 0%	Ô
Frouper Instance Group NSTANCES TOTAL JOBS OTAL JOBS	USED CAPACITY COM 0%	Û
ower NSTANCES 1 RUNNING JOBS 0 TOTAL JOBS	USED CAPACITY COM 0%	

19.1 Create an instance group

To create a new instance group:

 Click the icon from the Click the button. 	e left navigation menu to open the Ins	tance Groups configuration wir	ndow.
CREATE INSTANCE GROUP			8
DETAILS INSTANCES JOBS			
NAME 😧	POLICY INSTANCE MINIMUM	POLICY INSTANCE PERCENTAGE	0%
POLICY INSTANCE LIST		•	
Q			
		CANCEL	SAVE

3. Enter the appropriate details into the following fields:

- Name. Names must be unique and must not be named *tower*.
- **Policy Instance Minimum**. Enter the minimum number of instances to automatically assign to this group when new instances come online.
- **Policy Instance Percentage**. Use the slider to select a minimum percentage of instances to automatically assign to this group when new instances come online.
- Policy Instance List. Specify instances you want to assign to this group.

Note: Policy Instance fields are not required to create a new instance group. If you do not specify values, then the Policy Instance Minimum and Policy Instance Percentage default to 0.

4. Click Save.

Once the instance group is successfully created, the Details tab of the newly created instance group remains, which

allows you to review and edit your instance group information. This is the same menu that is opened if the Edit (\checkmark) button is clicked from the **Instance Group** link. You can also edit **Instances** and review **Jobs** associated with this instance group.

Instance Group 1		0
DETAILS INSTANCES JOBS		
NAME 🔞	POLICY INSTANCE MINIMUM	POLICY INSTANCE PERCENTAGE
Instance Group 1	2	25%
POLICY INSTANCE LIST		
Q		
		CANCEL
INSTANCE GROUPS 2		
SEARCH	Q KEY	•
Instance Group 1 INSTANCES 0 RUNNING JOBS 0		
tower INSTANCES 1 RUNNING JOBS 0		USED CAPACITY 0%
		ITEMS 1-2

19.1.1 Associate instances to an instance group

To associate instances to an instance group:

1. Click the **Instances** tab of the Instance Group window and click the

+	
	button.

2. Click the checkbox next to one or more available instances from the list to select the instance(s) you want to add to the instance group.

RUNNING JOBS	SELECT INSTANCE Instance Group 1			×		
	SELECTED Instance 1 X					
	SEARCH		Q KE	¢ 🗘		N
s 100	□ NAME ▼				1 50%	USED CA
	X Instance 1					
s 150	Instance 2				175%	USED CA
	Instance 3					
	Instance 4					
s (150)	< 1 2 3 4 5 6 7 8 9 10 > > PAGE 1 OF 15		ITEMS 1-1	0 OF 150	1 50%	USED CA
10 > » PAGE 1		CA	NCEL	SAVE		ITEMS 1-

3. In the following example, the instances added to the instance group displays along with information about their capacity.

Instance G	roup 1			
DETAILS	INSTANCES JOBS			
SEARCH		QKEY		•
ON)	Instance 1 RUNNING JOBS 100		CPU 8	USED CAPACITY
ON)	Instance 2 RUNNING JOBS 150		47 Forks	USED CAPACITY
OFF	Instance 3			
ON)	Instance 4 RUNNING JOBS 150		8 Forks RAM 52	USED CAPACITY 5%
< 1 2 3	4 5 6 7 8 9 10 > > PAGE 1 OF 15			ITEMS 1-10 OF 150 VIEW PER PAGE 10 v

This view also allows you to edit some key attributes associated with the instances in your instance group:

ON)	Instance 1 S RUNNING JOBS 100	Slider adjusts whether the Instance capacity algorithm yields less forks (towards the left) or yields more forks (towards the right)	CPU 8	16 Forks	RAM 52	USED CAPACITY
ON	Instance 2 MANUAL RUNNING JOBS 150 TOTAL JOBS 20	00)	CPU 8	47 Forks	RAM 52	USED CAPACITY
OFF	Instance 3					
ON)	RUNNING JOBS 150 TOTAL JOBS 20	Toggle takes the Instance online/offline and ensures that jobs won't be assigned to that instance	CPU 8 🔴	8 Forks	RAM 52	USED CAPACITY

19.1.2 View jobs associated with an instance group

To view the jobs associated with the instance group:

1. Click the **Jobs** tab of the Instance Group window.

Q KEY		
Ware Host PLAYBOOK RUN		
23/2017 352.55PM FINISHED 7/25/2017 5.53-45AM Iska		
pdate License Server	A	
cense Server		
en_machine		
Label Label Label		
eleted Data MANAGEMENTIDB 23/2017 9:52:53PM FINISHED 7/23/2017 9:53:45AM aska emplate Name cense Server <u>ben_machine</u> Label Label Label Label VEW MORE	Å	
ow 1 Workflow		
7/23/2017 9:52:53PM FINISHED 7/23/2017 9:53:45AM		
jlaska		
Template Name	1	
	69	
▲ ben_cloud		
Label Label Lorem ipsum dolor sit amet consectetur adipiscing elit Label Label Label Label Lorem ipsum dolor sit amet consectetur adipiscing elit Label		
Lorem ipsum dolor sit amet consectetur adipiscing elit Label Label Label VIEW LESS		
	Balzo17 9:52:53PM FINISHED 7/23/2017 9:53:45AM ska didate License Server ense Server ense Server elected Data mANAGEMENTICE B2/2017 9:52:53PM FINISHED 7/23/2017 9:53:45AM ska mplate Name ense Server Ver_machine abel Label Label Label VIEW MORE w1 worksrow 7/23/2017 9:53:45AM jiaska Template Name License Server Ver_machine abel Label Label 7/23/2017 9:53:45AM jiaska Template Name License Server Ver_machine License Server License	al 2017 9:52:53PM RINSHED 7/23/2017 9:53:45AM ska abel bel bel cented Data mawacemenrice 23/2017 9:52:53PM RINSHED 7/23/2017 9:53:45AM ska mplate Name abel Label Label Label view MORE w 1 worksrow 7/23/2017 9:53:45AM jiaska tenpidete Name License Server License Server Lic

2. Each job displays the job status, ID, and name; type of job, time started and completed, who started the job; and which template, inventory, and credential were used.

	-	Deleted Data MAN			Job type	The stress	
		7/23/2017 9:52:53PM	FINISHED	7/23/2017 9:53:45AN	1 ┥	— Timestamps	
		jlaska Template Name			 Template used 		ø
	INVENTORY	License Server 🛛 🗲				Inventory used	64
Job status	CREDENTIAL	🔍 ben_machine 🖌			——— Credential used		
JOD STATUS	LABELS	Label Label L	abel Labe	el Label VIEW MORE			

The instances are run in accordance with instance group policies. Refer to Instance Group Policies in the Ansible Tower Administration Guide.

CHAPTER

TWENTY

JOBS

A job is an instance of Tower launching an Ansible playbook against an inventory of hosts.

The Jobs link displays a list of jobs and their status-shown as completed successfully or failed, or as an active (running) job. Actions you can take from this screen include viewing the details and standard output of a particular job, relaunching jobs, or removing jobs.

JOBS		
SEARCH Q KEY		
Demo Job Template PLAYBOOK RUN STARTED 3/28/2018 1:51:18 PM FINISHED 3/28/2018 1:51:24 PM		
LAUNCHED BY admin JOB TEMPLATE Demo Job Template	3P	Û
INVENTORY Demo Inventory		
PROJECT Demo Project Demo Project SCM UPDATE		
STARTED 3/28/2018 1:51:11 PM FINISHED 3/28/2018 1:51:18 PM	3P	Û
PROJECT Demo Project		
		ITEMS 1-2

Starting with Ansible Tower 3.3, from the list view, you can re-launch the most recent job. You can re-run on all hosts in the specified inventory, even though some of them already had a successful run. This allows you to re-run the job without running the Playbook on them again. You can also re-run the job on all failed hosts. This will help lower the load on the Ansible Tower nodes as it does not need to process the successful hosts again.

The relaunch operation only applies to relaunches of playbook runs and does not apply to project/inventory updates, system jobs, workflow jobs, etc.

	Relaun parame	ch using eters	host
	Ð	A	Ŵ
RELAUNCH	ON		Ē
Failed			Ŵ
	Ð	A	Ê

- Selecting All relaunches all the hosts.
- Selecting Failed relaunches all failed and unreachable hosts.

When it relaunches, you remain on the same page.

Use the Tower Search feature to look up jobs by various criteria. For details about using the Tower Search, refer to the *Search* chapter.

Clicking on any type of job takes you to the Job Details View for that job, which consists of two sections:

- The Details pane provides information and status about the job
- The Standard Out pane displays the job processes and output

	Details pane		Standard Out pane	
DETAILS		# =	Network UI Project	
STATUS	Successful	177	PLAYS 2 TASKS 18 HOSTS 1 ELAPSED 00	0:00:06 📥 🔀
STARTED	9/12/2018 4:25:01 AM			
FINISHED	9/12/2018 4:25:07 AM		SEARCH	Q, KEY
JOB TYPE	Check			~ ~ * *
LAUNCHED BY	admin		1 Using /etc/ansible/ansible.cfg as config file	
PROJECT	Network UI Project		2	
INSTANCE GROUP	tower		3 PLAY [all] ***********************************	04:25:03
INSTANCE GROUP	tower		TASK [delete project directory before update] ************************************	04:25:03
			8 TASK [check repo using git] ************************************	04:25:03
			<pre>11 TASK [update project using git] ************************************</pre>	04:25:03
			14 TASK [Set the git repository version] ************************************	04:25:05
			<pre>17 TASK [update project using hg] **********************************</pre>	04:25:05
			19 20 TASK [Set the hg repository version] ************************************	04:25:05
			22 23 TASK [parse hg version string properly] ************************************	04:25:05
			25 26 mary fundate and under mark	04+25+06

20.1 Job Details - Inventory Sync

RESULTS		3P	Û	STANDARD OUT	×	*
NAME	Custom Inventory (inventory-custom - 34) - inventory-custom - 280					
STATUS	Successful			14.487 INF0 14.569 INF0 15.043 INF0	Updating inventory 8: inventory-custom Reading Ansible inventory source: /tmp/awx_inventory_paBjBp/tmpdn3C9B Processing JSNN output	
EXPLANATION				15.044 INFO	Loaded 1 groups, 5 hosts	
				15.161 WARNING 15.164 INFO	Group "Custom Inventory" from v1 API is not deleted by overwrite Group "Custom Inventory" from v1 API child group/host connections preserved	
LICENSE ERROR	False			15.185 INFO	Inventory variables unmodified	
STARTED	9/27/2017 8:28:40 PM				Group "hosts" variables unmodified Host "host-00" variables unmodified	
				15.212 INF0	Host "host-01" variables unmodified	
INISHED	9/27/2017 8:28:59 PM			15.212 INF0 15.212 INF0	Host "host-02" variables unmodified Host "host-03" variables unmodified	
ELAPSED	18.918 seconds			15.212 INF0	Host "host-04" variables unmodified	
				15.235 INF0 15.235 INF0	Host "host-00" already in group "hosts" Host "host-01" already in group "hosts"	
LAUNCH TYPE	Manual			15.235 INF0 15.235 INF0	Host "host-02" already in group "hosts" Host "host-03" already in group "hosts"	
SOURCE	Custom Script			15.235 INF0	Host "host-04" already in group "hosts"	
				15.529 INF0	Inventory import completed for Custom Inventory (inventory-custom - 34) - inventory-custom - 2	80 in
OVERWRITE	True					
OVERWRITE	False					
ARS						

20.1.1 Details

The **Details** pane shows the basic status of the job and its start time. The icons at the top right corner of the **Details** pane allow you to relaunch () or delete () the job.

The Details pane also provides details on the job execution:

- Name: The name of the associated inventory group.
- Status: Can be any of the following:
 - Pending The inventory sync has been created, but not queued or started yet. Any job, not just inventory source syncs, will stay in pending until it's actually ready to be run by the system. Reasons for inventory source syncs not being ready include dependencies that are currently running (all dependencies must be completed before the next step can execute), or there is not enough capacity to run in the locations it is configured to.
 - Waiting The inventory sync is in the queue waiting to be executed.
 - Running The inventory sync is currently in progress.
 - Successful The inventory sync job succeeded.
 - Failed The inventory sync job failed.
- Explanation: Describes reason(s) for failure.
- License Error: Only shown for Inventory Sync jobs. If this is *True*, the hosts added by the inventory sync caused Tower to exceed the licensed number of managed hosts.
- Started: The timestamp of when the job was initiated by Tower.
- Finished: The timestamp of when the job was completed.
- Elapsed: The total time the job took.
- Launch Type: Manual, Scheduled, or Dependency

- Credential: The credential used in this inventory sync.
- **Source**: The type of cloud inventory.
- **Overwrite**: If *True*, any hosts and groups that were previously present on the external source but are now removed, are removed from the Tower inventory. Hosts and groups that were not managed by the inventory source are promoted to the next manually created group or if there is no manually created group to promote them into, they are left in the "all" default group for the inventory. If *False*, local child hosts and groups not found on the external source remain untouched by the inventory update process.
- **Overwrite Vars**: If *True*, all variables for child groups and hosts are removed and replaced by those found on the external source. If *False*, a merge was performed, combining local variables with those found on the external source.

By clicking on these items, where appropriate, you can view the corresponding job templates, projects, and other Tower objects.

20.1.2 Standard Out

The **Standard Out** pane shows the full results of running the Inventory Sync playbook. This shows the same information you would see if you ran it through the Ansible command line, and can be useful for debugging. The icons at

the top right corner of the Standard Out pane allow you to toggle the output as a main view () or to download the output ().

Starting in Ansible Tower 3.3, the ANSIBLE_DISPLAY_ARGS_TO_STDOUT is set to False by default for all playbook runs. This matches Ansible's default behavior. This causes Tower to no longer display task arguments in task headers in the Job Detail interface to avoid leaking certain sensitive module parameters to stdout. If you wish to restore the prior behavior (despite the security implications), you can set ANSIBLE_DISPLAY_ARGS_TO_STDOUT to True via the AWX_TASK_ENV configuration setting. For more details, refer to the ANSIBLE_DISPLAY_ARGS_TO_STDOUT.

20.2 Job Details - SCM

TAILS		A 🗊	e2e-ae53906d-project
ATUS	Successful		PLAYS 2 TASKS 18 HOSTS 11 ELAPSED 00:00:03 🚣
ARTED	9/7/2018 8:57:42 AM		
IISHED	9/7/2018 8:57:46 AM		SEARCH Q
3 TYPE	Check		~ * *
JNCHED BY	admin		45 ok: [localhost] => {
DJECT	e2e-ae53906d-project		46 "msg": "Repository Version 347e44fea036c94d5f60e544de006453ee5c71ad"
TANCE GROUP	tower		47 }
INITEL GROOT	tower		48
			49 TASK [Write Repository Version] ************************************
			50 changed: [localhost]
			51 52 PLAY [all] ***********************************
			53
			54 TASK [detect requirements.yml] ************************************
			55 skipping: [localhost]
			56
			57 TASK [fetch galaxy roles from requirements.yml] ************************************
			58 skipping: [localhost]
			59
			60 TASK [fetch galaxy roles from requirements.yml (forced update)] ************************************
			61 skipping: [localhost] 62
			62 63 PLAY RECAP ************************************
			64 localhost : ok=4 changed=2 unreachable=0 failed=0

20.2.1 Details

The Details pane shows the basic status of the job and its start time. The icons at the top right corner of the Details

pane allow you to relaunch (\square) or delete (\square) the job.

The Details pane provides details on the job execution:

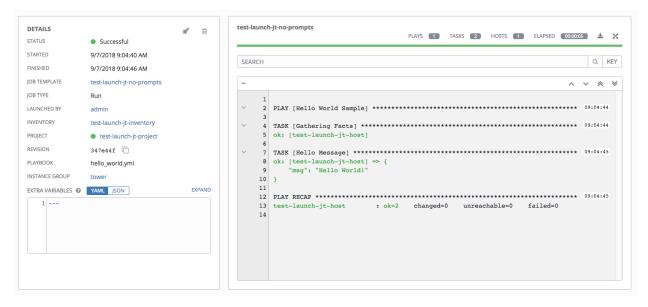
- Name: The name of the associated inventory group.
- Status: Can be any of the following:
 - Pending The SCM job has been created, but not queued or started yet. Any job, not just SCM jobs, will stay in pending until it's actually ready to be run by the system. Reasons for SCM jobs not being ready include dependencies that are currently running (all dependencies must be completed before the next step can execute), or there is not enough capacity to run in the locations it is configured to.
 - Waiting The SCM job is in the queue waiting to be executed.
 - Running The SCM job is currently in progress.
 - Successful The last SCM job succeeded.
 - Failed The last SCM job failed.
- Started: The timestamp of when the job was initiated by Tower.
- Finished: The timestamp of when the job was completed.
- Elapsed: The total time the job took.
- Launch Type: Manual or Scheduled.
- **Project**: The name of the project.

By clicking on these items, where appropriate, you can view the corresponding job templates, projects, and other Tower objects.

20.2.2 Standard Out

The **Standard Out** pane shows the full results of running the SCM Update. This shows the same information you would see if you ran it through the Ansible command line, and can be useful for debugging. The icons at the top right corner of the Standard Out pane allow you to toggle the output as a main view () or to download the output ().

20.3 Job Details - Playbook Run



The Job Details View for a Playbook Run job is also accessible after launching a job from the Job Templates page.

20.3.1 Details

The **Details** pane shows the basic status of the job and its start time. The icons at the top right corner of the **Details**

pane allow you to relaunch (\checkmark) or delete (\blacksquare) the job.

The **Details** pane provides details on the job execution:

- Status: Can be any of the following:
 - Pending The playbook run has been created, but not queued or started yet. Any job, not just playbook runs, will stay in pending until it's actually ready to be run by the system. Reasons for playbook runs not being ready include dependencies that are currently running (all dependencies must be completed before the next step can execute), or there is not enough capacity to run in the locations it is configured to.
 - Waiting The playbook run is in the queue waiting to be executed.
 - Running The playbook run is currently in progress.
 - Successful The last playbook run succeeded.
 - Failed The last playbook run failed.
- Template: The name of the job template from which this job was launched.
- Started: The timestamp of when the job was initiated by Tower.

- Finished: The timestamp of when the job was completed.
- Elapsed: The total time the job took.
- Launch By: The name of the user, job, or scheduled scan job which launched this job.
- Inventory: The inventory selected to run this job against.
- Machine Credential: The name of the credential used in this job.
- Verbosity: The level of verbosity set when creating the job template.
- Extra Variables: Any extra variables passed when creating the job template are displayed here.

By clicking on these items, where appropriate, you can view the corresponding job templates, projects, and other Tower objects.

20.3.2 Standard Out Pane

The **Standard Out** pane shows the full results of running the Ansible playbook. This shows the same information you would see if you ran it through the Ansible command line, and can be useful for debugging. You can view the event summary, host status, and the host events. The icons at the top right corner of the Standard Out pane allow you to toggle the output as a main view (

Events Summary

The events summary captures a tally of events that were run as part of this playbook:

- the number of plays
- · the number of tasks
- the number of hosts
- the elapsed time to run the job template

Network U	Il Project PLAYS 2 TASKS 18 HOSTS 1 ELAPSED 00:00:06 📩 🏂
SEARCH	Q KEY
1	v ≈ v × v × v × v
2	
4 5	TASK [delete project directory before update] ************************************

Host Status Bar

The host status bar runs across the top of the **Standard Out** pane. Hover over a section of the host status bar and the number of hosts associated with that particular status displays.

Remove VMWare Host 🕢	ОК [2]	PLAYS 1	TASKS 8	HOSTS 15	ELAPSED 00:00:05	*	К Л И И
SEARCH						Q	KEY

Search

Use the Tower Search to look up specific events, hostnames, and their statuses. To filter only certain hosts with a particular status, specify one of the following valid statuses:

- **Changed**: the playbook task actually executed. Since Ansible tasks should be written to be idempotent, tasks may exit successfully without executing anything on the host. In these cases, the task would return Ok, but not Changed.
- Failed: the task failed. Further playbook execution was stopped for this host.
- OK: the playbook task returned "Ok".
- Unreachable: the host was unreachable from the network or had another fatal error associated with it.
- Skipped: the playbook task was skipped because no change was necessary for the host to reach the target state.

The example below shows a search with only failed hosts.

Rem	ove VI	MWare Host ? OK 2 PLAYS 1 TASKS 8 HOSTS 15 ELAPSED 00:00:05	K 7 2 2
SEA	RCH	Q	KEY
×	or.stdo	ut.icontains:failed CLEAR ALL	
÷	Ξ		O
	1		
~	2	PLAY [add hosts to inventory] ************************************	
	3		
~	4	TASK [setup] ************************************	
	6		
~	7	TASK [create inventory] ************************************	4
~	18 19	RUNNING HANDLER [single host handler] ************************************	

For more details about using the Tower Search, refer to the Search chapter.

Standard output view

The standard output view displays all the events that occur on a particular job. By default, all rows are expanded so that all the details are displayed. Use the collapse-all button (-) to switch to a view that only contains the headers for plays and tasks. Click the (+) button to view all lines of the standard output.

Alternatively, you can display all the details of a specific play or task by clicking on the arrow icons next to them. Click an arrow from sideways to downward to expand the lines associated with that play or task. Click the arrow back to the sideways position to collapse and hide the lines.



Things to note when viewing details in the expand/collapse mode:

- Each displayed line that is not collapsed has a corresponding line number and start time.
- An expand/collapse icon is at the start of any play or task after the play or task has completed.
- If querying for a particular play or task, it will appear collapsed at the end of its completed process.
- In some cases, an error message will appear, stating that the output may be too large to display. This occurs when there are more than 4000 events. Use the search and filter for specific events to bypass the error.
- Hover over an event line in the **Standard Out** view, a tooltip displays above that line, giving the total hosts affected by this task and an option to view further details about the breakdown of their statuses.

DETAILS	A 🔛	Ī	Ren	nove	/MWare Host ? PLAYS 1 TASKS 8 HOSTS 15 ELAPSED 00:00:05		K. ⊻
TATUS	Successful						
TARTED	7/1/2016 1:00:07AM		SE	ARCH		Q	KE
INISHED	7/1/2016 1:00:12 AM						
EMPLATE	Update License Server 🔞		-	ŀ	· · · · · · · · · · · · · · · · · · ·	~	×
OB TYPE	Playbook Run		\sim	1	PLAY [Remove VMWare Host]	00:00):05
AUNCHED BY	User		>	2 3	GATHE TOTAL HOSTS 2	00:00):01
VENTORY	License Server		~	15 16	TASK: [ansiblelicense install required packages via yum]	00:00	0:01
ROJECT	ansible/ansible-it		Ť	17	ok: [74.207.226.226]		
EVISION	00000000			18 19	ok: [74.207.226.226]		
LAYBOOK	store.vml		>	20	TASK: [ansiblelicense update setuptools] ************************************	00:00	J:01
REDENTIAL	ben_cloud			28		00.00	0.04
IMIT	Store		>	29	TASK: [ansiblelicense update pip] **********************************	00:00	2:01
ERBOSITY	Update License Server		~	35 36 37	TASK: [ansiblelicense create unprivileged user for ansiblelicense] ************************************	00:00):01
ISTANCES	Instance 01 ISOLATED			38 39	skipping: [74.207.226.226]		
XTRA VARIABLES	0	EXPAND	\sim	39 40	TASK: [ansiblelicense configure ansiblelicense directory permissions] ************************************	00:00):01
2 vmware_l	ne: 1453164676 nost: cent7issue			41 42 43	changed: [74.287.226.226] changed: [74.287.226.226]		
3 4			>	44	TASK: [ansiblelicense enable maintenance page] ************************************	00:00	J:01
RTIFACTS 🕜		EXPAND	~	65 66 67	TASK: [ansiblelicense check ssh connection to github] ************************************	00:00):01

Click on a line of an event from the **Standard Out** pane and a **Host Events** dialog displays in a separate window. This window shows the host that was affected by that particular event.

Host Events

The **Host Events** dialog shows information about the host affected by the selected event and its associated play and task:

- the Host
- the Status
- a unique ID
- a Created time stamp
- the name of the Play
- the name of the Task
- if applicable, the Ansible Module for the task, and any arguments for that module
- the **Standard Out** of the task

JOB TYPE R 2 "_onsible_parsed": true, 3 "_onsible_no_log": false, LAUNCHED BY B 4 "changed": true, C 5 "state": "absent", !b6 INVENTORY L 6 "diff": { ?easter": "after": { PROJECT 8 "path": "/tmp/tower", .easter": "absent" REVISION 4 .easter": "absent" .easter": "absent" PLAYBOOK b .cLOSE .crue	HOSTS ELAPS: Charlenge to the second
DETAILS PLAY Build and push Tower Development Container Image STATUS TASK Cleanup Tower STATED MODULE file FINISHED goon JOB TYPE B 1 { 	Nn9d2439ce6cf9: a3: Preparing\n3
STATUS TASK Cleanup Tower STATED MODULE file FINISHED 9 JSON JOB TYPE 8 1 { 2 "_ansible_parsed": true, 3 3 "_ansible_no_log": false, 1 LAUNCHED BY 8 4 "changed": true, 5 "state": "absent", 1 1 7 "after": { 1 PROJECT 8 "path": "/tmp/tower", 1 9 _state": "absent" 1 REVISION 4 CLOSE 1	Nn9d2439ce6cf9: a3: Preparing\n3
STARTED 9 MODULE file FINISHED 9 JSON JOB TEMPLATE B 1 { JOB TYPE R 2 "_ansible_parsed": true, idl JOB TYPE R 3 "_ansible_no_log": false, idl LAUNCHED BY B 4 "changed": true, ist S "state": "absent", ist ist INVENTORY Li 6 "diff": { ist PROJECT 8 "path": "/tmp/tower", ist 9 "state": "absent" ist PLAYBOOK CLOSE ist	Nn9d2439ce6cf9: a3: Preparing\n3
FINISHED 9 JSON JOB TYPE R 2 "_ansible_parsed": true, JOB TYPE R 2 "_ansible_no_log": false, LAUNCHED BY B 4 "changed": true, S "state": "absent", ibe INVENTORY Lt 6 "diff": { PROJECT 0 "state": "/tmp/tower", 9 10 1 "closent" ibe PLAYBOOK CLOSE :cr	.a3: Preparing\n3
JOB TEMPLATE JOB TYPE LAUNCHED BY REVISION JOB TYPE REVISION JOB TYPE REVISION CLOSE JOB TYPE REVISION CLOSE JOB TYPE REVISION CLOSE JOB TYPE REVISION CLOSE JOB TYPE REVISION CLOSE JOB TYPE JOB TYPE REVISION CLOSE JOB TYPE JOB TYPE REVISION CLOSE JOB TYPE JOB TYPE REVISION CLOSE JOB TYPE JOB TYPE REVISION CLOSE JOB TYPE JOB TY	.a3: Preparing\n3
C 1 {	.a3: Preparing\n3
IOB TYPE R 2 "_ansible_parsed": true, Insible_no_log": false, IAUNCHED BY B 4 "changed": true, Insible_no_log": false, INVENTORY I 6 "diff": { Insible_no_log": false, PROJECT 8 "path": "/tmp/tower", Insible_no_log": false, Insible_no_log: REVISION Insible_no_log: Insible_no_log: Insible_no_log: Insible_no_log: PLAYBOOK Insible_no_log: Insible_no_log: Insible_no_log: Insible_no_log: CLOSE Insible_no_log: Insible_no_log: Insible_no_log: Insible_no_log:	.a3: Preparing\n3
a 9 "state": "absent" ug	le2: Waiting\nfc de30951a95: Wait ing\n35028ab89a7
PLAYBOOK bi	• Tower] *******
PEDENTIAL	e, "path": "/tmp
Engineering Tower Repo Achine Credential Achine Credentia Achine Credential Achine Creden	****
LIMIT localhost 39 0 failed=0	anged=8 unrea

To view the results in JSON format, click on the **JSON** tab.

20.4 Ansible Tower Capacity Determination and Job Impact

The Ansible Tower capacity system determines how many jobs can run on an instance given the amount of resources available to the instance and the size of the jobs that are running (referred to as *Impact*). The algorithm used to determine this is based entirely on two things:

- How much memory is available to the system (mem_capacity)
- How much CPU is available to the system (cpu_capacity)

Capacity also impacts Instance Groups. Since Groups are made up of instances, likewise, instances can be assigned to multiple groups. This means that impact to one instance can potentially affect the overall capacity of other Groups.

Instance Groups (not instances themselves) can be assigned to be used by jobs at various levels (see Clustering). When the Task Manager is preparing its graph to determine which group a job will run on, it will commit the capacity of an Instance Group to a job that hasn't or isn't ready to start yet.

Finally, in smaller configurations, if only one instance is available for a job to run, the Task Manager will allow that job to run on the instance even if it pushes the instance over capacity. This guarantees that jobs themselves won't get stuck as a result of an under-provisioned system.

Therefore, Capacity and Impact is not a zero-sum system relative to jobs and instances/Instance Groups.

For information on sliced jobs and their impact to capacity, see Job slice execution behavior.

20.4.1 Resource determination for capacity algorithm

The capacity algorithms are defined in order to determine how many forks a system is capable of running simultaneously. This controls how many systems Ansible itself will communicate with simultaneously. Increasing the number of forks a Tower system is running will, in general, allow jobs to run faster by performing more work in parallel. The trade-off is that this will increase the load on the system, which could cause work to slow down overall.

Tower can operate in two modes when determining capacity. mem_capacity (the default) will allow you to overcommit CPU resources while protecting the system from running out of memory. If most of your work is not CPUbound, then selecting this mode will maximize the number of forks.

Memory relative capacity

mem_capacity is calculated relative to the amount of memory needed per fork. Taking into account the overhead for Tower's internal components, this comes out to be about 100MB per fork. When considering the amount of memory available to Ansible jobs, the capacity algorithm will reserve 2GB of memory to account for the presence of other Tower services. The algorithm formula for this is:

(mem - 2048) / mem_per_fork

As an example:

(4096 - 2048) / 100 == ~20

Therefore, a system with 4GB of memory would be capable of running 20 forks. The value mem_per_fork can be controlled by setting the Tower settings value (or environment variable) SYSTEM_TASK_FORKS_MEM, which defaults to 100.

CPU relative capacity

Often, Ansible workloads can be fairly CPU-bound. In these cases, sometimes reducing the simultaneous workload allows more tasks to run faster and reduces the average time-to-completion of those jobs.

Just as the Tower mem_capacity algorithm uses the amount of memory need per fork, the cpu_capacity algorithm looks at the amount of CPU resources is needed per fork. The baseline value for this is 4 forks per core. The algorithm formula for this is:

```
cpus * fork_per_cpu
```

For example, a 4-core system:

4 * 4 == 16

The value fork_per_cpu can be controlled by setting the Tower settings value (or environment variable) SYSTEM_TASK_FORKS_CPU which defaults to 4.

20.4.2 Capacity job impacts

When selecting the capacity, it's important to understand how each job type affects capacity.

It's helpful to understand what forks mean to Ansible: https://www.ansible.com/blog/ansible-performance-tuning (see the section on "Know Your Forks").

The default forks value for Ansible is 5. However, if Tower knows that you're running against fewer systems than that, then the actual concurrency value will be lower.

When a job is run, Tower will add 1 to the number of forks selected to compensate for the Ansible parent process. So if you are running a playbook against 5 systems with a forks value of 5, then the actual forks value from the perspective of Job Impact will be 6.

Impact of job types in Tower

Jobs and Ad-hoc jobs follow the above model, forks + 1. If you set a fork value on your job template, your job capacity value will be the minimum of the forks value supplied, and the number of hosts that you have, plus one. The plus one is to account for the parent Ansible process.

Instance capacity determines which jobs get assigned to any specific instance. Jobs and ad hoc commands use more capacity if they have a higher forks value.

Other job types have a fixed impact:

- Inventory Updates: 1
- Project Updates: 1
- System Jobs: 5

If you don't set a forks value on your job template, your job will use Ansible's default forks value of five. Even though Ansible defaults to five forks, it will use fewer if your job has fewer than five hosts. In general, setting a forks value higher than what the system is capable of could cause trouble by running out of memory or over-committing CPU. So, the job template fork values that you use should fit on the system. If you have playbooks using 1000 forks but none of your systems individually has that much capacity, then your systems are undersized and at risk of performance or resource issues.

Selecting the right capacity

Selecting a capacity out of the CPU-bound or the memory-bound capacity limits is, in essence, selecting between the minimum or maximum number of forks. In the above examples, the CPU capacity would allow a maximum of 16 forks while the memory capacity would allow 20. For some systems, the disparity between these can be large and often times you may want to have a balance between these two.

The instance field capacity_adjustment allows you to select how much of one or the other you want to consider. It is represented as a value between 0.0 and 1.0. If set to a value of 1.0, then the largest value will be used. The above example involves memory capacity, so a value of 20 forks would be selected. If set to a value of 0.0 then the smallest value will be used. A value of 0.5 would be a 50/50 balance between the two algorithms which would be 18:

16 + (20 - 16) * 0.5 == 18

To view or edit the capacity in the Tower user interface, select the Instances tab of the Instance Group.

Instance G	nstance Group 1					
DETAILS	INSTANCES RUNNING JOBS					
SEARCH	Q KEY 🔯		+			
ON)	Instance 1 RUNNING JOBS (100)	CPU 8 6 Forks USED CAPACITY	50%			
ON)	Instance 2 Slider adjusts whether the Instance capacity algorithm yields less RUNNING JOBS (50) forks (towards the left) or yields more forks (towards the right)	47 Forks CPU 8 RAM 52 USED CAPACITY	15%			
OFF	Instance 3					
ON)	Instance 4 RUNNING JOBS (150)	8 Forks CPU 8 RAM 52 USED CAPACITY	5%			
< 1 2 3	4 5 6 7 8 9 10 > > PAGE 1 OF 15	ITEMS 1-10 OF 150 VIEW F	PER PAGE 10 V			

CHAPTER

TWENTYONE

NOTIFICATIONS

A Notifier is an instance of a Notification type (Email, Slack, Webhook, etc.) with a name, description, and a defined configuration.

For example:

- A username, password, server, and recipients are needed for an Email notifier
- The token and a list of channels are needed for a Slack notifier
- The URL and Headers are needed for a Webhook notifier

A Notification is a manifestation of the notifier; for example, when a job fails, a notification is sent using the configuration defined by the Notifier.

At a high level, the typical flow for the notification system works as follows:

- A user creates a notifier to the Tower REST API at the /api/v2/notifiers endpoint (either through the API or through the Tower UI).
- A user assigns the notifier to any of the various objects that support it (all variants of job templates as well as organizations and projects) and at the appropriate trigger level for which they want the notification (error, success, or any). For example a user may wish to assign a particular Notifier to trigger when Job Template 1 fails. In which case, they will associate the notifier with the job template at /api/v2/job_templates/n/ notifiers_error API endpoint.

21.1 Notifier Hierarchy

Notifiers assigned at certain levels will inherit notifiers defined on parent objects as such:

- Job Templates will use notifiers defined on it as well as inheriting notifiers from the Project used by the Job Template and from the Organization that it is listed under (via the Project).
- Project Updates will use notifiers defined on the project and will inherit notifiers from the Organization associated with it
- Inventory Updates will use notifiers defined on the Organization that it is listed under
- · Ad-hoc commands will use notifiers defined on the Organization that the inventory is associated with

21.2 Workflow

When a job succeeds or fails, the error or success handler will pull a list of relevant notifiers using the procedure defined above. It will then create a Notification object for each one containing relevant details about the job and then sends it to the destination (email addresses, slack channel(s), sms numbers, etc). These Notification objects are available as related resources on job types (jobs, inventory updates, project updates), and also at /api/v2/notifications. You may also see what notifications have been sent from a notifier by examining its related resources.

If a notification fails, it will not impact the job associated to it or cause it to fail. The status of the notification can be viewed at its detail endpoint (/api/v2/notifications/<n>).

21.3 Create a Notification Template

To create a Notification Template:

1. Click the Notifications () icon from the left navigation bar.

NOTIFICATIONS			•
NOTIFICATION TEMPLATES			
	PLI	EASE ADD ITEMS TO THIS LIST	
	putton.		
SETTINGS / NOTIFICATIONS / CREATE NOTIFICATION	TEMPLATE		0
NEW NOTIFICATION TEMPLATE *NAME	DESCRIPTION	*ORGANI Q	ZATION
*TYPE Choose a type	•		
			CANCEL SAVE

- 3. Enter the name of the notification, a description, and the organization it belongs to in their respective fields.
- 4. Choose a type of notification from the **Type** drop-down menu. Refer to the subsequent sections for additional information.
- 5. Once all required information is complete, click **Save** to add the notification.

21.4 Notification Types

Notification types supported with Ansible Tower 3.4.4:

• Email		
• Slack		
• Twilio		
PagerDuty		
• HipChat		
• Webhook		
• Mattermost		
• Rocket.Chat		
• IRC		

Each of these have their own configuration and behavioral semantics and testing them may need to be approached in different ways. The following sections will give as much detail as possible.

21.4.1 Email

The email notification type supports a wide variety of SMTP servers and has support for TLS/SSL connections.

You must provide the following details to setup an email notification: - Username - Host - Sender email - Recipient list - Password - Port

NEW NOTIFICATION TEMPLATE			Θ
*NAME Tell Me	DESCRIPTION notification test	*ORGANIZATION	
*TYPE Email v			
TYPE DETAILS			
* USERNAME	*HOST	*SENDER EMAIL	
luckydog	honeydog.com	info@honeydog.com	
*RECIPIENT LIST @ engineering@honeydog.com	*PASSWORD SHOW	*PORT	
release-managers@honeydog.com			
OPTIONS Use TLS Use SSL			
		CANCEL	SAVE

Caution: TLS and SSL connections are mutually exclusive and should not be selected at the same time. Be sure to only select one-checking both causes the notification to silently fail.

21.4.2 Slack

Slack, a collaborative team communication and messaging tool, is pretty easy to configure.

You must supply the following to setup Slack notifications:

- A token (which you can obtain from creating a bot in the integrations settings for the Slack team at https: //api.slack.com/bot-users)
- Destination channel(s)

You must also invite the notification bot to join the channel(s) in question. Note that private messages are not supported.

NEW NOTIFICATION TEMPLATE			0
*NAME	DESCRIPTION	* ORGANIZATION	
Tell Me	notification test	Q Honey Dog, Inc.	
*TYPE			
Slack 💌			
TYPE DETAILS			
* DESTINATION CHANNELS 🔞	*TOKEN		
#engineering #helpdesk #support	SHOW		
		CANCEL	SAVE

21.4.3 Twilio

Twilio service is an Voice and SMS automation service. Once you are signed in, you must create a phone number from which the message will be sent. You can then define a "Messaging Service" under Programmable SMS and associate the number you created before with it.

Note that you may need to verify this number or some other information before you are allowed to use it to send to any numbers. The Messaging Service does not need a status callback URL nor does it need the ability to Process inbound messages.

Under your individual (or sub) account settings, you will have API credentials. Twilio uses two credentials to determine which account an API request is coming from. The "Account SID", which acts as a username, and the "Auth Token" which acts as a password.

To setup Twilio, provide the following details:

- Account Token
- Source phone number (this is the number associated with the messaging service above and must be given in the form of "+15556667777")
- Destination phone number (this will be the list of numbers to receive the SMS and should be the 10-digit phone number)
- Account SID

DESCRIPTION	* ORGANIZATION
notification test	Q Honey Dog, Inc.
v	
* SOURCE PHONE NUMBER @	* DESTINATION SMS NUMBER @
9109876555	9109676565
	notification test SOURCE PHONE NUMBER

21.4.4 PagerDuty

PagerDuty is a fairly straightforward integration. The user must first create an API Key in the pagerduty system (this is the token that is given to Tower) and then create a "Service" which provides an "Integration Key" that will also be given to Tower. The other options of note are:

- API Token: The user must first create an API Key in the PagerDuty system (this is the token that is given to Tower.
- PagerDuty Subdomain: When you sign up for the PagerDuty account, you receive a unique subdomain to communicate with. For instance, if you signed up as "towertest", the web dashboard will be at towertest. pagerduty.com and you will give the Tower API towertest as the subdomain (not the full domain).
- API Service/Integration Key
- Client Identifier: This will be sent along with the alert content to the pagerduty service to help identify the service that is using the api key/service. This is helpful if multiple integrations are using the same API key and service.

NEW NOTIFICATION TEMPLATE			Θ
*NAME	DESCRIPTION	* ORGANIZATION	
Tell Me	notification test	Q Honey Dog, Inc.	
*TYPE			
Pagerduty 👻			
TYPE DETAILS			
* API TOKEN	* PAGERDUTY SUBDOMAIN	* API SERVICE/INTEGRATION KEY	
SHOW			
* CLIENT IDENTIFIER			
		CANCEL	

21.4.5 HipChat

There are several ways to integrate with HipChat. The Tower implementation uses HipChat "Integrations". Currently you can find this at the bottom right of the main HipChat webview. From there, you will select "Build your own Integration". After creating that, it will list the auth_token that needs to be supplied to Tower. Some other relevant details on the fields accepted by Tower for the HipChat notification type:

- Destination Channels: Channels which should receive the notification ("engineering" or "#support", for example).
- Token: The token listed after building your own HipChat integration.
- Label to be shown with notification: Along with the integration name itself this will put another label on the notification (which could be helpful if multiple services are using the same integration to distinguish them from each other).
- API URL: The URL of the Hipchat API service. If you create a team hosted by them it will be something like: https://team.hipchat.com. For a self-hosted integration, use a base URL similar to https://hipchat.yourcompany.com/ and add in appropriate Destination Channels without the # leading them ("engineering" rahter than "#engineering").
- Notification Color: This will highlight the message as the given color. If set to something HipChat does not expect, then the notification will generate an error in the given color.
- Notify Channel: Selecting this will cause the bot to "notify" channel members. Normally it will just be stuck as a message in the chat channel without triggering anyone's notifications. This option will notify users of the channel respecting their existing notification settings (browser notification, email fallback, etc.).

NEW NOTIFICATION TEMPLATE			8
*NAME Tell Me	DESCRIPTION notification test	*ORGANIZATION Q Honey Dog, Inc.	
*TYPE HipChat			
TYPE DETAILS * DESTINATION CHANNELS @ #engineering #support	*TOKEN SHOW ·····	* LABEL TO BE SHOWN WITH NOTIFICATION hipchat-notifier-tell-me	
*API URL https://mycompany.hipchat.com	* NOTIFICATION COLOR @ red	O NOTIFY CHANNEL	
		CANCEL	

21.4.6 Webhook

The webhook notification type in Ansible Tower provides a simple interface to sending POSTs to a predefined web service. Tower will POST to this address using application/json content type with the data payload containing all relevant details in json format.

The parameters are pretty straightforward:

- Target URL: The full URL that will be POSTed to
- HTTP Headers: Headers in JSON form where the keys and values are strings. For example:

{"Authentication":	"988881adc9fc3655077dc2	2d4d757d480b5ea0e11", "Messa	ageType":
NEW NOTIFICATION TEMPLATE			8
*NAME	DESCRIPTION	* ORGANIZATION	
Tell Me	notification test	Q Honey Dog, Inc.	
*TYPE			
Webhook	•		
TYPE DETAILS *TARGET URL			
http://www.honeydog.com/web/db/notificat	on		
1 {"Authentication": "988881adc9	fc3655077dc2d4d757d480b5ea0ell", "Messa	geType": "Test"}	
			CANCEL

21.4.7 Mattermost

The Mattermost notification type in Ansible Tower provides a simple interface to Mattermost's messaging and collaboration workspace. The parameters that can be specified are:

- Target URL (required): The full URL that will be POSTed to
- Username
- Channel
- Icon URL: specifies the icon to display for this notifier
- Disable SSL Verification: Turns off Tower's attempt to verify the authenticity of the target's certificate. Environments that use internal or private CA's should select this option to disable verification.

NEW NOTIFICATION TEMPLATE			Θ
* NAME Tell me	DESCRIPTION notification test	* ORGANIZATION Q Honey Dog, Inc.	J
* TYPE Mattermost			
TYPE DETAILS * TARGET URL http://1.2.3.4:8065/hooks/j5kurmybl513b4pnf9sdpl	USERNAME	CHANNEL my-channel	
ICON URL https://www.myicon/favicon.ico	DISABLE SSL VERIFICATION		
		CANCEL	SAVE

21.4.8 Rocket.Chat

The Rocket.Chat notification type in Ansible Tower provides an interface to Rocket.Chat's collaboration and communication platform. The parameters that can be specified are:

- Target URL (required): The full URL that will be POSTed to
- Username:
- Icon URL: specifies the icon to display for this notifier
- Disable SSL Verification: Turns off Tower's attempt to verify the authenticity of the target's certificate. Environments that use internal or private CA's should select this option to disable verification.

NEW NOTIFICATION TEMPLATE			0
* NAME	DESCRIPTION	* ORGANIZATION	
Tell me	notification test	Q Honey Dog, Inc.	
* TYPE			
Rocket.Chat 🔹			
TYPE DETAILS			
* TARGET URL	USERNAME	ICON URL	
http://1.2.3.4:8065/hooks/dj4ruwjew847w84q93308	jerry	https://www.myicon/favicon.ico	
DISABLE SSL VERIFICATION			
		CANCEL	SAVE

21.4.9 IRC

The Tower IRC notification takes the form of an IRC bot that will connect, deliver its messages to channel(s) or individual user(s), and then disconnect. The Tower notification bot also supports SSL authentication. The Tower bot does not currently support Nickserv identification. If a channel or user does not exist or is not on-line then the Notification will not fail; the failure scenario is reserved specifically for connectivity.

Connectivity information is straightforward:

- IRC Server Password: IRC servers can require a password to connect. If the server does not require one, leave blank
- IRC Server Port: The IRC server Port
- IRC Server Address: The host name or address of the IRC server
- IRC Nick: The bot's nickname once it connects to the server
- Destination Channels or Users: A list of users and/or channels to which to send the notification.
- SSL Connection: Should the bot use SSL when connecting

NEW NOTIFICATION TEMPLATE			8
*NAME	DESCRIPTION	*ORGANIZATION	
Tell Me	notification test	Q Honey Dog, Inc.	
*TYPE			
IRC			
TYPE DETAILS			
* IRC SERVER PASSWORD	* IRC SERVER PORT	* IRC SERVER ADDRESS	
SHOW	6667	Ç irc.testirc.net	
*IRC NICK	*DESTINATION CHANNELS OR USERS 🔞		
helpbot	#engineering #release-engineers	SSL CONNECTION	
		CANCEL	SAVE

21.5 Configuring the towerhost hostname

In /etc/tower/settings.py, you can modify TOWER_URL_BASE='https://tower.example.com' to change the notification hostname, replacing https://tower.example.com with your preferred hostname. You must restart Tower services after saving your changes with ansible-tower-service restart.

Refreshing your Tower license also changes the notification hostname. New installations of Ansible Tower 3.0 should not have to set the hostname for notifications.

21.5.1 Resetting the TOWER_URL_BASE

The primary way that Tower determines how the base URL (TOWER_URL_BASE) is defined is by looking at an incoming request and setting the server address based on that incoming request.

Tower takes settings values from the database first. If no settings values are found, Tower falls back to using the values from the settings files. If a user posts a license by navigating to the Tower host's IP adddress, the posted license is written to the settings entry in the database.

To change the TOWER_URL_BASE if the wrong address has been picked up, navigate to the license from the Tower



Settings (Menu's License tab using the DNS entry you wish to appear in notifications, and re-add your license.

CHAPTER TWENTYTWO

SETTING UP AN INSIGHTS PROJECT

Tower supports integration with Red Hat Insights. Once a host is registered with Insights, it will be continually scanned for vulnerabilities and known configuration conflicts. Each of the found problems may have an associated fix in the form of an Ansible playbook. Insights users create a maintenance plan to group the fixes and, ultimately, create a playbook to mitigate the problems. Tower tracks the maintenance plan playbooks via an Insights project in Tower. Authentication to Insights via Basic Auth, from Tower, is backed by a special Insights Credential, which must first be established in Tower. To ultimately run an Insights Maintenance Plan in Tower, you need an Insights project, an inventory, and a Scan Job template.

22.1 Create Insights Credential

To create a new credential for use with Insights:



1. Click the Credentials () icon from the left navigation bar to access the Credentials page.



- 2. Click the **button** located in the upper right corner of the Credentials screen.
- 3. Enter the name of the credential to be used in the Name field.
- 4. Optionally enter a description for this credential in the **Description** field.
- 5. In the **Organization** field, optionally enter the name of the organization with which the credential is associated,

or click the \bigcirc button and select it from the pop-up window.

6. In the **Credential Type** field, enter **Insights** or click the ^Q button and select it from the credential type pop-up window.

SELECT CREDENTIAL TYPE	8
SEARCH	QKEY
NAME [▲]	
 Amazon Web Services 	
 Google Compute Engine 	
Insights	
 Machine 	
O Microsoft Azure Classic (deprecated)	
< 1 2 3 > PAGE 1 OF 3	ITEMS 1 - 5 OF 13
	CANCEL

7. Enter a valid Insights credential in the **Username** and **Password** fields. The Insights credential is the user's Red Hat Customer Portal account username and password.

SETTINGS / CREDENTIALS / CREATE CREDENTIAL		0
NEW CREDENTIAL DETAILS PERMISSIONS		0
NAME Insights Credential CREDENTIAL TYPE Q Insights TYPE DETAILS		ORGANIZATION @ Q Default
* USERNAME	* PASSWORD	
mycreds@redhat.com	SHOW	CANCEL

8. Click Save when done.

22.2 Create an Insights Project

To create a new Insights project:



1. Click the Projects icon from the left navigation bar to access the Projects page.



button located in the upper right corner of the Projects screen. 2. Click the

- 3. Enter the appropriate details into the required fields, at minimum. Note the following fields requiring specific Insights-related entries:
- Name: Enter the name for your Insights project.
- Organization: Enter the name of the organization associated with this project, or click the button and select it from the pop-up window.
- SCM Type: Select Red Hat Insights.
- Upon selecting the SCM type, the Source Details field expands.
- 4. The Credential field is pre-populated with the Insights credential you previously created. If not, enter the credential, or click the ^{LL} button and select it from the pop-up window.

5. Click to select the update option(s) for this project from the **Options** field, and provide any additional values, if

applicable. For information about each option, click the Help 🖤 button next to the options.

NEW PROJECT				
DETAILS PERMISSIONS NOTI	FICATIONS JOB TEMPLATES			
* NAME		DESCRIPTION	* ORGANIZATION	
Insights Project			Q Default	
SCM TYPE				
Red Hat Insights	Ŧ			
SOURCE DETAILS				
* CREDENTIAL		SCM UPDATE OPTIONS		
Q Insights Credentials		Clean @		
		Delete on Update @ Update on Launch @		

6. Click Save when done.

All SCM/Project syncs occur automatically the first time you save a new project. However, if you want them to be

updated to what is current in Insights, manually update the SCM-based project by clicking the ^{see} button under the project's available Actions.

This process syncs your Tower Insights project with your Insights account solution. Notice that the status dot beside the name of the project updates once the sync has run.

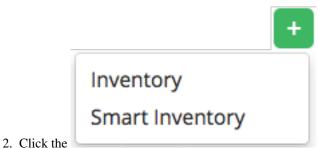
SEARCH		Q KEY				1	+ ADD
NAME 🔺	TYPE 🗢	REVISION 🗘	LAST UPDATED			AC	TIONS
O Demo Project	Git			6	Ê	(MAR)	Û
Insights Project	Red Hat Insights	644£3-5 🗋	10/9/2017 11:16:19 PM	۵	Ê		Ê

22.3 Create Insights Inventory

The Insights playbook contains a hosts: line where the value is the hostname that Insights itself knows about, which may be different than the hostname that Tower knows about. Therefore, make sure that the hostnames in the Tower inventory match up with the system in the Red Hat Insights Portal.

To create a new inventory for use with Insights:

) icon from the left navigation bar to access the Inventories page. 1. Click the Inventories (



button and select Inventory from the drop-down menu

list to launch a New Inventory window.

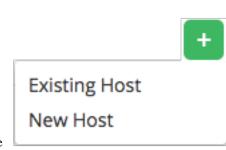
- 3. Enter the name and organization to be used in their respective fields.
- 4. In the Insights Credential field, enter the name of the Insights credential you previously created, or click the Q

button and select it from the pop-up window.

INVENTORIES / Insights Inventory				0
Insights Inventory DETAILS PERMISSIONS GROUPS HOSTS SOURCES	COMPLETED JOBS			8
* NAME	DESCRIPTION	* OR	GANIZATION	
Insights Inventory		Q	Default	
INSIGHTS CREDENTIAL Q Insights Credential VARIABLES @ YAML	INSTANCE GROUPS @			
1				
			CANCEL	AVE

5. Click Save and proceed to add a host.

Note: Typically, your inventory already contains Insights hosts. Tower just doesn't know about them yet. The Insights credential allows Tower to get information from Insights about an Insights host. Tower identifying a host as an Insights host can occur without an Insights credential with the help of scan_facts.yml file. For instructions, refer to the *Create a Scan Job Template* section.



button to open the Create Host

- 6. Click the **Hosts** tab and click the dialog.
- 7. Enter the name in the Host Name field associated with the Insights host that will be used.
- 8. Click **Save** when done.

22.4 Create a Scan Project

In order for Tower to utilize Insights Maintenance Plans, it must have visibility to them. Create and run a scan job against the inventory using a stock manual scan playbook.



1. Click the Projects () icon from the left navigation bar to access the Projects page.



2. Click the **button** located in the upper right corner of the Projects screen.

- 3. Enter the appropriate details into the required fields, at minimum. Note the following fields requiring specific Insights-related entries:
- Name: Enter the name for your scan project.
- **Organization**: The name of the organization is pre-populated with the organization you chose from creating the inventory.
- SCM Type: Select Git.
- Upon selecting the SCM type, the Source Details field expands.
- 4. In the SCM URL field, enter https://github.com/ansible/awx-facts-playbooks. This is the location where the scan job template is stored.
- 5. Click to select the update option(s) for this project from the Options field, and provide any additional values, if

applicable. For information about each option, click the Help 🖤 button next to the options.

PROJECTS / CREATE PROJECT			۵
NEW PROJECT DETAILS PERMISSIONS NOTIFICATIONS JOB TEMPLATES			0
* NAME	DESCRIPTION	* ORGANIZATION	
Scan Project		Q Default	
* SCM TYPE			
Git *			
SOURCE DETAILS			
* SCM URL @	SCM BRANCH/TAG/COMMIT	SCM CREDENTIAL	
https://github.com/ansible/awx-facts-playbooks		٩	
SCM UPDATE OPTIONS Clean @ Delete on Update @ Update on Launch @			
		CANCEL	SAVE

6. Click **Save** when done.

All SCM/Project syncs occur automatically the first time you save a new project. However, if you want them to be

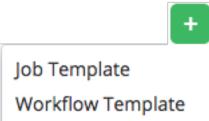
updated to what is current in Insights, manually update the SCM-based project by clicking the th button under the project's available Actions.

Syncing imports into Tower any Maintenance Plans in your Insights account that has a playbook solution. It will use the default Plan resolution. Notice that the status dot beside the name of the project updates once the sync has run.

22.5 Create a Scan Job Template

Create a scan job template that uses the fact scan playbook:





button and select Job Template from the drop-down menu

list to launch a New Job Template window.

2. Click the

- 3. Enter the appropriate details into the required fields, at minimum. Note the following fields requiring specific Insights-related entries:
- Name: Enter the name of your scan job.
- Job Type: Choose Run from the drop-down menu list.
- Inventory: Enter the name of the Insights inventory, or click the subtract button and select it from the pop-up window.

- **Project**: Enter the name of the Scan project you previously created, or click the button and select it from the pop-up window.
- **Playbook**: Select scan_facts.yml from the drop-down menu list. This is the playbook associated with the Scan project you previously set up.
- **Credential**: Enter the credential to use for this project or click the window. The credential does not have to be an Insights credential.
- Verbosity: Keep the default setting, or select the desired verbosity from the drop-down menu list.
- 4. Click to select Enable Privilege Escalation and Enable Fact Cache from the Options field.

A scan job template for Insights should be launched with the Privilege Escalation option enabled to allow the job to access /etc/redhat-access-insights/machine-id as a root user in order to obtain the value of system_id from the target host. What this does is activate the Insights button from the Host, which is needed to *remediate the Insights inventory*. Otherwise, the system_id parameter in the result of your scan job is set to null and the Insights button will not appear.

EMPLATES / Insights Scan				0
Insights Scan				0
	IFICATIONS COMPLETED JOBS	ADD SURVEY DESCRIPTION	* JOB TYPE @ Run	PROMPT ON LAUNCH
* INVENTORY @ Q Insights Inventory	PROMPT ON LAUNCH	PROJECT C Scan Project	* PLAYBOOK @ scan_facts.yml	¥
	PROMPT ON LAUNCH	FORKS @		PROMPT ON LAUNCH
* VERBOSITY @ 0 (Normal)	PROMPT ON LAUNCH	INSTANCE GROUPS @	JOB TAGS @	PROMPT ON LAUNCH
SKIP TAGS 🔞	PROMPT ON LAUNCH		SHOW CHANGES @	PROMPT ON LAUNCH
OPTIONS Cable Privilege Escalation Allow Provisioning Callbacks Enable Concurrent Jobs Use Fact Cache EXTRA VARIABLES YAML JSON				PROMPT ON LAUNCH
1				
				CANCEL

5. Click Save when done.

6. Click the icon to launch the scan job template.

Once complete, the job results display in the Job Details page.

DETAILS		1	l lr	nsights Sc	an PLAYS 1 TASKS 8 HOSTS 1 ELAF	PSED 00:00:07
STATUS	Successful		Ш.,			
STARTED	10/6/2017 12:14:22 PM					
INISHED	10/6/2017 12:14:30 PM			SEARCH		Q KEY
EMPLATE	Insights Scan					
OB TYPE	Run			+ =	TASK [Scan puckages (#Linuows)]	16.14.67
AUNCHED BY	admin			20	skipping: [localhost]	
NVENTORY	Insights Inventory			21		
ROIECT	 Scan Project 			 22 23 	TASK [Scan services (Windows)] ************************************	12:14:29
REVISION	77cbb77			25	skipping: [localnost]	
LAYBOOK	_			₹ 25	TASK [Scan files (Windows)] ************************************	12:14:30
	scan_facts.yml			26	skipping: [localhost]	
	Demo Credential			27		
				28		12:14:30
ORKS	0				localhost : ok=4 changed=0 unreachable=0 failed=0	
MACHINE CREDENTIAL FORKS	Demo Credential				PLAY RECAP ************************************	12
/ERBOSITY	0 (Normal)			30		

22.6 Remediate Insights Inventory

Remediation of an Insights inventory allows Tower to run Insights playbooks with a single click.

Ъ.

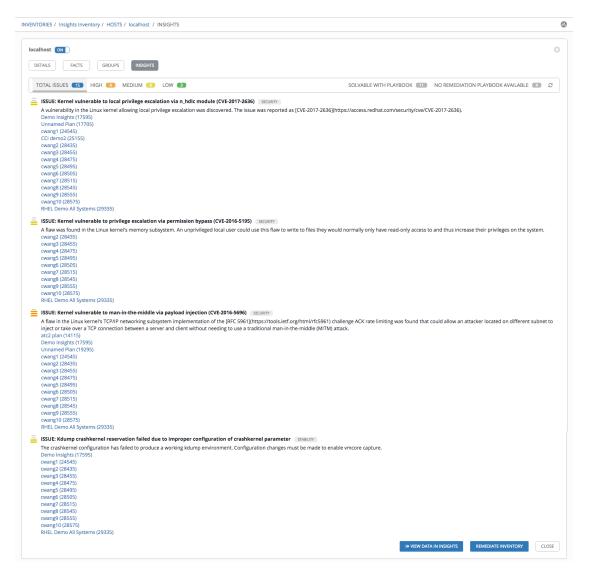
- 1. Click the Inventories (LLLLL) icon from the left navigation bar to access the Inventories page.
- 2. In the list of inventories, click to open the details of your Insights inventory.
- 3. Click the **Hosts** tab to access the Insights hosts that have been loaded from the scan process.
- 4. Click to open the host that was loaded from Insights.

Notice the Insights tab is now shown on Hosts page. This indicates that Insights and Tower have reconciled the inventories and is now set up for one-click Insights playbook runs.

INVENTORIES / Insights Inventory / HOSTS / localhost		0
Iocalhost ON DETAILS FACTS GROUPS INSIGHTS		٢
* HOST NAME @ localhost VARIABLES @ YAML JSON	DESCRIPTION	
1 2 ansible_connection: local		
		ANCEL

5. Click Insights.

The screen below populates with a list of issues and whether or not the issues can be resolved with a playbook is shown.



6. Scroll down to the bottom of the Insights inventory page, and click the **Remediate Inventory** button to update hosts in the inventory.

sights Inventory		
DETAILS PERMISSIONS GROUPS	HOSTS SOURCES COMPLETED JOBS REMEDIATE INVENTORY	
SEARCH	Q KEY	RUN COMMANDS + ADD HOS
HOSTS 🔺	RELATED GROUPS	ACTION
🗆 💽 🕒 localhost		ø

Upon remediation, the New Job Template window opens. Notice the Inventory and Project fields are pre-populated.

NEW JOB TEMPLATE				
DETAILS PERMISSIONS NOTIFIC	CATIONS COMPLETED JOBS	ADD SURVEY		
NAME		DESCRIPTION	* JOB TYPE 😡	PROMPT ON LAUNCH
			Run	*
NVENTORY @	PROMPT ON LAUNCH	* PROJECT 😨	* PLAYBOOK @	
Q Insights Inventory		Q Insights Project	Choose a playbook	•
CREDENTIAL 🕖	PROMPT ON LAUNCH	FORKS @	LIMIT 😡	PROMPT ON LAUNCH
Q		DEFAULT	\$	
VERBOSITY @	PROMPT ON LAUNCH	INSTANCE GROUPS 🔞	JOB TAGS @	PROMPT ON LAUNCH
0 (Normal)	*	Q		
SKIP TAGS 😡	PROMPT ON LAUNCH	LABELS 🚱	SHOW CHANGES 🚱	PROMPT ON LAUNCH
			OFF	

Use this new job template to create a job template that pulls Maintenance Plans from Insights.

- 7. Enter the appropriate details into the required fields, at minimum. Note the following fields requiring specific Insights-related entries:
- Name: Enter the name of your Maintenance Plan.
- Job Type: If not already populated, select Run from the drop-down menu list.
- Inventory: This field is pre-populated with the Insights inventory you previously created.
- Project: This field is pre-populated with the Insights project you previously created.
- **Playbook**: Select a playbook associated with the Maintenance Plan you want to run from the drop-down menu list.
- **Credential**: Enter the credential to use for this project or click the window. The credential does not have to be an Insights credential.
- Verbosity: Keep the default setting, or select the desired verbosity from the drop-down menu list.

IPLATES / CREATE JOB TEMPLATE						
NEW JOB TEMPLATE						e
DETAILS PERMISSIONS NO	TIFICATIONS COMPLETED JOBS	ADD SURVEY				
NAME		DESCRIPTION		* JOB TYPE 🚱	PROMPT ON LAUNCH	
Maintenance Plan Job				Run	v	
INVENTORY @	PROMPT ON LAUNCH	* PROJECT @		* PLAYBOOK @		
Q Insights Inventory		Q Insights Project		ansiblefest_demo-33675.yml	v	
REDENTIAL 😡	PROMPT ON LAUNCH	FORKS 🚱		LIMIT @	PROMPT ON LAUNCH	
Q × 4 DEMO CREDENTIAL		DEFAULT	Ĵ			
VERBOSITY 🚱	PROMPT ON LAUNCH	INSTANCE GROUPS @		JOB TAGS 🕖	PROMPT ON LAUNCH	
0 (Normal)	▼	Q				
KIP TAGS 🔞	PROMPT ON LAUNCH	LABELS @		SHOW CHANGES @	PROMPT ON LAUNCH	
				OFF		
DPTIONS Denable Privilege Escalation Denable Privilege Escalation Denable Privisioning Calibacks Denable Concurrent Jobs Duse Fact Cache Denable Privilege Privil						
XTRA VARIABLES @ YAML JSON					PROMPT ON L	LAUN
1						

- 8. Click Save when done.
- 9. Click the *icon to launch the job template.*

Once complete, the job results display in the Job Details page.

CHAPTER TWENTYTHREE

BEST PRACTICES

23.1 Use Source Control

While Tower supports playbooks stored directly on the Tower server, best practice is to store your playbooks, roles, and any associated details in source control. This way you have an audit trail describing when and why you changed the rules that are automating your infrastructure. Plus, it allows for easy sharing of playbooks with other parts of your infrastructure or team.

23.2 Ansible file and directory structure

Please review the Ansible best practices from the Ansible documentation at http://docs.ansible.com/playbooks_best_ practices.html. If creating a common set of roles to use across projects, these should be accessed via source control submodules, or a common location such as /opt. Projects should not expect to import roles or content from other projects.

Note: Playbooks should not use the vars_prompt feature, as Tower does not interactively allow for vars_prompt questions. If you must use vars_prompt, refer to and make use of the *Surveys* functionality of Tower.

Note: Playbooks should not use the pause feature of Ansible without a timeout, as Tower does not allow for interactively cancelling a pause. If you must use pause, ensure that you set a timeout.

Jobs run in Tower use the playbook directory as the current working directory, although jobs should be coded to use the playbook_dir variable rather than relying on this.

23.3 Use Dynamic Inventory Sources

If you have an external source of truth for your infrastructure, whether it is a cloud provider or a local CMDB, it is best to define an inventory sync process and use Tower's support for dynamic inventory (including cloud inventory sources and custom inventory scripts). This ensures your inventory is always up to date.

Note: With the release of Ansible Tower 2.4.0, edits and additions to Inventory host variables now persist beyond an inventory sync as long as --overwrite_vars is **not** set. To have inventory syncs behave as they did before, it is now required that both --overwrite and --overwrite_vars are set.

23.4 Variable Management for Inventory

Keeping variable data along with the objects in Tower (see the inventory editor) is encouraged, rather than using group_vars/ and host_vars/. If you use dynamic inventory sources, Tower can sync such variables with the database as long as the **Overwrite Variables** option is not set.

23.5 Autoscaling

Using the "callback" feature to allow newly booting instances to request configuration is very useful for auto-scaling scenarios or provisioning integration.

23.6 Larger Host Counts

Consider setting "forks" on a job template to larger values to increase parallelism of execution runs. For more information on tuning Ansible, see the Ansible blog.

23.7 Continuous integration / Continuous Deployment

For a Continuous Integration system, such as Jenkins, to spawn an Tower job, it should make a curl request to a job template, or use the Tower CLI tool. The credentials to the job template should not require prompting for any particular passwords. Using the API to spawn jobs is covered in the Tower API guide.

CHAPTER TWENTYFOUR

SECURITY

The following sections will help you gain an understanding of how Ansible Tower handles and lets you control file system security.

All playbooks are executed via the awx file system user. For running jobs, Ansible Tower defaults to offering job isolation via Linux namespacing and chroots. This projection ensures jobs can only access playbooks and roles from the Project directory for that job template and common locations such as /opt. Playbooks are not able to access roles, playbooks, or data from other Projects by default.

If you need to disable this protection (not recommended), you can edit /etc/tower/settings.py and set AWX_PROOT_ENABLED to False.

Note: In this scenario, playbooks have access to the file system and all that implies; therefore, users who have access to edit playbooks **must** be trusted.

For credential security, users may choose to upload locked SSH keys and set the unlock password to "ask". You can also choose to have the system prompt them for SSH credentials or sudo passwords rather than having the system store them in the database.

24.1 Playbook Access and Information Sharing

By default, Tower's multi-tenant security prevents playbooks from reading files outside of their project directory. In older version of Ansible Tower a system called proot was used to isolate tower job processes from the rest of the system. For Tower version 3.1 and later, bubblewrap is used instead, due to its light weight and maintained process isolation system.

By default bubblewrap is enabled, but can be turned off via the Configure Tower screen in the Tower User Interface or from the tower settings file.



To access the Configure Tower screen, refer to the Tower Configuration section. To customize your bubblewrap settings through the settings file, navigate to the /etc/tower/settings.py file.

Process isolation, when enabled, will be used for the following Job types:

- Job Templates Launching jobs from regular job templates
- · Ad-hoc Commands Launching ad-hoc commands against one or more hosts in an inventory

By default, process isolation hides the following directories from the above tasks:

- /etc/tower to prevent exposing Tower configuration
- /var/lib/awx with the exception of the current project being used (for regular job templates)
- /var/log
- /tmp (or whatever the system temp directory is) with the exception of the processes' own temp files.

You can customize what to hide or expose when running playbooks, using the Configure Tower screen or the settings file. Refer the next section, *Bubblewrap functionality and variables* for more information.

24.1.1 Bubblewrap functionality and variables

The bubblewrap functionality in Ansible Tower limits which directories on the Tower file system are available for playbooks to see and use during playbook runs. You may find that you need to customize your bubblewrap settings in some cases. To fine tune your usage of bubblewrap, there are certain variables that can be set.

To disable or enable bubblewrap support for running jobs (playbook runs only), ensure you are logged in as the Admin user:



- 1. Click the Settings () icon from the left navigation bar.
- 2. Click the "Jobs" tab.
- 3. Scroll down until you see "Enable Job Isolation" and change the toggle button selection to **OFF** to disable bubblewrap support or select **ON** to enable it.

By default, the Tower will use the system's tmp directory (/tmp by default) as its staging area. This can be changed in the **Job Isolation Execution Path** field of the Configure tower screen, or by updating the following entry in the settings file:

```
AWX_PROOT_BASE_PATH = "/opt/tmp"
```

If there is other information on the system that is sensitive and should be hidden, you can specify those in the Configure Tower screen in the **Paths to Hide to Isolated Jobs** or by updating the following entry in the settings file:

AWX_PROOT_HIDE_PATHS = ['/list/of/', '/paths']

If there are any directories that should specifically be exposed, you can specify those in the Configure Tower screen in the **Paths to Expose to Isolated Jobs** or by updating the following entry in the settings file:

AWX_PROOT_SHOW_PATHS = ['/list/of/', '/paths']

Note: The primary file you may want to add to AWX_PROOT_SHOW_PATHS is /var/lib/awx/. ssh, if your playbooks need to use keys or settings defined there.

If you made changes in the settings file, be sure to restart services with the ansible-tower-service restart command after your changes have been saved.

24.2 Role-Based Access Controls

Role-Based Access Controls (RBAC) are built into Tower and allow Tower administrators to delegate access to server inventories, organizations, and more. Administrators can also centralize the management of various credentials, allowing end users to leverage a needed secret without ever exposing that secret to the end user. RBAC controls allow Tower to help you increase security and streamline management.

RBACs are easiest to think of in terms of Roles which define precisely who or what can see, change, or delete an "object" for which a specific capability is being set. In releases prior to Ansible Tower version 3.0, RBAC was thought of in terms of granting permissions to users or teams. Starting with Tower 3.0, RBAC is best thought of as granting roles to users or teams, which is a more intuitive approach.

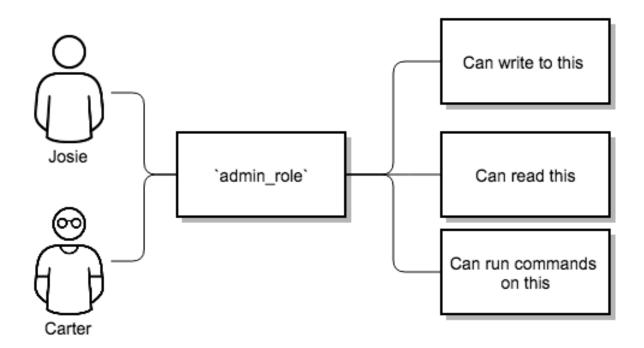
There are a few main concepts that you should become familiar with regarding Tower's RBAC design–roles, resources, and users. Users can be members of a role, which gives them certain access to any resources associated with that role, or any resources associated with "descendant" roles.

A role is essentially a collection of capabilities. Users are granted access to these capabilities and Tower's resources through the roles to which they are assigned or through roles inherited through the role hierarchy.

Roles associate a group of capabilities with a group of users. All capabilities are derived from membership within a role. Users receive capabilities only through the roles to which they are assigned or through roles they inherit through the role hierarchy. All members of a role have all capabilities granted to that role. Within an organization, roles are relatively stable, while users and capabilities are both numerous and may change rapidly. Users can have many roles.

24.2.1 Role Hierarchy and Access Inheritance

Imagine that you have an organization named "SomeCompany" and want to allow two people, "Josie" and "Carter", access to manage all the settings associated with that organization. You should made both people members of the organization's admin_role.

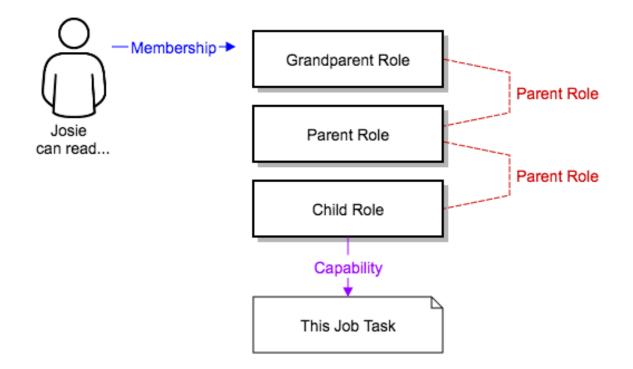


Often, you will have many Roles in a system and you will want some roles to include all of the capabilities of other roles. For example, you may want a System Administrator to have access to everything that an Organization Administrator has access to, who has everything that a Project Administrator has access to, and so on.

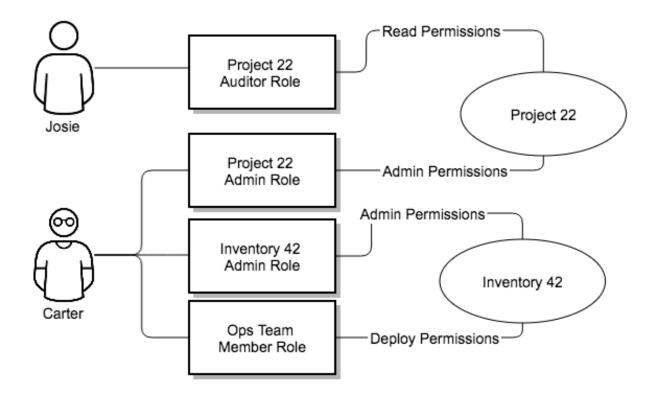
This concept is referred to as the 'Role Hierarchy':

- Parent roles get all capabilities bestowed on any child roles
- Members of roles automatically get all capabilities for the role they are a member of, as well as any child roles.

The Role Hierarchy is represented by allowing Roles to have "Parent Roles". Any capability that a Role has is implicitly granted to any parent roles (or parents of those parents, and so on).



Often, you will have many Roles in a system and you will want some roles to include all of the capabilities of other roles. For example, you may want a System Administrator to have access to everything that an Organization Administrator has access to, who has everything that a Project Administrator has access to, and so on. We refer to this concept as the 'Role Hierarchy' and it is represented by allowing Roles to have "Parent Roles". Any capability that a Role has is implicitly granted to any parent roles (or parents of those parents, and so on). Of course Roles can have more than one parent, and capabilities are implicitly granted to all parents.



RBAC controls also give you the capability to explicitly permit User and Teams of Users to run playbooks against certain sets of hosts. Users and teams are restricted to just the sets of playbooks and hosts to which they are granted capabilities. And, with Tower, you can create or import as many Users and Teams as you require–create users and teams manually or import them from LDAP or Active Directory.

RBACs are easiest to think of in terms of who or what can see, change, or delete an "object" for which a specific capability is being determined.

24.2.2 Applying RBAC

The following sections cover how to apply Tower's RBAC system in your environment.

Editing Users

When editing a user, a Tower system administrator may specify the user as being either a *System Administrator* (also referred to as the Superuser) or a *System Auditor*.

- System administrators implicitly inherit all capabilities for all objects (read/write/execute) within the Tower environment.
- System Auditors implicitly inherit the read-only capability for all objects within the Tower environment.

Editing Organizations

When editing an organization, system administrators may specify the following roles:

- · One or more users as organization administrators
- · One or more users as organization auditors
- And one or more users (or teams) as organization members

Users/teams that are members of an organization can view their organization administrator.

Users who are organization administrators implicitly inherit all capabilities for all objects within that Tower organization.

Users who are organization auditors implicitly inherit the read-only capability for all objects within that Tower organization.

Editing Projects in an Organization

When editing a project in an organization for which they are the administrator, system administrators and organization administrators may specify:

- One or more users/teams that are project administrators
- One or more users/teams that are project members
- And one or more users/teams that may update the project from SCM, from among the users/teams that are members of that organization.

Users who are members of a project can view their project administrators.

Project administrators implicitly inherit the capability to update the project from SCM.

Administrators can also specify one or more users/teams (from those that are members of that project) that can use that project in a job template.

Creating Inventories and Credentials within an Organization

All access that is granted to use, read, or write credentials is now handled through roles. You no longer set the "team" or "user" for a credential. Instead, you use Tower's RBAC system to grant ownership, auditor, or usage roles.

System administrators and organization administrators may create inventories and credentials within organizations under their administrative capabilities.

Whether editing an inventory or a credential, System administrators and organization administrators may specify one or more users/teams (from those that are members of that organization) to be granted the usage capability for that inventory or credential.

System administrators and organization administrators may specify one or more users/teams (from those that are members of that organization) that have the capabilities to update (dynamic or manually) an inventory. Administrators can also execute ad hoc commands for an inventory.

Editing Job Templates

System administrators, organization administrators, and project administrators, within a project under their administrative capabilities, may create and modify new job templates for that project.

When editing a job template, administrators (Tower, organization, and project) can select among the inventory and credentials in the organization for which they have usage capabilities or they may leave those fields blank so that they will be selected at runtime.

Additionally, they may specify one or more users/teams (from those that are members of that project) that have execution capabilities for that job template. The execution capability is valid regardless of any explicit capabilities the user/team may have been granted against the inventory or credential specified in the job template.

User View

A user can:

- See any organization or project for which they are a member
- Create their own credential objects which only belong to them
- See and execute any job template for which they have been granted execution capabilities

If a job template a user has been granted execution capabilities on does not specify an inventory or credential, the user will be prompted at run-time to select among the inventory and credentials in the organization they own or have been granted usage capabilities.

Users that are job template administrators can make changes to job templates; however, to change to the inventory, project, playbook, or credentials used in the job template, the user must also have the "Use" role for the project and inventory currently being used or being set.

24.2.3 Roles

As stated earlier in this documentation, all access that is granted to use, read, or write credentials is now handled through roles, and roles are defined for a resource.

Built-in roles

The following table lists the RBAC system roles and a brief description of the how that role is defined with regard to privileges in Tower.

System Role	What it can do
System Administrator - System wide singleton	Manages all aspects of the system
System Auditor - System wide singleton	Views all aspects of the system
Ad Hoc Role - Inventory	Runs ad hoc commands on an Inventory
Admin Role - Organizations, Teams, Inventory,	Manages all aspects of a defined Organization, Team, Inven-
Projects, Job Templates	tory, Project, or Job Template
Auditor Role - All	Views all aspects of a defined Organization, Project, Inven-
	tory, or Job Template
Execute Role - Job Templates	Runs assigned Job Template
Member Role - Organization, Team	User is a member of a defined Organization or Team
Read Role - Organizations, Teams, Inventory,	Views all aspects of a defined Organization, Team, Inventory,
Projects, Job Templates	Project, or Job Template
Update Role - Project	Updates the Project from the configured source control man-
	agement system
Update Role - Inventory	Updates the Inventory using the cloud source update system
Owner Role - Credential	Owns and manages all aspects of this Credential
Use Role - Credential, Inventory, Project	Uses the Credential, Inventory, or Project in a Job Template

A Singleton Role is a special role that grants system-wide permissions. Ansible Tower currently provides two built-in Singleton Roles but the ability to create or customize a Singleton Role is not supported at this time.

Common Team Roles - "Personas"

Tower support personnel typically works on ensuring that Tower is available and manages it a way to balance supportability and ease-of-use for users. Often, Ansible Tower support will assign "Organization Owner/Admin" to users in order to allow them to create a new Organization and add members from their team the respective access needed. This minimizes supporting individuals and focuses more on maintaining uptime of the service and assisting users who are using Ansible Tower.

Below are some common roles managed by the Tower Organization:

System Role (for Organizations)	Common User Roles	Description
Owner	Team Lead - Technical Lead	This user has the ability to control access for other users in their organization. They can add/remove and grant users specific access to projects, inventories, and job templates. This user also has the ability to create/remove/modify any aspect of an organization's projects, templates, inventories, teams, and credentials.
Auditor	Security Engineer - Project Manager	This account can view all aspects of the organization in read-only mode. This may be good for a user who checks in and maintains compliance. This might also be a good role for a service account who manages or ships job data from Ansible Tower to some other data collector.
Member - Team	All other users	These users by default as an organization member do not receive any access to any aspect of the organization. In order to grant them access the respective organization owner needs to add them to their respective team and grant them Admin, Execute, Use, Update, Ad-hoc permissions to each component of the organization's projects, inventories, and job templates.
Member - Team "Owner" 24.2. Role-Based Access	Controls	Organization Owners can provide "admin" through the team interface, over any component of their organization including projects, inventories, and job templates. These users are able to modify and utilize the respective component given access. 231
Member -	Developers -	This will be the most common and

24.3 Function of roles: editing and creating

A new organization "resource roles" functionality was introduced in Ansible Tower 3.3 that are specific to a certain resource type - such as workflows. Being a member of such a role usually provides two types of permissions, in the case of workflows, where a user is given a "workflow admin role" for the organization "Default":

- this user can create new workflows in the organization "Default"
- user can edit all workflows in the "Default" organization

One exception is job templates, where having the role is irrelevant of creation permission (more details on its own section).

24.3.1 Independence of resource roles and organization membership roles

Resource-specific organization roles are independent of the organization roles of admin and member. Having the "workflow admin role" for the "Default" organization will not allow a user to view all users in the organization, but having a "member" role in the "Default" organization will. The two types of roles are delegated independently of each other.

Necessary permissions to edit job templates

Users can edit fields not impacting job runs (non-sensitive fields) with a Job Template admin role alone. However, to edit fields that impact job runs in a job template, a user needs the following:

- **admin** role to the job template
- **use** role to related project
- use role to related inventory

An "organization job template admin" role was introduced, but having this role isn't sufficient by itself to edit a job template within the organization if the user does not have use role to the project / inventory a job template uses.

In order to delegate *full* job template control (within an organization) to a user or team, you will need grant the team or user all 3 organization-level roles:

- job template admin
- project admin
- inventory admin

This will ensure that the user (or all users who are members of the team with these roles) have full access to modify job templates in the organization. If a job template uses an inventory or project from another organization, the user with these organization roles may still not have permission to modify that job template. For clarity of managing permissions, it is best-practice to not mix projects / inventories from different organizations.

RBAC permissions

Each role should have a content object, for instance, the org admin role has a content object of the org. To delegate a role, you need admin permission to the content object, with some exceptions that would result in you being able to reset a user's password.

Parent is the organization.

Allow is what this new permission will explicitly allow.

Scope is the parent resource that this new role will be created on. Example: Organization. project_create_role.

An assumption is being made that the creator of the resource should be given the admin role for that resource. If there are any instances where resource creation does not also imply resource administration, they will be explicitly called out.

Here are the rules associated with each admin type:

Project Admin

- · Allow: Create, read, update, delete any project
- Scope: Organization
- User Interface: Project Add Screen Organizations

Inventory Admin

- Parent: Org admin
- · Allow: Create, read, update, delete any inventory
- Scope: Organization
- User Interface: Inventory Add Screen Organizations

Note: As it is with the **Use** role, if you give a user Project Admin and Inventory Admin, it allows them to create Job Templates (not workflows) for your organization.

Credential Admin

- Parent: Org admin
- Allow: Create, read, update, delete shared credentials
- Scope: Organization
- User Interface: Credential Add Screen Organizations

Notification Admin

- · Parent: Org admin
- Allow: Assignment of notifications
- Scope: Organization

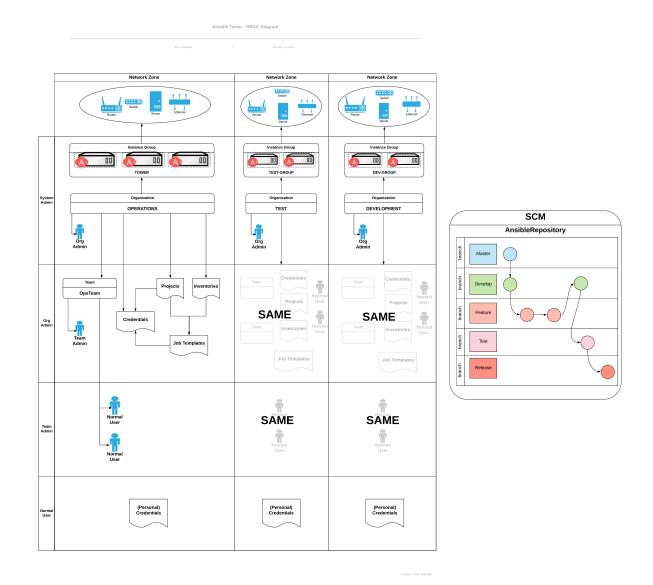
Workflow Admin

- Parent: Org admin
- Allow: Create a workflow
- Scope: Organization

Org Execute

- Parent: Org admin
- Allow: Executing JTs and WFJTs
- Scope: Organization

The following is a sample scenario showing an organization with its roles and which resource(s) each have access to:



CHAPTER

TWENTYFIVE

INDEX

• genindex

CHAPTER

TWENTYSIX

COPYRIGHT © 2019 RED HAT, INC.

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 a modified version of it, you must provide attribution to Red Hat, Inc. and provide a link to the original version.

Third Party Rights

Ubuntu and Canonical are registered trademarks of Canonical Ltd.

The CentOS Project is copyright protected. The CentOS Marks are trademarks of Red Hat, Inc. ("Red Hat").

Microsoft, Windows, Windows Azure, and Internet Explore are trademarks of Microsoft, Inc.

VMware is a registered trademark or trademark of VMware, Inc.

Rackspace trademarks, service marks, logos and domain names are either common-law trademarks/service marks or registered trademarks/service marks of Rackspace US, Inc., or its subsidiaries, and are protected by trademark and other laws in the United States and other countries.

Amazon Web Services", "AWS", "Amazon EC2", and "EC2", are trademarks of Amazon Web Services, Inc. or its affiliates.

OpenStackTM and OpenStack logo are trademarks of OpenStack, LLC.

ChromeTM and Google Compute EngineTM service 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.

INDEX

Symbols

|at|
 credential types,55

A

activity streams, 13 ad hoc commands, 115 inventories.115 add new inventories,93 smart inventories, 93 adding new applications, 74 credentials, 52 adding tokens applications, 74 admin menu, 19 Amazon Web Services credential types, 54 inventories, 108 Ansible Galaxy, 89 Ansible Galaxy integration features.3 Ansible Tower inventories.114 API considerations credential types,66 applications adding new, 74 adding tokens, 74 authentication, 73 create,74 getting started, 73 tokens, 73, 74 authentication, 73 applications, 73 automation features, 2 autoscaling best practices, 221 autoscaling flexibility features, 3

AWS cloud credentials, 145

В

backup and restore features, 3 best practices, 220 autoscaling, 221 deployment, continuous, 221 dynamic inventory sources, 220 file and directory structure, 220 host counts, larger, 221 integration, continuous, 221 source control, 220 variable inventory management, 221 bubblewrap functionality, 223 playbooks, 222 troubleshooting, 223 variables, 223

С

callbacks extra variables, 148 capacity jobs, 197 check job types, 118 cloud credentials AWS, 145 Google, 145 job templates, 143 MS Azure, 145 OpenStack, 144 Rackspace, 145 VMware, 146 cloud flexibility features, 3 CloudForms credential types, 61 components licenses,8

configure Tower settings menu, 20 create applications, 74 create template notifications, 201 creating new credential types, 68 credential types, 54, 66 |at|,55 Amazon Web Services, 54 API considerations, 66 CloudForms, 61 creating new, 68 Google Compute Engine, 55 insights, 56 machine, 57 Microsoft Azure Resource Manager, 58 network, 59 OpenStack, 61 oVirt, 62 Red Hat Satellite, 62 Red Hat Virtualization, 62 rhv. 62 source control, 63 Vault.64 VMware, 64 credentials, 50 adding new, 52 getting started, 51, 67 how they work, 50Insights, 209 types, 54 custom fact scan job, 140 custom fact scans playbook, 140 system tracking, 140 custom script inventories, 115

D

```
dashboard, 15
   host count, 15
   job status, 15
   jobs tab, 15
   main menu, 13
   schedule status, 15
DEB files
   licenses, 8
deployment, continuous
   best practices, 221
distributed
   job types, 151
```

dynamic inventory sources best practices, 220

E

Email notifications types, 202 environment, FIPS features, 6 evaluation, 7 extra variables callbacks, 148 provisioning callbacks, 148 surveys, 148, 157, 182 extra_vars, 148, 182

F

fact cache features,4 fact caching playbook, 141 fact scan job custom, 140 playbook, 138 fact scan playbook system tracking, 138 facts scan job templates, 141 features.6 Ansible Galaxy integration, 3 automation, 2 autoscaling flexibility, 3 backup and restore, 3 cloud flexibility, 3 environment, FIPS,6 fact cache,4 inventory sources, Red Hat CloudForms, 4 inventory sources, Red Hat Satellite 6,4 jobs, distribution, 6jobs, slicing, 6 notifications,4 OpenStack inventory support, 3 overview, 2 playbooks, Red Hat Insights, 4 real-time playbook, 2 remote command execution, 4 RESTful API, 3 role-based access control, 2 run-time job customization, 4 system tracking, 4 workflows, convergence nodes, 5 workflows, inventory overrides, 5 workflows, nesting, 5

file and directory structure
 best practices, 220
forks
 jobs, 197
functionality
 bubblewrap, 223

G

Galaxy support, 89 getting started applications, 73 credentials, 51, 67 Google cloud credentials, 145 Google Compute Engine credential types, 55 inventories, 109 groups notifications, 200

Η

Hipchat notifications types, 202 host count dashboard, 15 host counts, larger best practices, 221 hostname configuration notifications, 208 how they work credentials, 50

I

Insights credentials, 209 inventory, 212, 216 project, 211 projects, 209 insights credential types, 56 installation bundle licenses,8 instance groups, 183 integration, continuous best practices, 221 inventories,90 ad hoc commands, 115 add new, 93 Amazon Web Services, 108 Ansible Tower, 114 custom script, 115 Google Compute Engine, 109 groups, 99 groups; add new, 99

Microsoft Azure Classic (deprecated). 110 Microsoft Azure Resource Manager, 110 OpenStack, 113 project-sourced, 107 Red Hat CloudForms. 113 Red Hat Satellite 6,112 Red Hat Virtualization, 114 scan job templates, 138 smart, 92 VMware vCenter, 111 inventory Insights, 212, 216 inventory sources notifications, 200 inventory sources, Red Hat CloudForms features,4 inventory sources, Red Hat Satellite 6 features, 4 inventory sync job results, 189 IRC notifications types, 202

J

job results, 188 inventory sync, 189 job slice, 151 job splitting, 151 job status dashboard, 15 job templates, 118 cloud credentials, 143 job variables, 148 jobs, launching, 134 provisioning callbacks, 146 relaunch, 149 scheduling, 130 survey creation, 132 survey extra variables, 148 survey optional questions, 134 surveys, 132 job templates, hierarchy, 148 job templates, overview, 148 job types check, 118 distributed, 151 run, 118 scan, 118 slice, 151 splitting, 151 job variables job templates, 148

workflow templates, 182 jobs, 187 capacity, 197 event summary, 194 events summary, 193 forks, 197 host events, 196 host status bar, 194 host summary, 194 job summary, 193 notifications, 200 results, 188 views.15 jobs results playbook run, 192 SCM, 191 jobs tab dashboard, 15 jobs, distribution features, 6 jobs, launching job templates, 134 workflow templates, 180 jobs, slicing features, 6

L

```
license, 6, 7
   nodes, 8
   trial, 7
   troubleshooting, 11
   types, 7
license features, 6
license, add manually, 11
license, import, 10
licenses
   components, 8
   DEB files, 8
   installation bundle, 8
   RPM files, 8
logging in, 9
```

Μ

machine credential types, 57 main menu dashboard, 13 Mattermost notifications types, 202 Microsoft Azure Classic (deprecated) inventories, 110 Microsoft Azure Resource Manager credential types, 58

```
inventories,110
MS Azure
cloud credentials,145
my view,15
```

Ν

network credential types, 59 new schedule addition projects,88 notifications create template, 201 features,4 groups, 200 hostname configuration, 208 inventory sources, 200 jobs, 200 notifier,200 notifier hierarchy, 200 notifier workflow, 201 organizations, 34 resetting the TOWER_URL_BASE, 208 template, 201 troubleshooting TOWER_URL_BASE, 208 types, 202types Email, 202 types Hipchat, 202 types IRC, 202 types Mattermost, 202 types pagerduty, 202 types Rocket. Chat, 202 types Slack, 202 types Twilio, 202 types Webhook, 202 notifier notifications, 200 notifier hierarchy notifications, 200 notifier workflow notifications, 201

0

```
OpenStack
cloud credentials, 144
credential types, 61
inventories, 113
OpenStack inventory support
features, 3
ordering
sorting, 23
organization
summary, 35
organizations, 25
notifications, 34
```

```
permissions, 29
users, 27, 39
overview
features, 2
oVirt
credential types, 62
```

Ρ

pagerduty notifications types, 202 permissions organizations, 29 projects, 82 teams, 47 users, 39 playbook custom fact scans, 140 fact caching, 141 fact scan job, 138 scan job, 138 playbook run jobs results, 192 playbooks bubblewrap, 222 manage manually, 79 process isolation, 222 projects, 79, 80 PRoot settings, 222 sharing access, 222 sharing content, 222 source control, 80 playbooks, Red Hat Insights features,4 process isolation playbooks, 222 project Insights, 211 Scan, 213 project-sourced inventories, 107 projects,77 add new, 78 Insights, 209 new schedule addition, 88 permissions, 82 playbooks, 79, 80 source control update, 81 PRoot settings playbooks, 222 provisioning callbacks extra variables, 148 job templates, 146

R

Rackspace cloud credentials, 145 RBAC security, 224 real-time playbook features, 2 Red Hat CloudForms inventories.113 Red Hat Satellite credential types, 62Red Hat Satellite 6 inventories, 112 Red Hat Virtualization credential types, 62inventories, 114 relaunch job templates, 149 remote command execution features,4 resetting the TOWER URL BASE notifications. 208 RESTful API features, 3 rhv credential types, 62Rocket.Chat notifications types, 202 role-based access control features, 2 role-based access controls, 224 RPM files licenses,8 run job types, 118 run-time job customization features,4

S

Scan project, 213 scan job types, 118 scan job playbook, 138 scan job templates facts, 141 inventories, 138 schedule views, 15 schedule status dashboard, 15 scheduling add new, 130, 167

job templates, 130 workflow template, 167 workflow templates, 167 SCM jobs results, 191 searching, 21 security, 222 RBAC, 224 settings menu configure Tower, 20 view license, 20 sharing access playbooks, 222 sharing content playbooks, 222 Slack notifications types, 202 slice job types, 151 smart inventories, 92 smart inventories add new, 93 sorting ordering, 23 source control best practices, 220 credential types, 63 source control update projects, 81 splitting job types, 151 summary organization, 35 support, 6,7 survey extra variables job templates, 148 workflow templates, 182 workflows, 157 surveys creation, 132, 169 extra variables, 148, 157, 182 job templates, 132 optional questions, 134, 171 workflow templates, 169 system tracking custom fact scans, 140 fact scan playbook, 138 features,4 scan job, 118

Т

teams,44 permissions,47

users, 39, 45 template notifications, 201 token authentication, 73 tokens applications, 73, 74 Tower admin menu. 19 Tower settings menu, 20 trial.7 troubleshooting bubblewrap, 223 license, 11 troubleshooting TOWER_URL_BASE notifications, 208 Twilio notifications types, 202 types Email, notifications, 202 Hipchat, notifications, 202 IRC, notifications, 202 Mattermost, notifications, 202 notifications, 202 pagerduty, notifications, 202 Rocket.Chat.notifications.202 Slack, notifications, 202 Twilio, notifications, 202 Webhook, notifications, 202

U

updates,7 users,36 organizations,27,39 permissions,39 teams,39,45

V

variable inventory management best practices, 221 variable precedence, 148, 182 variables bubblewrap, 223 Vault credential types, 64 view license settings menu, 20 views jobs, 15 schedule, 15 visualizer workflow, 171 VMware cloud credentials, 146 credential types, 64

VMware vCenter

inventories, 111

W

```
Webhook
   notifications types, 202
workflow
   visualizer, 171
workflow job templates, 160
workflow template
   scheduling, 167
workflow templates
   job variables, 182
   jobs, launching, 180
   scheduling, 167
   survey creation, 169
   survey extra variables, 182
   survey optional questions, 171
   surveys, 169
   workflow visualizer, 171
workflow templates, hierarchy, 182
workflow templates, overview, 182
workflow visualizer
   workflow templates, 171
workflows, 154
   survey extra variables, 157
workflows, convergence nodes
   features, 5
workflows, inventory overrides
   features, 5
workflows, nesting
   features, 5
```