Ansible Tower 3.0.x Upgrade and Migration

Release Ansible Tower 3.0.3

Red Hat, Inc.

Jun 06, 2017

CONTENTS

1 Release Notes for Ansible Tower Version 3.0.3 **2**
- 1.1 Ansible Tower Version 3.0.3 2
- 1.2 Ansible Tower Version 3.0.2 3
- 1.3 Ansible Tower Version 3.0.1 4
- 1.4 Ansible Tower Version 3.0 5

2 Upgrading Ansible Tower **7**
- 2.1 Upgrade Planning 7
- 2.2 Obtaining Ansible Tower 7
- 2.3 Setting up Passwords 8
- 2.4 Setting up the Inventory File 8
- 2.5 The Setup Playbook 9

3 System Tracking Migration **11**

4 Role-Based Access Controls **12**
- 4.1 Enhanced and Simplified RBAC System 12
- 4.2 Specific Changes to Note 12

5 Job Template Changes **13**
- 5.1 Prompt on Launch 13
- 5.2 Permissions/RBAC Notes 15
- 5.3 Surveys 15

6 Job Output View Changes **17**
- 6.1 Results 19
- 6.2 Standard Out 20
- 6.3 Job Details View 21
- 6.4 Event Summary 24

7 Using virtualenv with Ansible Tower **27**
- 7.1 Modifying the virtualenv 27
- 7.2 Tower virtualenv under Red Hat Enterprise Linux 6 and derivatives 27

8 Index **28**

9 Copyright © 2016 Red Hat, Inc. **29**

Index **30**

Thank you for your interest in Ansible Tower by Red Hat. 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 3.0.x Upgrade and Migration Guide* discusses how to upgrade your Ansible Tower 2.4.4 (or later) system to the 3.0 version. The Ansible Tower 3.0 release introduces many updates and changes to Tower, including changes to the way upgrades are run and a completely rewritten RBAC system. This guide covers these and other related changes that you should keep in mind as you plan to migrate your data and prepare for this upgrade.

Ansible Tower Version 3.0.3; October 28, 2016; https://access.redhat.com/

CHAPTER

ONE

RELEASE NOTES FOR ANSIBLE TOWER VERSION 3.0.3

1.1 Ansible Tower Version 3.0.3

- Added support for new AWS regions, including an update to the boto version included with Tower
- Fixed various minor UI and API related bugs
- Fixed a regression with authentication restrictions
- Fixed an issue where restoring the database failed when using the RHEL6 bundled installation method
- Fixed an issue where, when viewing a host, “extra vars” were not initially formatted properly
- Fixed an issue where users were able to relaunch jobs they did not have permission to initially launch
- Fixed an issue where, after editing a Job Template, retrieving Job Templates failed when filtered
- Fixed an issue where Satellite 6 inventory marked all hosts as disabled
- Fixed an issue where Inventory variables were displayed incorrectly when editing hosts
- Fixed a rendering issue with the Host Event details window
- Fixed an issue where, when launching an inventory update, users were navigated away from the inventory manage view
- Fixed an issue where organization auditors could see the user permissions of other users in their organization
- Fixed an issue where canceling a Windows job in Tower left an orphaned process running on the control machine
- Fixed an issue where empty Host Variable Data produces a 500 error in the API browser after upgrading from 2.4.5
- Fixed an issue when using an Azure Service Principal in conjunction with Microsoft Azure inventory
- Fixed an issue where Inventory syncs fail against a resource group if it contains a non-standard virtual machine size when using Azure
- Fixed an issue where navigating to the admin or users from the organizations view in Tower caused 404 errors
- Fixed an issue where, when updating a Rackspace inventory, TypeError messages appeared
- Improved the run time performance for playbooks in Tower
- Improved support around how YAML is handled with Tower’s variable parser
- Improved the population of manual projects in Tower
- Improved Event Summary status badge counts
- Improved PostgreSQL configuration with regard to authentication (CVE-2016-7070)
- Updated PostgreSQL repository location for installation methods

1.2 Ansible Tower Version 3.0.2

- Added support for IAM Roles when configuring an EC2 Inventory Sync
- Added support for backing up and restoring Databases created when installing 3.0.x
- Added the display of a "working" indicator when toggling Tower components on/off
- Added the ability to toggle the view of job labels (view less/view more)
- Added the ability to add skip tags to job templates (which may also be prompted for at launch time)
- Added documentation around resetting the Tower URL provided in Notification links
- Fixed an issue where users could not remove inventory or credentials from job template
- Fixed an issue where admins were not properly allowed to copy or edit to Job Templates via the API
- Fixed an issue where Home/Host column views were not sortable
- Fixed the display of schedules to only show those with future activity
- Fixed an error where clicking to a different page number while editing a resource and making a new selections indicated an item other than the one currently selected/being edited
- Fixed an issue where relaunching a job ignored search filters
- Fixed an issue where searching for a user on an inventory permission page queried a project access list URL instead of the inventory access list URL
- Fixed an issue where pressing the Enter key (instead of clicking 'Ok' with your mouse) closes a pop up error message and, unexpectedly, navigates the user back to the Tower home page
- Fixed an issue where system job templates were not being included when viewing unified job template results
- Fixed an issue related to relaunching ad hoc commands
- Fixed an issue preventing projects from being deleted during an SCM update
- Fixed an issue where, when viewing the "Event Summary" field, filtering by task status summary dots returned incorrect tasks information
- Fixed an issue where selecting a host on one page, then going to the next page and selecting another host, did not save the prior selection as expected
- Fixed an issue where processing extra_vars in a survey caused errors
- Fixed an issue regarding how passwords are stored with surveys
- Fixed an issue where, when running a playbook with an ignored task, the ignored task was incorrectly marked as failing
- Fixed an issue so that Webhook notifications properly display the host summary information
- Fixed an issue where provisioning callbacks were running multiple times in a row
- Fixed various minor issues related to RBAC permissions and credentials
- Fixed various minor API bugs
- Fixed various minor UI and tooltips bugs
- Fixed an issue related to SAML logins hanging after multiple authorization attempts
- Fixed an issue where the "start date" header and schedule preview do not match what is set by the browser locale
- Fixed an issue where users could not properly edit their profile

- Fixed an issue related to backup/restoring with the setup.sh script
- Improved Tower installer compatibility with RHUI repos on RHEL non-AWS instances
- Improved upon what the auditor role can view (organization auditors can view inventory script contents in their own organizations, view notification templates in the activity stream, team credentials views)
- Improved the consistency of how scheduling is displayed within the Tower UI
- Improved how credentials are handled in that they should only be shareable when the organization field is not "null"
- Improved how teams are displayed for different organizations when viewing permissions
- Improved support for CloudForms and Red Hat Satellite 6 with Tower 3.0.x
- Reorganized activity stream views/access for organization admins and auditors
- Removed the requirement of needing a password for the network credential when using an SSH key
- Removed the requirement of needing AUTH with Email notifications using SMTP

1.3 Ansible Tower Version 3.0.1

- Added a stock schedule job for the 'Cleanup Fact Details' management job
- Fixed an issue with inventory syncs using Red Hat Satellite 6 credentials
- Fixed an issue which incorrectly allowed users assigned to a system auditor role to be able to escalate privileges to teams
- Fixed an issue with Webhook notifications where the content-type was being set incorrectly
- Fixed an issue where canceling a new job failed to change state from "new" to "canceled"
- Fixed an upgrade and credential migration issue which involved null inventory fields in job templates
- Fixed an upgrade and migration issue where hosts which had previously been deleted were not skipped during the upgrade process
- Fixed an upgrade and migration issue where job templates linked to deleted inventories caused migrations to fail
- Fixed an upgrade and migration issue where job templates without inventories caused migrations to fail
- Fixed an error related to the logging of RBAC migration data which caused installations to fail
- Fixed an issue related to license checks
- Fixed other various issues related to upgrading and migration
- Fixed the need for elevated permissions to make changes to job templates under some scenarios
- Fixed an issue where Organization-level admins could not edit scan jobs that were created prior to upgrading to Tower 3.0
- Fixed an issue regarding Software Collections (SCL) installation on EL6
- Fixed a problem with subsequent logins after upgrading to Tower 3.0 when using Google OAuth or SAML authentication
- Discovered an issue with MS Azure inventory imports using new-style credentials being unsupported on distributions that ship python-2.7 (e.g. not EL6)
- Updated the UI to display new jobs in the Jobs overview screen and added a cancellation method for these new jobs

1.4 Ansible Tower Version 3.0

- Added a notifications system for Tower which supports services like Slack, HipChat, IRC, etc.
- Added support for the new Azure inventory system and the latest Ansible Azure modules (legacy Azure inventory and credentials are still supported)
  - Azure inventory imports using new-style credentials are only supported on distributions that ship python-2.7 (e.g. not EL6)
- Added support for keystone v3 which supports the latest Openstack versions
- Added counts and more detail to Organization endpoints (API)
- Added prompting for Job Templates
- Added labels for Job Templates
- Added support for user customization as Ansible tasks now run in their own environment
- Added support for new Ansible Network Credentials
- Added inventory support for Red Hat Cloudforms and Red Hat Satellite 6
- Added SUSE, OpenSuse, and Debian support for scan jobs
- Added a link to the schedule in the job detail view if the job was started as a result of a schedule
- Added survey spec management without requiring that surveys be enabled on job templates
- Added additional strict extra_vars validation. 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
- Added a deprecation notice for Ubuntu 12 and RHEL 6
- Changed how Projects are linked so that they now tie singularly to an Organization
- Changed how system tracking and scan data are stored–now in postgres. MongoDB dependency removed.
- Discovered an issue with ECDSA credentials–if your Tower server has a version of OpenSSH that predates 5.7, jobs will fail when launched jobs with ECDSA credentials
- Fixed issues with scan jobs on RHEL5
- Fixed an issue with the websocket service when Tower is run on CentOS or RHEL 7.2
- Fixed issues with Ansible's no_log causing errors or not hiding data when running jobs
- Fixed the way setting a license is done so that it propagates to standby Tower nodes in an HA configuration
- Fixed GCE credential handling and inventory filtering
- Improved (through a complete rewrite to expand and simplify) the Role-Based Access Control system in Tower
- Improved job templates so that multiple invocations of the same job template will only block if the job templates used the same inventory
- Improved the setup playbook so that it now hides potentially sensitive information from stdout and the setup log
- Improved the Setup process now supports installing and configuring postgres on a remote system
- Removed MongoDB and changed view queries to use a Postgres implementation

- Removed soft-deletes: Tower now permanently deletes removed objects and the utilities to manage the cleanup of those soft-deleted objects have been removed
- Removed Munin monitoring
- Updated the look and feel of the entire Tower UI for a more approachable and intuitive user experience
- Updated and simplifed the Tower setup process so that new Tower installs are now preloaded with Organization, Inventory, Project, and Job Template demo data
- Updated the setup process to support installing and configuring Postgres on a remote system
- Updated dependencies
- Updated Red Hat Enterprise Linux 6/CentOS 6 to use python 2.7 (for Tower only)
- Updated the minimum open file descriptor check and configuration by raising it from 1024 to 4096

CHAPTER

TWO

UPGRADING ANSIBLE TOWER

Topics:

- *Upgrade Planning*
- *Obtaining Ansible Tower*
- *Setting up Passwords*
- *Setting up the Inventory File*
- *The Setup Playbook*

2.1 Upgrade Planning

This section covers changes that you should keep in mind as you attempt to upgrade your Ansible Tower Instance

- If you are not yet using a 2.4.x version of Ansible Tower, **do not** attempt to upgrade directly to Ansible Tower 3.0. You must start with a system which has a verison of Tower 2.4.x installed or the upgrade will fail.
- Ansible Tower 3.0 simplifies installation and removes the need to run `./configure/` as part of the initial setup.
- The file `tower_setup_conf.yml` is no long used. Instead, you should now edit the **inventory** file in the `/ansible-tower-setup-<tower_version>/` directory.
- Earlier version of Tower used MongoDB when setting up an initial database; please note that Ansible Tower 3.0 has replaced the use of MongoDB with PostgreSQL.

2.2 Obtaining Ansible Tower

Download and then extract the Ansible Tower installation/upgrade tool: http://releases.ansible.com/ansible-tower/setup/

```
root@localhost:~$ tar xvzf ansible-tower-setup-latest.tar.gz
root@localhost:~$ cd ansible-tower-setup-<tower_version>
```

To install or upgrade, start by editing the inventory file in the `ansible-tower-setup-<tower_version>` directory, replacing `<tower_version>` with the version number, such as `2.4.5` or `3.0.0`. directory.

2.3 Setting up Passwords

2.4 Setting up the Inventory File

As you edit your inventory file, there are a few things you must keep in mind:

- The contents of the inventory file should be defined in `./inventory`, next to the `./setup.sh` installer playbook.
- For **installations and upgrades**: If you need to make use of external databases, you must ensure the database sections of your inventory file are properly setup. Edit this file and add your external database information before running the setup script.
- For **redundant installations**: If you are creating a redundant setup, you must replace `localhost` with the hostname or IP address of the primary instance.
- For **installations**: When performing an installation, you must supply any necessary passwords in the inventory file.
- For **upgrades**: When upgrading, your prior configuration should migrate over and filling out the passwords in the inventory file should be unnecessary.

```
admin_password='password'
redis_password='password'
pg_password='password'
```

Note: Redis passwords must not contain spaces or any of the following characters: `@`, `:`, `-`, `\`, `/`, `#`

For example, `Passw0rd` is acceptable, but `P@ssword` is not. (It is not recommended that you use this example password as anything other than an example. Setting your actual password to anything other than a secure password that is only known by you or your trusted administration staff is extremly dangerous.)

Example Inventory file

```
[primary]
localhost ansible_connection=local

[secondary]

[database]

[all:vars]
admin_password='password'
redis_password='password'

pg_host=''
pg_port=''

pg_database='awx'
pg_username='awx'
pg_password='password'
```

Example Inventory file for an external existing database

```
[primary]
node1.example.com ansible_connection=local
```

```
[secondary]
node2.example.com

[database]

[all:vars]
admin_password='password'
redis_password='password'

pg_host='database.example.com'
pg_port='5432'

pg_database='awx'
pg_username='awx'
pg_password='password'
```

Example Inventory file for external database which needs installation

```
[primary]
node1.example.com ansible_connection=local

[secondary]
node2.example.com

[database]
database.example.com

[all:vars]
admin_password='password'
redis_password='password'

pg_host='database.example.com'
pg_port='5432'

pg_database='awx'
pg_username='awx'
pg_password='password'
```

Once any necessary changes have been made, you are ready to run `./setup.sh`.

Note: Root access to the remote machines is required. With Ansible, this can be achieved in different ways:

- ansible_ssh_user=root ansible_ssh_password="your_password_here" inventory host or group variables
- ansible_ssh_user=root ansible_ssh_private_key_file="path_to_your_keyfile.pem" inventory host or group variables
- ANSIBLE_BECOME_METHOD='sudo' ANSIBLE_BECOME=True ./setup.sh
- ANSIBLE_SUDO=True ./setup.sh

2.5 The Setup Playbook

Note: Ansible Tower 3.0 simplifies installation and removes the need to run `./configure/` as part of the installation setup. Users of older versions should follow the instructions available in the v.2.4.5 (or earlier) releases of the Tower Documentation available at: http://docs.ansible.com/

The Tower setup playbook script uses the `inventory` file and is invoked as `./setup.sh` from the path where you unpacked the Tower installer tarball.

```
root@localhost:~$ ./setup.sh
```

The setup script takes the following arguments:

- `-h` – Show this help message and exit
- `-i INVENTORY_FILE` – Path to Ansible inventory file (default: `inventory`)
- `-e EXTRA_VARS` – Set additional Ansible variables as key=value or YAML/JSON (i.e. `-e bundle_install=false` forces an online installation)
- `-b` – Perform a database backup in lieu of installing
- `-r` – Perform a database restore in lieu of installing (a default restore path is used unless EXTRA_VARS are provided with a non-default path, as shown in the code example below)

```
./setup.sh -e 'restore_backup_file=/path/to/nondefault/location' -r
```

Note: Please note that a issue was discovered in Tower 3.0.0 and 3.0.1 that prevented proper system backups and retorations.

If you need to back up or restore your Tower v3.0.0 or v3.0.1 installation, use the v3.0.2 installer to do so.

CHAPTER

THREE

SYSTEM TRACKING MIGRATION

Once your system has been upgraded to Ansible Tower 3.0, you will find that your system tracking data has been migrated from MongoDB to PostgreSQL. If you want to delete the old data in MongoDB, you can do so manually. First, connect to your mongo database using the mongo command line client, then run the following commands:

```
$ use system_tracking
$ db.runCommand( { dropDatabase: 1 } )
```

At this point, you can also remove the MongoDB packages.

CHAPTER FOUR

ROLE-BASED ACCESS CONTROLS

Ansible Tower 3.0 has changed significantly around the way that the Role-Based Access Control (RBAC) system works. For the latest RBAC documentation, refer to the Role-Based Access Controls section in the Tower User Guide.

4.1 Enhanced and Simplified RBAC System

Bason 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.)

4.2 Specific Changes to Note

There are a few changes you should keep in mind as you work with the RBAC system as redesigned for Ansible Tower version 3.0:

- 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.
- Deletion of job run data is now restricted to system and organization administrators.
- Projects no longer have multiple organizations. You *must* provide an organization when creating a new project through the API:

```
- projects/:id/organizations --> removed
```

- New Auditor type in Tower has been added which can see all aspects of the systems automation but does not have permission to run or change things.

CHAPTER

FIVE

JOB TEMPLATE CHANGES

Job templates have been updated in Tower 3.0 to allow you more flexibility when creating and working with them.

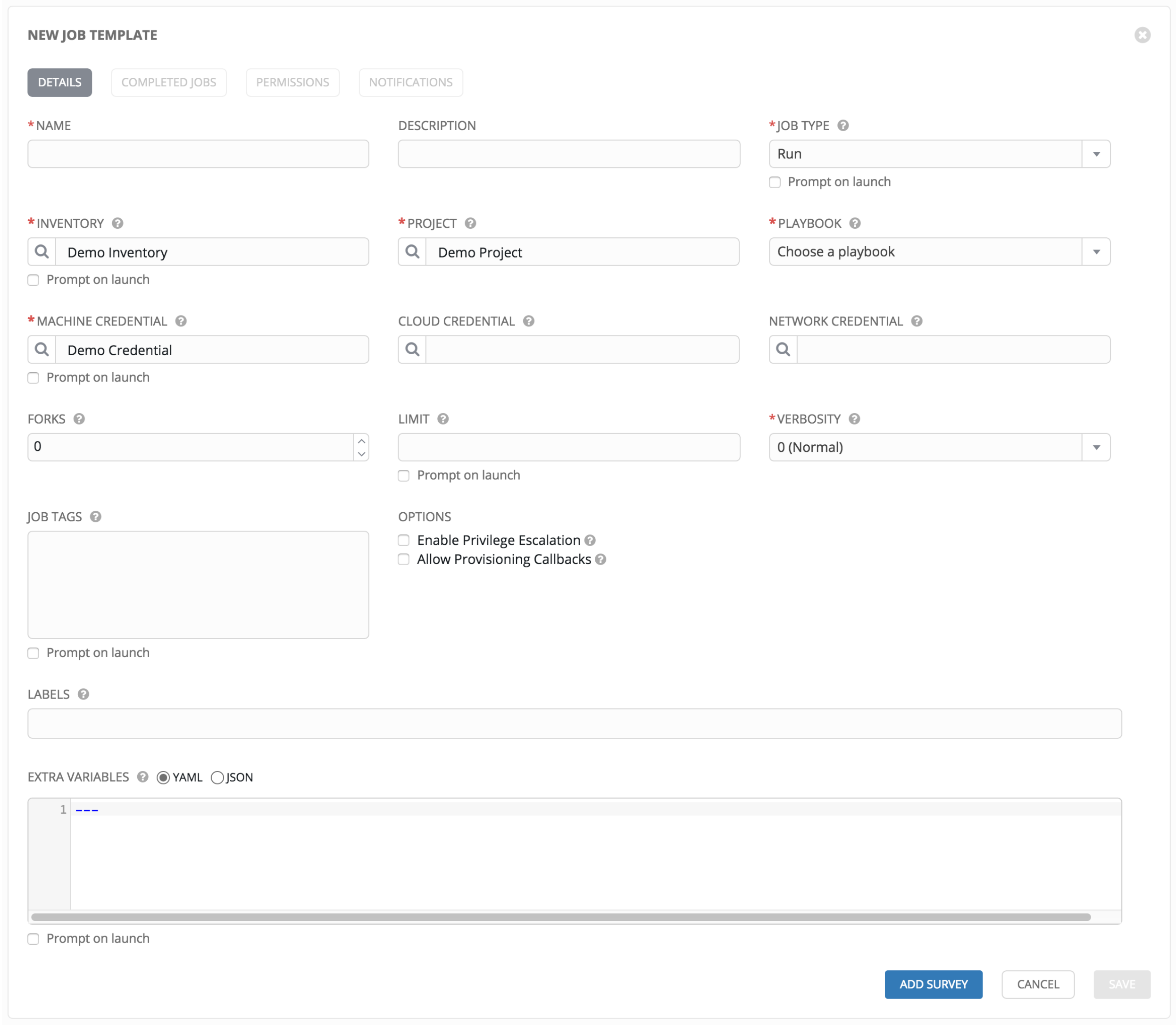
5.1 Prompt on Launch

In prior versions of Ansible Tower, you could set "Prompt on Launch" against Extra Varaibles that you want to potentially pass through the job template. Now, starting with version 3.0, Ansible Tower allows you to prompt for an inventory selection, job type, and more.

Selecting "Prompt on Launch" means that even if a value is supplied at the time of the job template creation, the user launching the job will be prompted to supply new information or confirm what was entered in the job template originally.

The following job template settings allow for prompting at the time of launch:

- Job Type (run or check type jobs only, as scan jobs cannot be changed at the time of launch)
- Inventory
- Machine credential
- Limit
- Job Tags
- Extra variables



*NAME:	DESCRIPTION:	*JOB TYPE: Run ☐ Prompt on launch
*INVENTORY: Demo Inventory ☐ Prompt on launch	*PROJECT: Demo Project	*PLAYBOOK: Choose a playbook
*MACHINE CREDENTIAL: Demo Credential ☐ Prompt on launch	CLOUD CREDENTIAL:	NETWORK CREDENTIAL:
FORKS: 0	LIMIT: ☐ Prompt on launch	*VERBOSITY: 0 (Normal)
JOB TAGS: ☐ Prompt on launch	OPTIONS: ☐ Enable Privilege Escalation ☐ Allow Provisioning Callbacks	

LABELS

EXTRA VARIABLES [✓ YAML] JSON

```
---
```

☐ Prompt on launch

As you work with migrating your Tower 2.4.5 job templates to 3.0, please keep in mind the following:

- All "Prompt on Launch" fields are set to *False* by default after migrating to 3.0 (new job templates created also have all "Prompt on Launch" fields set to *False* by default).
  - With one exception for those upgrading from 2.4.5 to 3.0: if a credential used in Tower 2.4.5 was null, the credential will be prompted for in 3.0.
- If you have Job Templates with a null credential, in the migration from 2.4.5 to 3.0, "ask_credential_on_launch" is set to *True*.
  - Note that there was no way to set a default credential in 2.4.5. However, in 3.0, you can set a default credential and select to prompt the user at launch time to confirm the default credential or change it to something new.
- All other "ask_xx_on_launch" prompts are set to *False*.
- Starting with Tower 3.0, if "ask_variables_on_launch" is set to *False*, extra variables passed at launch time (via UI or API) that are not part of an enabled survey are ignored.
- While there are no changes to how "ask_variables_on_launch" behaves, keep in mind that these variables combine with survey answers.

5.2 Permissions/RBAC Notes

Job template visibility is no longer configured via a combination of permissions on inventory, projects, and credentials. Admins who want to give any user or team permissions to use a job template can quickly 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.

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.

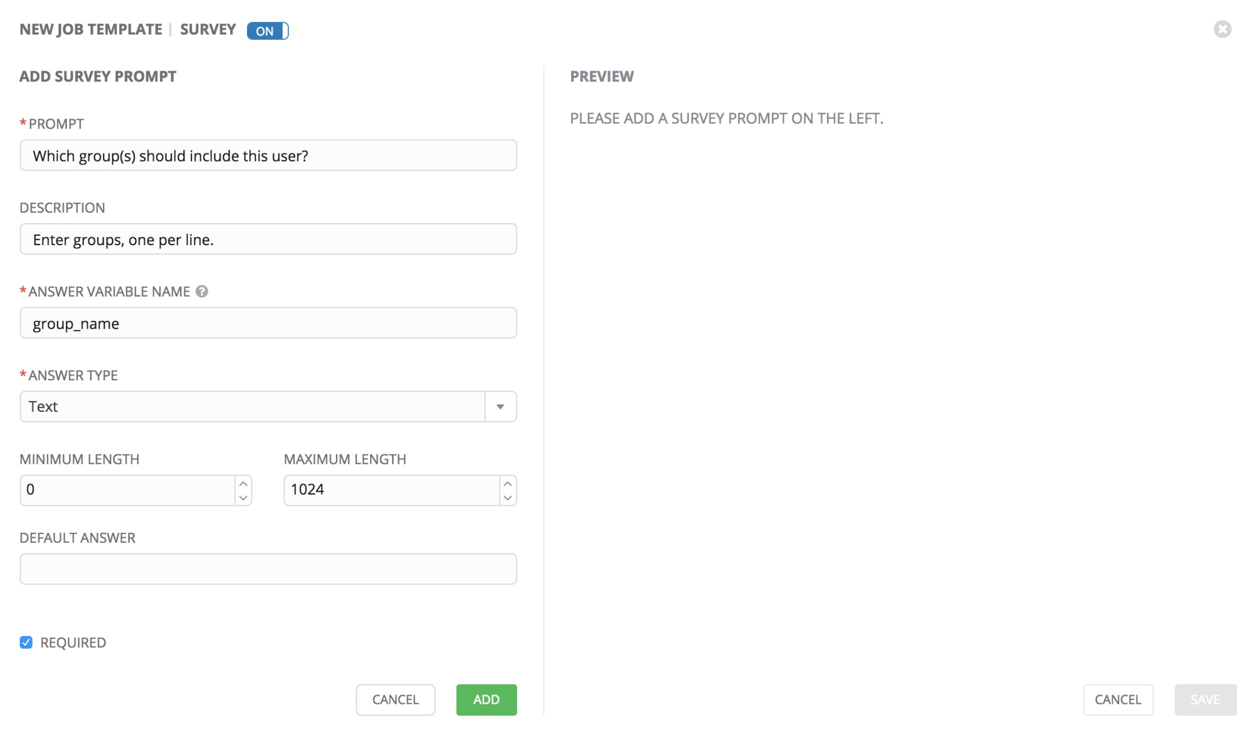
Users that are job template administrators can make changes to job templates; however, to make changes to the inventory, project, playbook, or credentials used in the job template, the user must also have the "Use" role for the project, inventory, and all credentials currently being used or being set.

5.3 Surveys

In prior versions of Ansible Tower, you had to select a checkbox to "Enable Survey" on the Job Template before a button appeared allowing you to "Create Survey".

Enabling and creating surveys is much simplier in Ansible Tower 3.0.

At the bottom of each job template is a button (ADD SURVEY) which opens a new dialog where you can enter your survey questions and reposnses.



PLEASE ADD A SURVEY PROMPT ON THE LEFT.

*PROMPT: Which group(s) should include this user?

DESCRIPTION: Enter groups, one per line.

*ANSWER VARIABLE NAME: group_name

*ANSWER TYPE: Text

MINIMUM LENGTH: 0

MAXIMUM LENGTH: 1024

DEFAULT ANSWER:

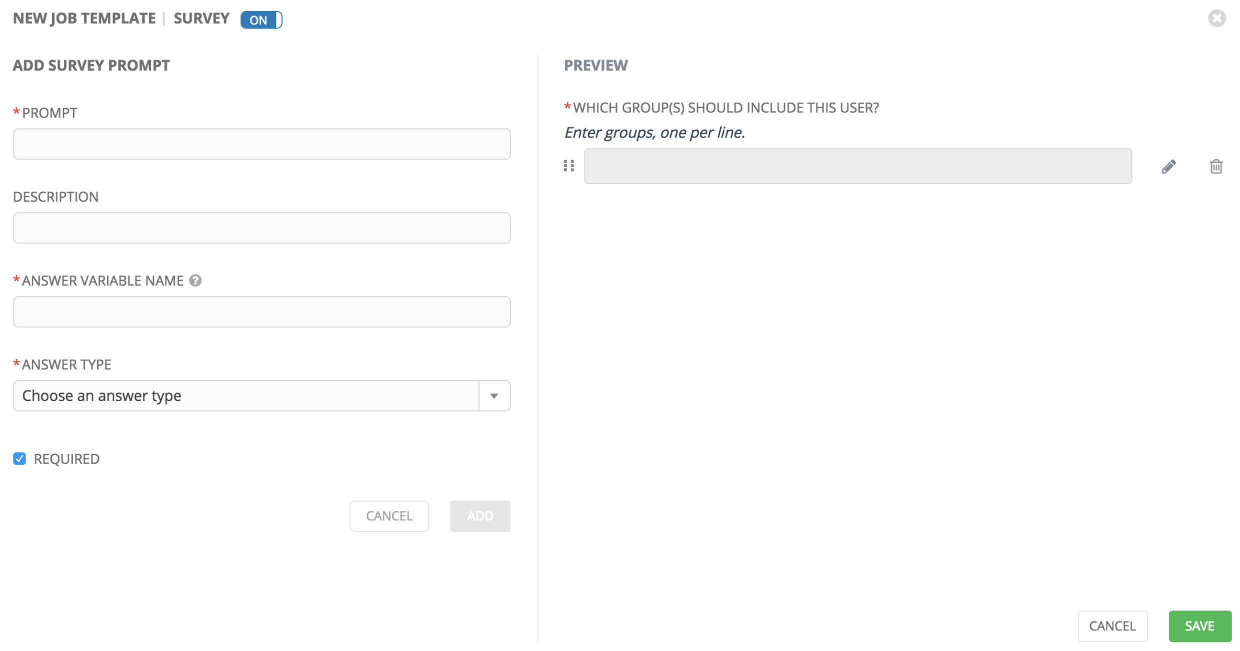
☑ REQUIRED

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

Once you have entered the question information, click **Add** to add the survey prompt.

A stylized preview of the survey is presented, along with a **New Question** button. Click this button to add additional questions.

For any question, you can click on the **Edit** button to edit the question, the **Delete** button to delete the question, and click on the Up and Down arrow buttons to rearrange the order of the questions. Click **Save** to save the survey.



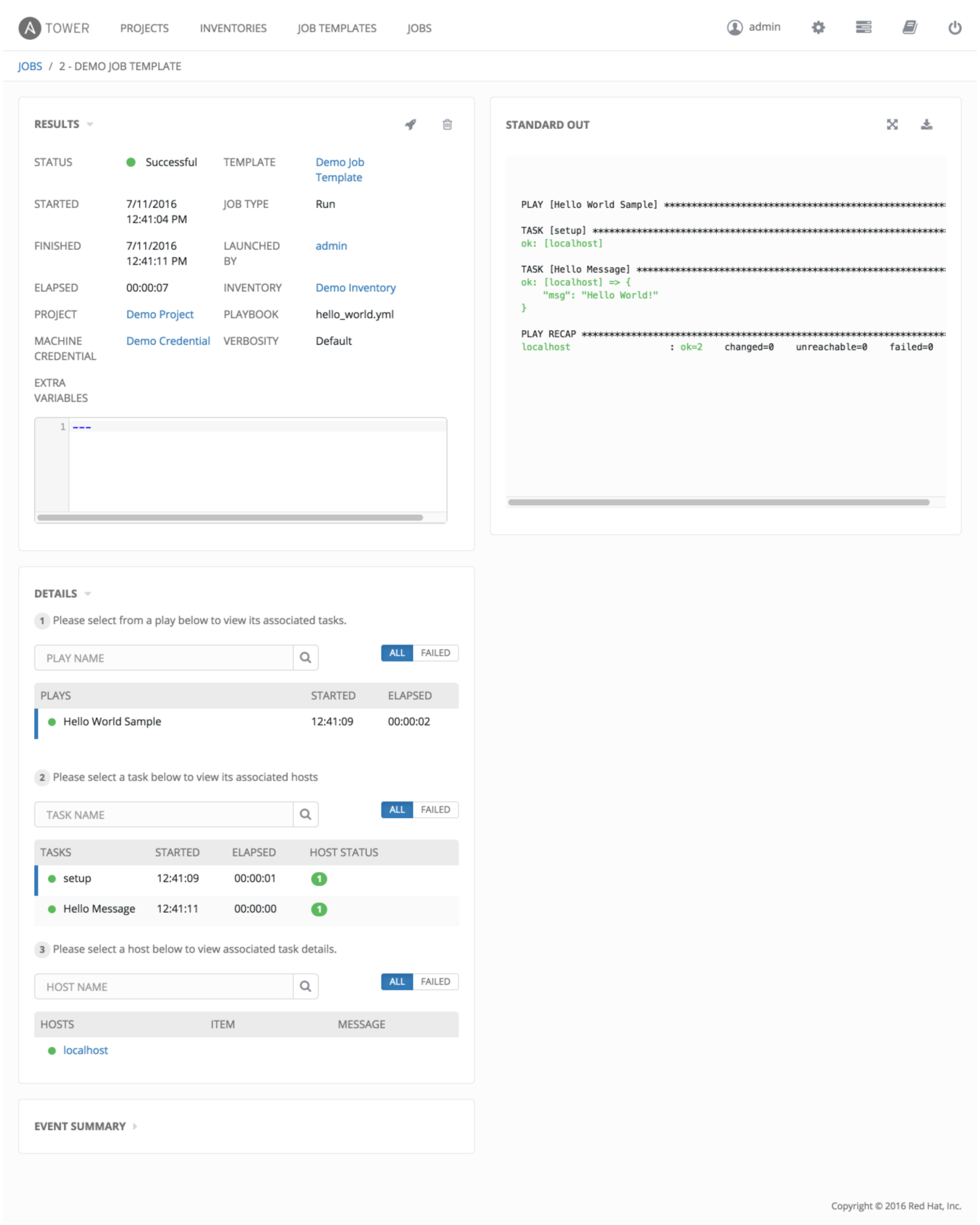
CHAPTER

SIX

JOB OUTPUT VIEW CHANGES

With the update of the overall Tower user interface, it is worth noting the changes to how job results are displayed.

Job results for inventory syncs and SCM updates only show the Results and Standard Out of the job recently Run. Job results for playbook runs consist of Results, Standard Out, Details, and the Event Summary.



STATUS	● Successful	TEMPLATE	Demo Job Template
STARTED	7/11/2016 12:41:04 PM	JOB TYPE	Run
FINISHED	7/11/2016 12:41:11 PM	LAUNCHED BY	admin
ELAPSED	00:00:07	INVENTORY	Demo Inventory
PROJECT	Demo Project	PLAYBOOK	hello_world.yml
MACHINE CREDENTIAL	Demo Credential	VERBOSITY	Default

EXTRA VARIABLES

```
---
```

```
PLAY [Hello World Sample] ******************************************************

TASK [setup] *******************************************************************
ok: [localhost]

TASK [Hello Message] ***********************************************************
ok: [localhost] => {
    "msg": "Hello World!"
}

PLAY RECAP *********************************************************************
localhost                  : ok=2    changed=0    unreachable=0    failed=0
```

1 Please select from a play below to view its associated tasks.

PLAY NAME

PLAYS	STARTED	ELAPSED
● Hello World Sample	12:41:09	00:00:02

2 Please select a task below to view its associated hosts

TASK NAME

TASKS	STARTED	ELAPSED	HOST STATUS
● setup	12:41:09	00:00:01	1
● Hello Message	12:41:11	00:00:00	1

3 Please select a host below to view associated task details.

HOST NAME

HOSTS	ITEM	MESSAGE
● localhost		

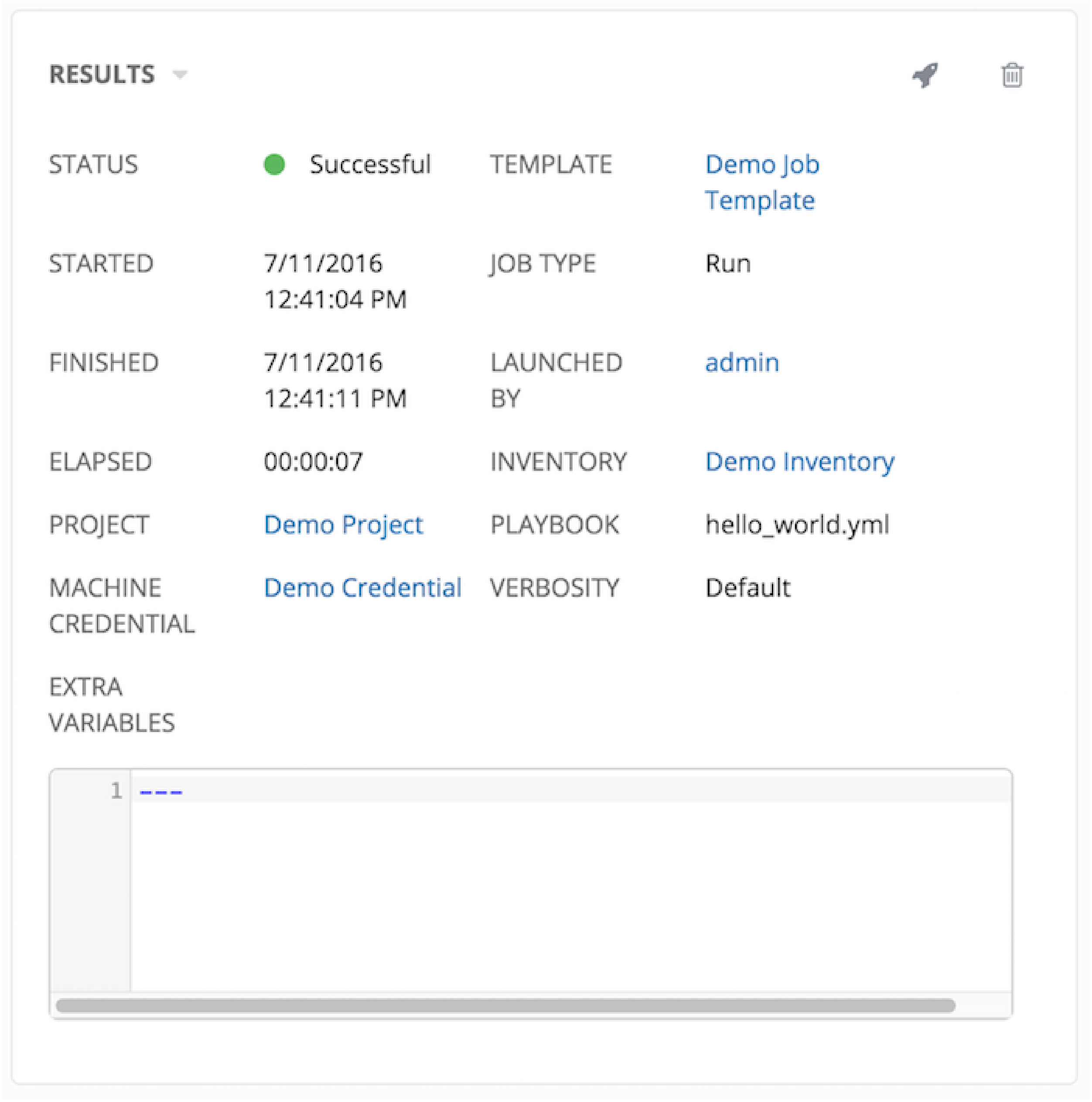
Copyright © 2016 Red Hat, Inc.

For more details regarding Job Results, refer to Jobs in the *Ansible Tower User Guide*.

6.1 Results

The **Results** area shows the basic status of the job (*Running*, *Pending*, *Successful*, or *Failed*), its start and end times, which template was used, how long the job run took, who launched it, and more. The buttons in the top right of the Results view allow you to relaunch or delete the job.

By clicking on these Results entries, where appropriate, you can view the corresponding job templates, projects, and other Tower objects.



STATUS	● Successful	TEMPLATE	Demo Job Template
STARTED	7/11/2016 12:41:04 PM	JOB TYPE	Run
FINISHED	7/11/2016 12:41:11 PM	LAUNCHED BY	admin
ELAPSED	00:00:07	INVENTORY	Demo Inventory
PROJECT	Demo Project	PLAYBOOK	hello_world.yml
MACHINE CREDENTIAL	Demo Credential	VERBOSITY	Default

EXTRA VARIABLES

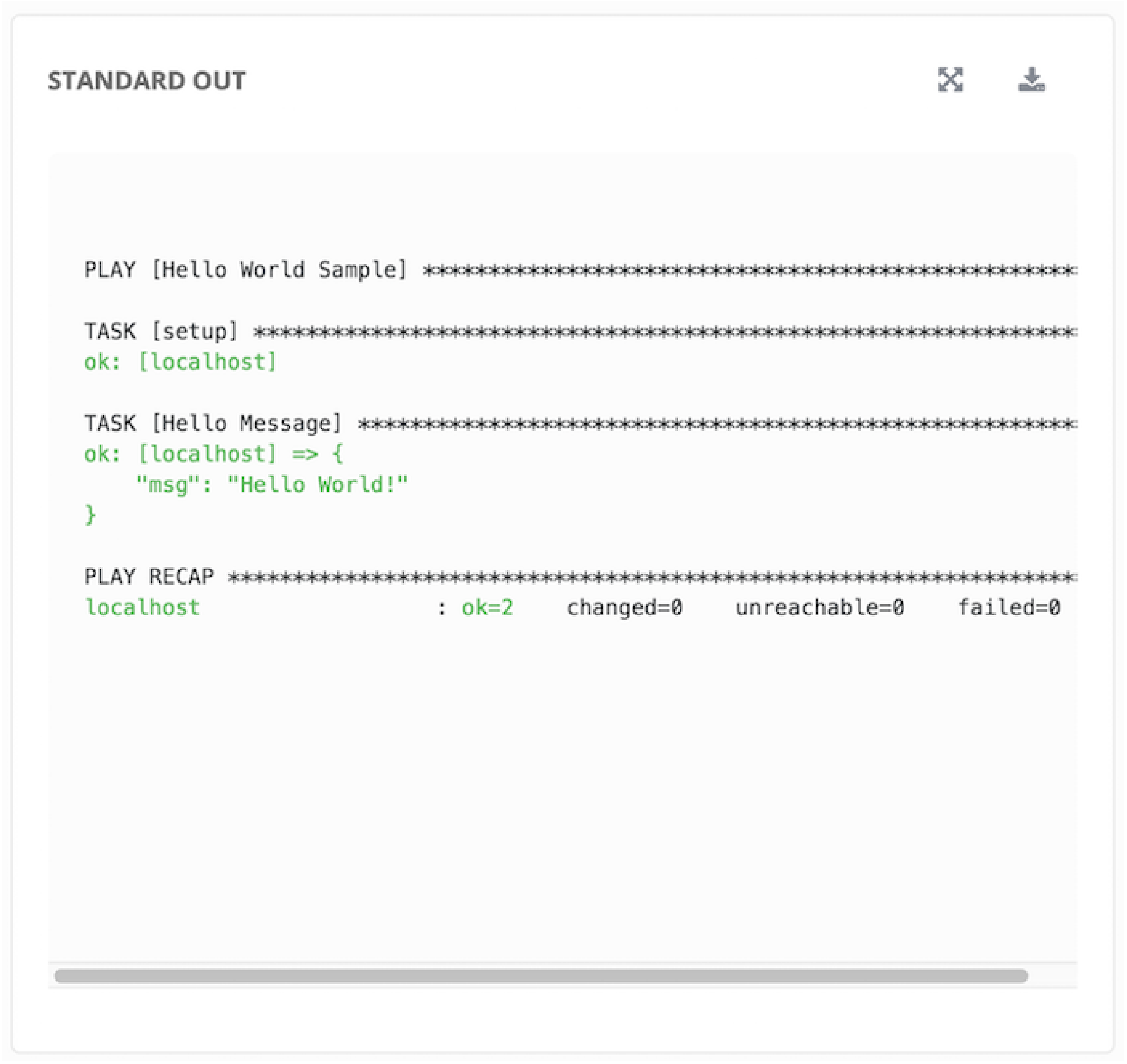
```
---
```

6.2 Standard Out

The **Standard Out** display shows the full results of running the SCM Update or Inventory Sync playbook. This shows the same information you would see if you ran the Ansible playbook using Ansible from the command line, and can be useful for debugging.

Prior to Ansible Tower 3.0, the Standard Out was a separate display and was not included in the overall job results view.

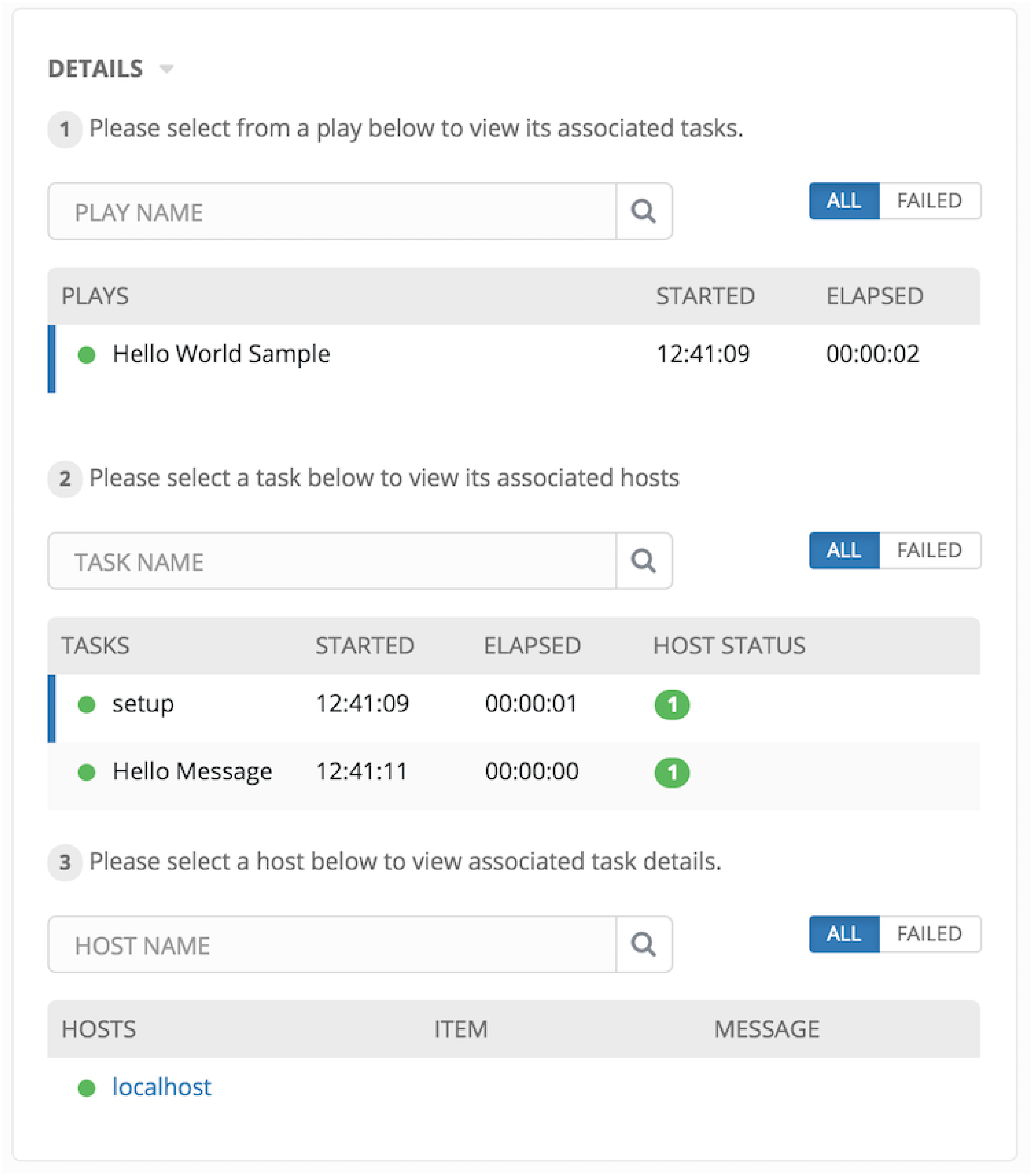
The buttons in the top right corner of the Standard Out display allow you to toggle the output as a main view or to download the output.



```
PLAY [Hello World Sample] ******************************************************

TASK [setup] *******************************************************************
ok: [localhost]

TASK [Hello Message] ***********************************************************
ok: [localhost] => {
    "msg": "Hello World!"
}

PLAY RECAP *********************************************************************
localhost                  : ok=2    changed=0    unreachable=0    failed=0
```

6.3 Job Details View

The Job Details view in Ansible Tower 3.0 now offers step-by-step views into Plays, Tasks, and Hosts, walking you through each section of the job results until you drill down into your host informaiton.


DETAILS

1 Please select from a play below to view its associated tasks.

PLAY NAME

ALL FAILED

PLAYS	STARTED	ELAPSED
● Hello World Sample	12:41:09	00:00:02

2 Please select a task below to view its associated hosts

TASK NAME

ALL FAILED

TASKS	STARTED	ELAPSED	HOST STATUS
● setup	12:41:09	00:00:01	1
● Hello Message	12:41:11	00:00:00	1

3 Please select a host below to view associated task details.

HOST NAME

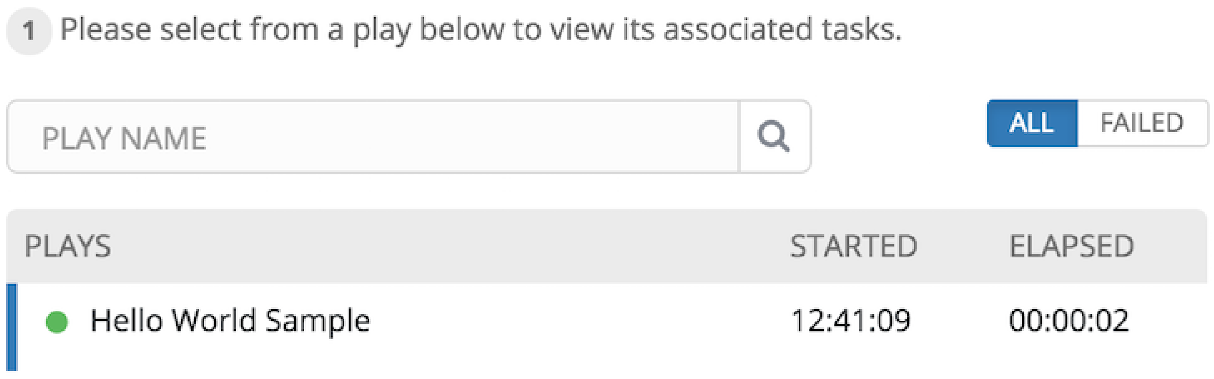
ALL FAILED

HOSTS	ITEM	MESSAGE
● localhost		

6.3.1 Plays

The **Plays** area shows the plays that were run as part of this playbook. The displayed plays can be filtered by **Play Name**, and can be limited to show only failed plays (using the **ALL/FAIL** view toggle).

For each play, Tower shows the **Play Name**, start time for the play, the elapsed time of the play, the play **Name**, and whether the play succeeded or failed (indicated by the status dot to the left of the **pPlay Name**). Clicking on a specific play filters the **Tasks** and **Host Events** area to only display tasks and hosts relative to that selected play.

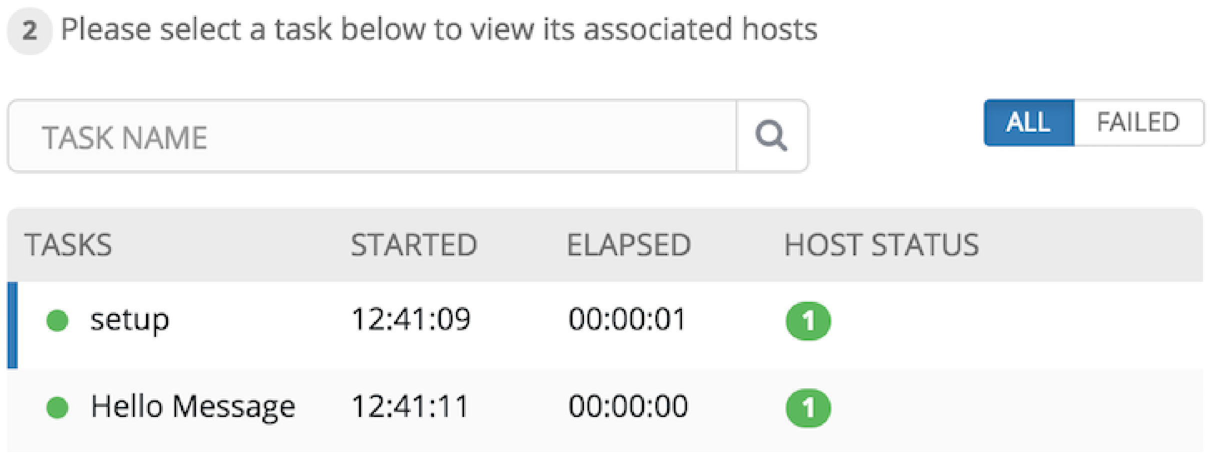

1
Please select from a play below to view its associated tasks.
PLAY NAME
ALL
FAILED

PLAYS	STARTED	ELAPSED
● Hello World Sample	12:41:09	00:00:02

6.3.2 Tasks

The **Tasks** area shows the tasks run as part of plays in the playbook. The displayed tasks can be filtered by **Task Name**, and can be limited to only failed tasks.

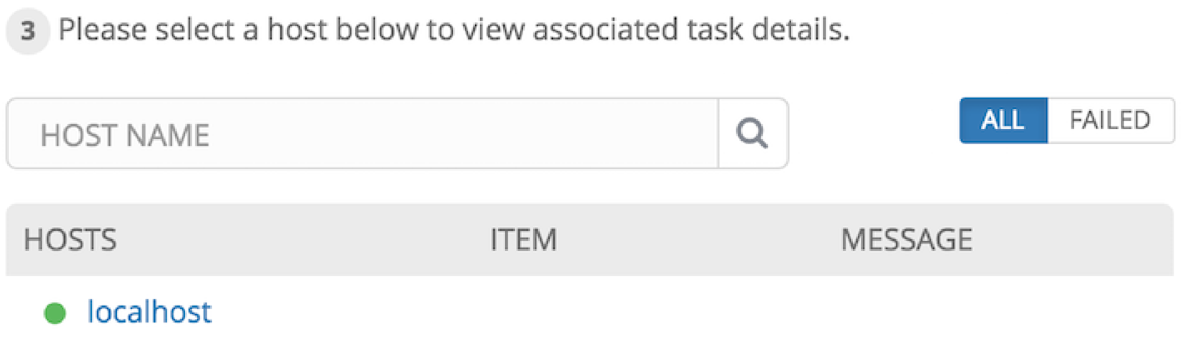
For each task, Tower shows the task **Name**, the start time for the task, the elapsed time of the task, whether the task succeeded or failed. Clicking on a specific task filters the **Host Events** area to only display hosts relative to that task.



TASKS	STARTED	ELAPSED	HOST STATUS
● setup	12:41:09	00:00:01	1
● Hello Message	12:41:11	00:00:00	1

6.3.3 Host Events

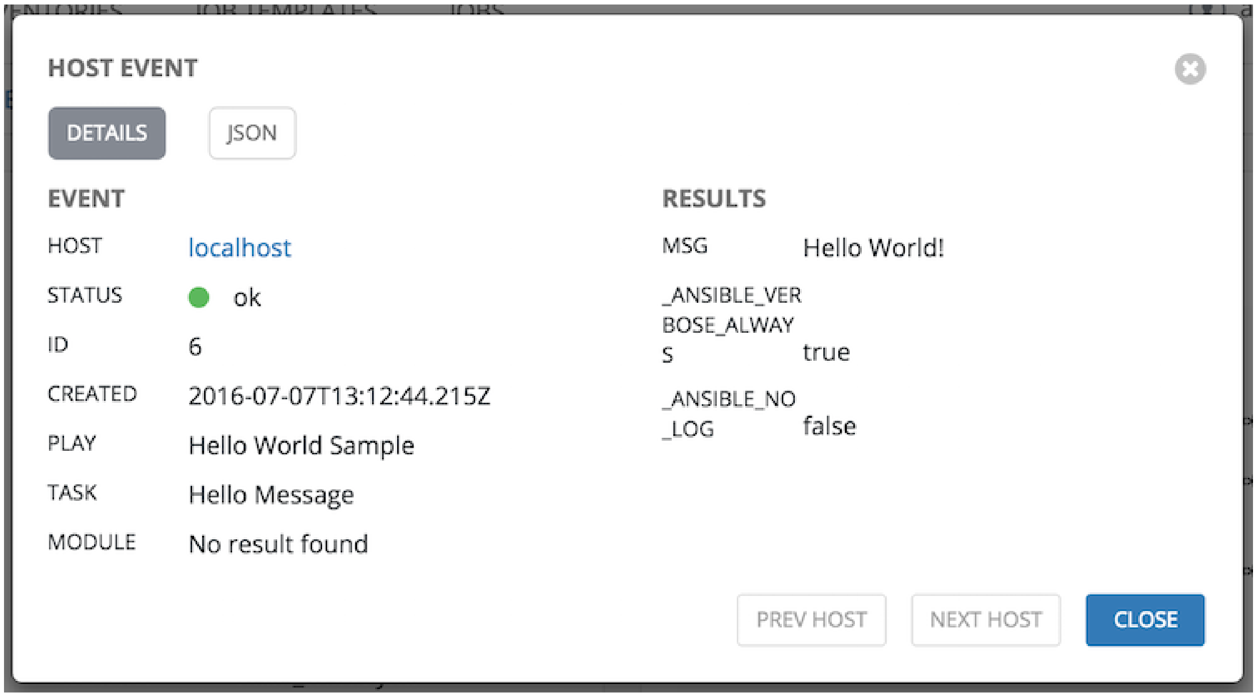
The **Host Events** area shows hosts affected by the selected play and task. For each host, Tower shows the host's status, its name, and any **Item** or **Message** set by that task.


3 Please select a host below to view associated task details.
HOST NAME
ALL
FAILED
HOSTS
ITEM
MESSAGE
localhost

Clicking on the linked hostname brings up the **Host Event** dialog for that host and task.

The **Host Event** dialog shows the events for this host and the selected play and task.

There is also a **JSON** tab which displays the result in JSON format.



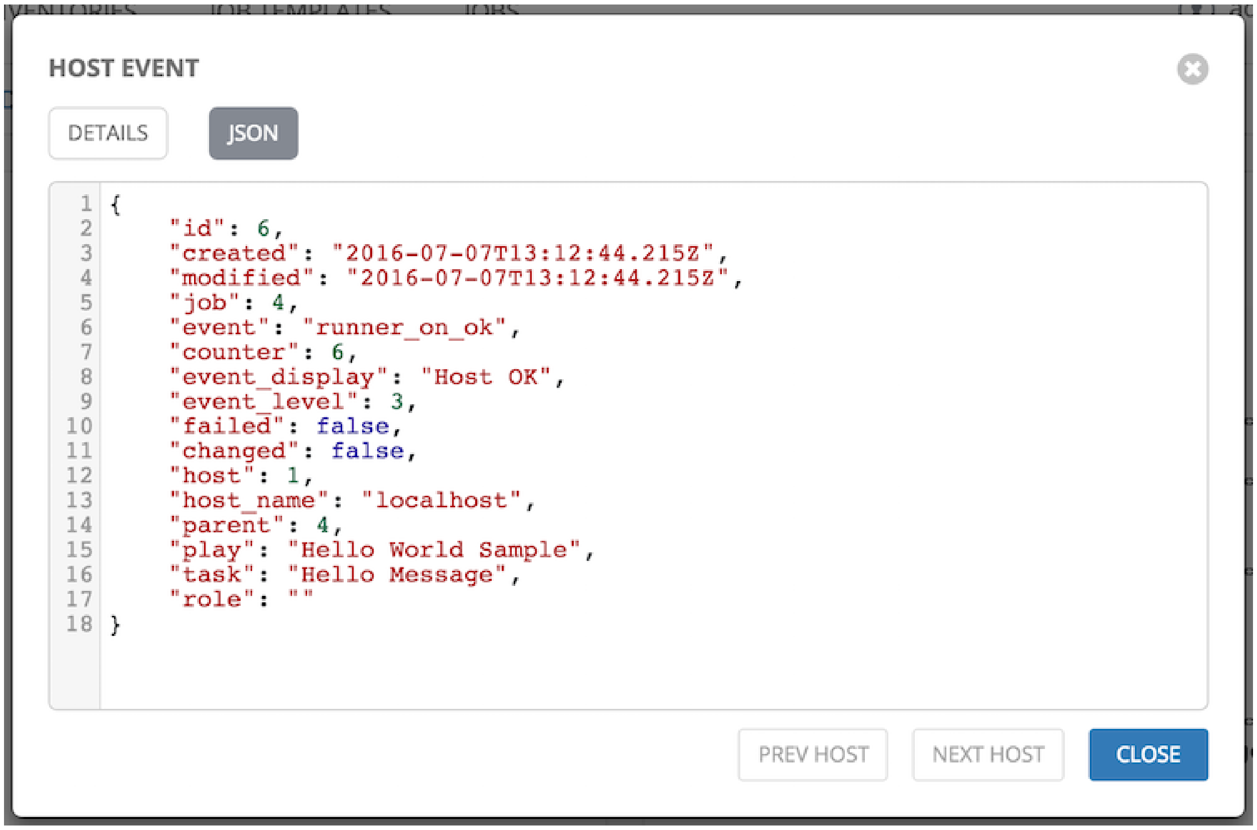
EVENT

HOST	localhost
STATUS	● ok
ID	6
CREATED	2016-07-07T13:12:44.215Z
PLAY	Hello World Sample
TASK	Hello Message
MODULE	No result found

RESULTS

MSG	Hello World!
_ANSIBLE_VERBOSE_ALWAYS	true
_ANSIBLE_NO_LOG	false

There is also a **JSON** tab which displays the result in JSON format.



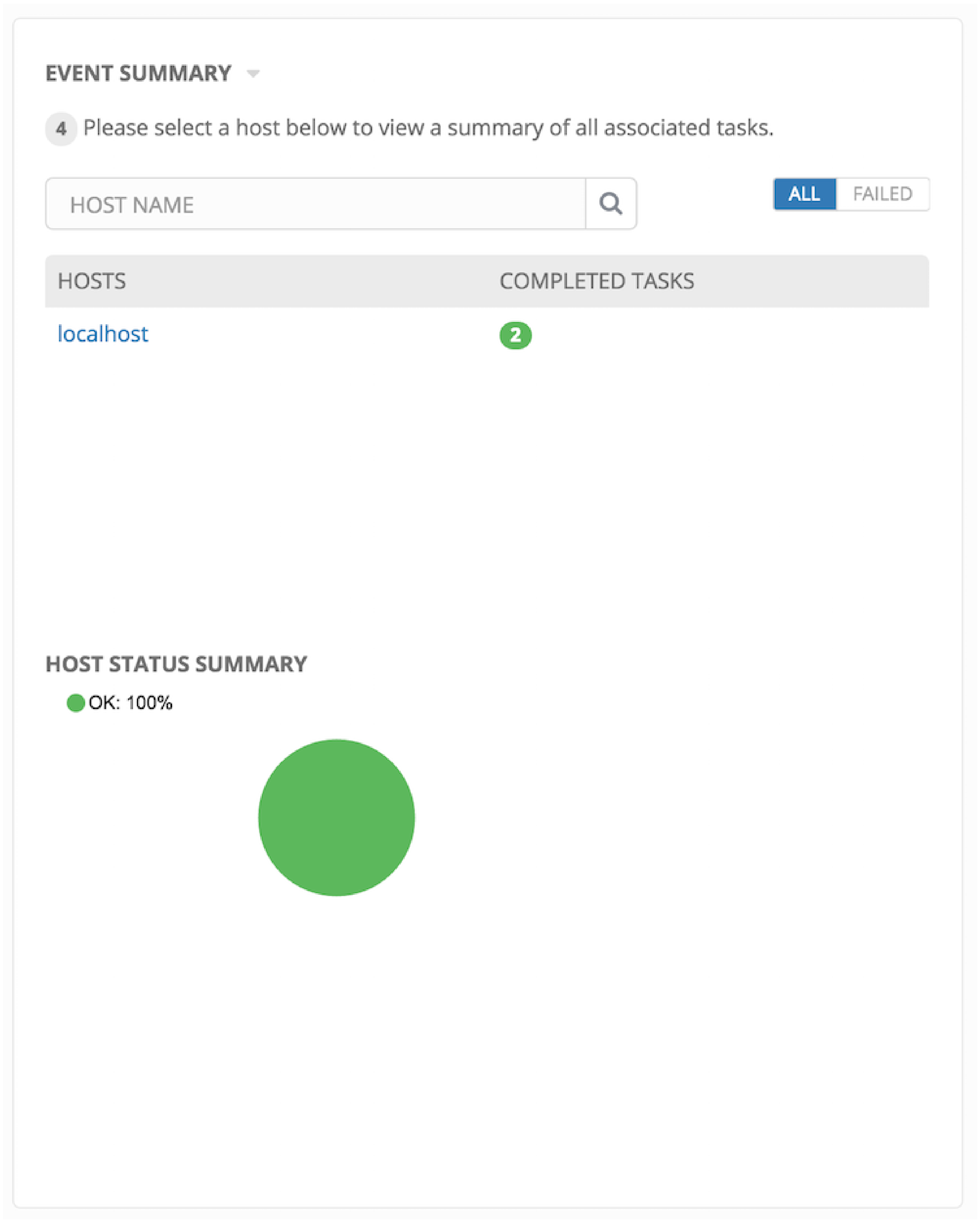
```
{
    "id": 6,
    "created": "2016-07-07T13:12:44.215Z",
    "modified": "2016-07-07T13:12:44.215Z",
    "job": 4,
    "event": "runner_on_ok",
    "counter": 6,
    "event_display": "Host OK",
    "event_level": 3,
    "failed": false,
    "changed": false,
    "host": 1,
    "host_name": "localhost",
    "parent": 4,
    "play": "Hello World Sample",
    "task": "Hello Message",
    "role": ""
}
```

6.4 Event Summary

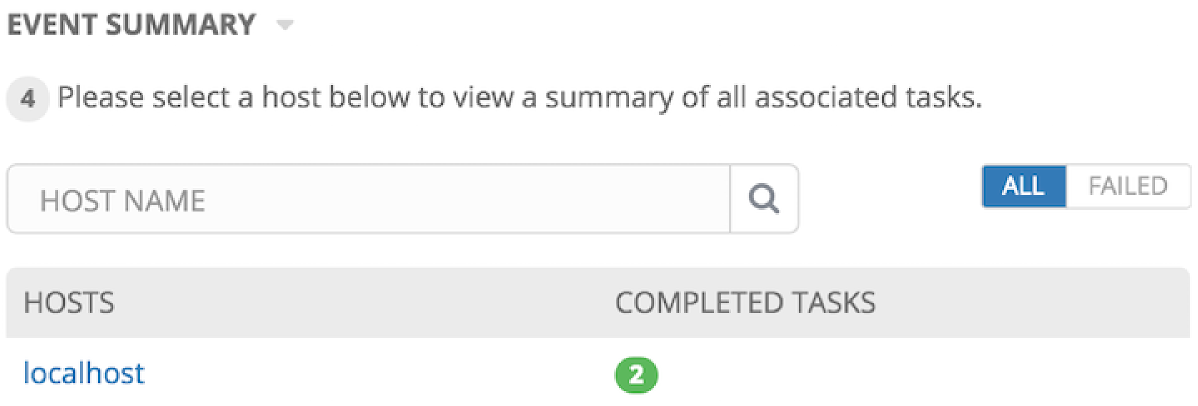
The **Event Summary** area shows a summary of events for all hosts affected by this playbook as well as the **Host Status Summary**.

By default, the **Event Summary** is collapsed and must be expanded before it can be viewed. Versions of Tower prior to 3.0 always displayed the **Event Summary**.

Hosts can be filtered by their hostname, and can be limited to showing only changed, failed, OK, and unreachable hosts.



For each host, the **Event Summary** area shows the hostname and the number of completed tasks for that host, sorted by status.


EVENT SUMMARY

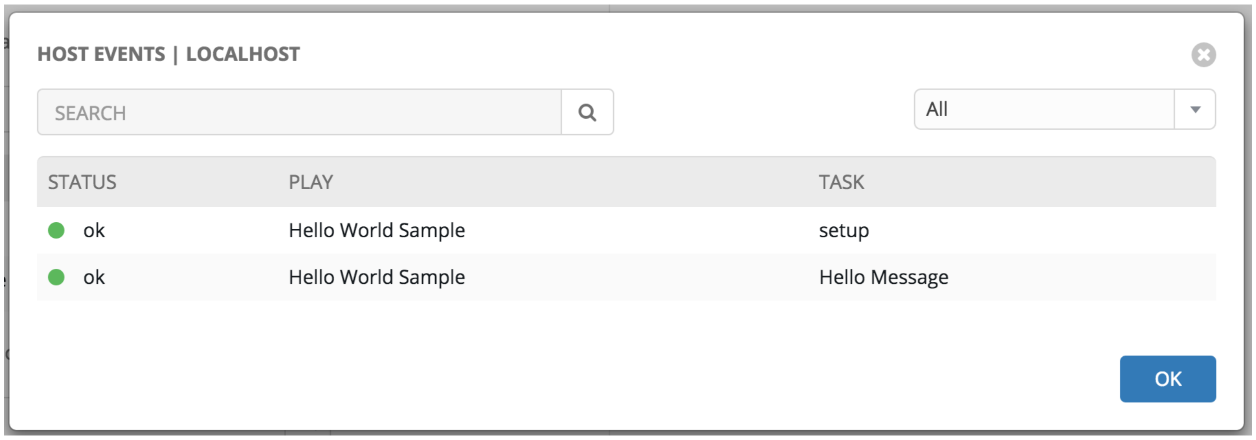
4 Please select a host below to view a summary of all associated tasks.

HOSTS	COMPLETED TASKS
localhost	2

Clicking on the hostname brings up a **Host Events** dialog, displaying all tasks that affected that host.

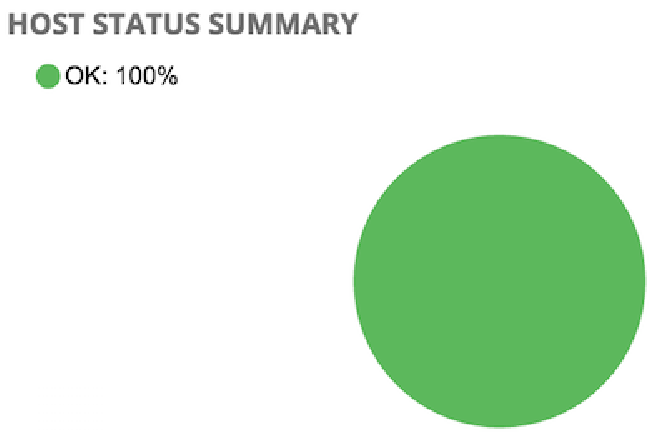
This dialog can be filtered by the status of the tasks, as well as by the hostname.

For each event, Tower displays the status, the play name, and the task name.


HOST EVENTS | LOCALHOST

STATUS	PLAY	TASK
● ok	Hello World Sample	setup
● ok	Hello World Sample	Hello Message

The **Host Summary** area shows a graph summarizing the status of all hosts affected by this playbook run.



CHAPTER

SEVEN

USING VIRTUALENV WITH ANSIBLE TOWER

Ansible Tower 3.0 uses *virtualenv*. Virtualenv creates isolated Python environments to avoid problems caused by conflicting dependencies and differing versions. Virtualenv works by simply creating a folder which contains all of the necessary executables and dependencies for a specific version of Python. Ansible Tower creates two virtualenvs during installation–one is used to run Tower, while the other is used to run Ansible. This allows Tower to run in a stable environment, while allowing you to add or update modules to your Ansible Python environment as necessary to run your playbooks.

Note: For more information on virtualenv, see Virtual Environments

7.1 Modifying the virtualenv

Modifying the virtualenv used by Tower is unsupported and not recommended. Instead, you can add modules to the virtualenv that Tower uses to run Ansible.

To do so, activate the Ansible virtualenv:

```
. /var/lib/awx/venv/ansible/bin/activate
```

...and then install whatever you need using `pip`:

```
pip install mypackagename
```

7.2 Tower virtualenv under Red Hat Enterprise Linux 6 and derivatives

Red Hat Enterprise Linux 6 and derivatives (including CentOS 6 and Oracle Linux 6) use Python version 2.6. However, Tower 3.0 requires several components (most notably Django 1.8) that require Python 2.7, so the Python 2.7 software collection is installed with Ansible Tower on these platforms. Note that this is only used to run Tower, not Ansible. Ansible still runs under the system Python 2.6. As a result, the two virtualenvs under `/var/lib/awx/venv` will each target a different Python interpreter.

CHAPTER EIGHT

INDEX

- genindex

CHAPTER

NINE

COPYRIGHT © 2016 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.

OpenStack™ and OpenStack logo are trademarks of OpenStack, LLC.

Chrome™ and Google Compute Engine™ 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

A

Ansible, executing in a virtual enviroment, 27

I

installation script
    inventory file setup, 8
    playbook setup, 9
inventory file setup, 8

J

job output, sdout, 17
jobs
    event summary, 24
    host events, 22
    host summary, 24
    job details, 21
    plays, 22
    tasks, 22

M

migration, 11
migration considerations, 11
MongoDB data removal, 11

P

permissions, 12, 13
playbook setup, 9
    setup.sh, 9
PostgreSQL data migration, 11

R

RBAC, 12, 13
release notes, v3.0, 2
roles, 12

S

singleton roles, 12
system tracking data, 11
system-wide roles, 12

U

upgrade, 7
upgrade considerations, 7

V

virtual enviroment, 27