How to Integrate Red Hat Ansible Tower with CA Continuous Delivery Director

The Red Hat Ansible Tower plug-in makes it simple to launch an Ansible job from CA Continuous Delivery Director (CDD) based on a job template. You can also use this plug-in to manage Ansible job template parameters within the CDD pipeline.

Contents

What is Ansible?

Ansible is an IT automation tool. It can configure systems, deploy software, and orchestrate more advanced IT tasks such as continuous deployments or zero downtime rolling updates.

The main components of Ansible are playbooks, written in YAML format. Playbooks can be used to manage configurations of and deployments to remote machines. Playbooks are stored in the project base path for each project.

For more information, see: http://docs.ansible.com/ansible/latest/playbooks.html

What is Ansible Tower?

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

What is a Job Template?

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 facilitate the reuse of Ansible playbook content and collaboration between teams.

How Does the Ansible Tower Plug-in Work?

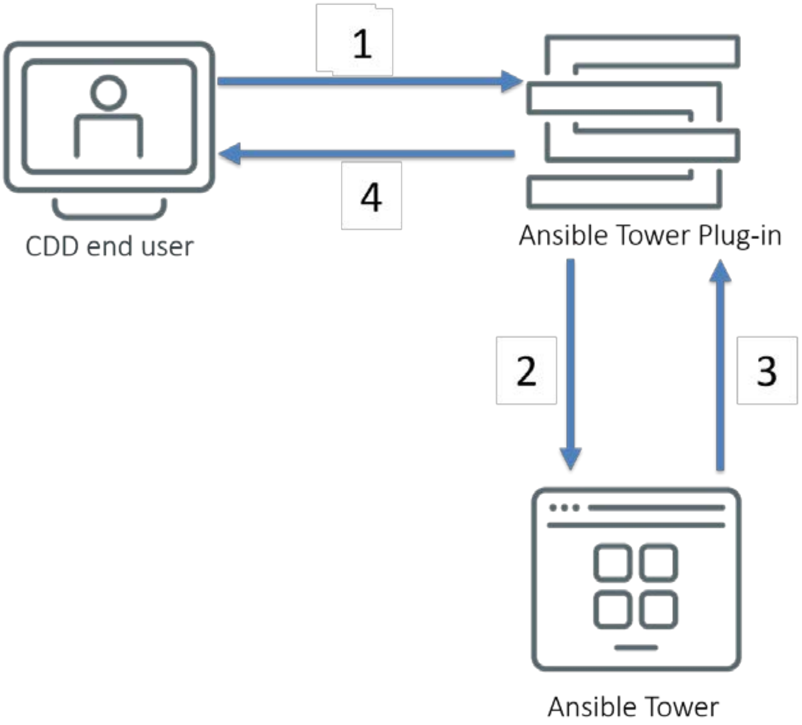


Figure 1 CDD – Ansible Tower Integration

1. A CDD user configures a task using the plug-in, specifying the job template and the template parameters in JSON format.

2. The plug-in validates the data and sends an HTTP request to Ansible Tower. (Default: via SSH protocol).

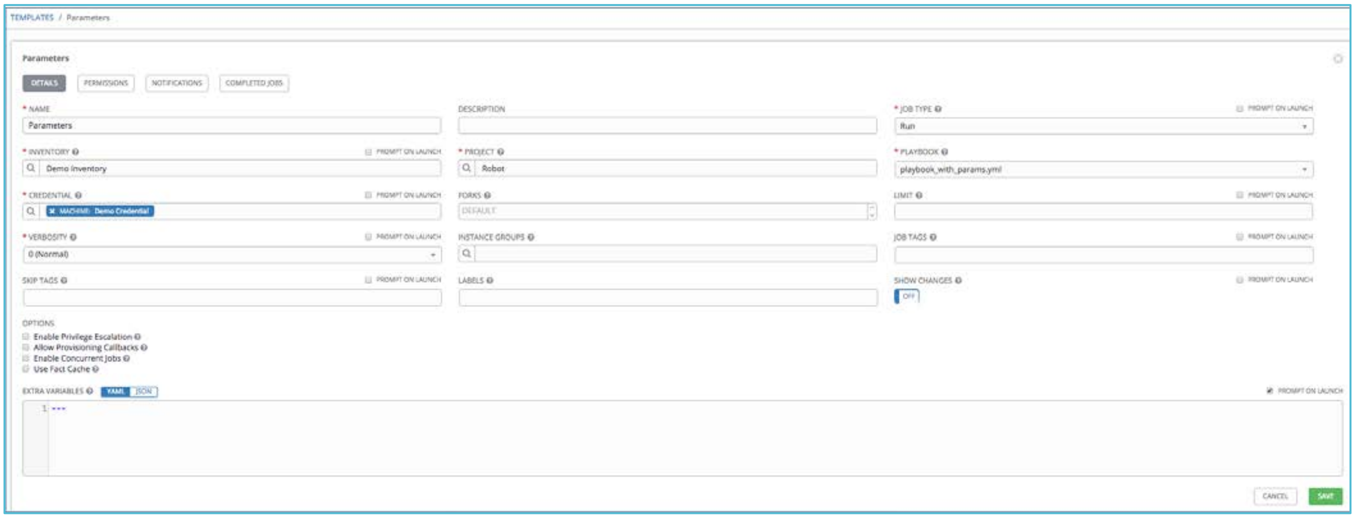
3. Ansible Tower accepts the request and runs an Ansible job, depending on the path of request, template, parameters and type of HTTP method.

4. The plug-in accepts the request from Ansible Tower, handles it, and returns the result to CDD.

For more information, see: https://docops.ca.com/ca-continuous-delivery-director/integrations/en/plug-ins/red-hat-ansible-tower-plug-in

Example 1: Running Ansible Tower Job with Parameters

A job template is configured in Ansible Tower named Parameters:



TEMPLATES / Parameters

Parameters

DETAILS | PERMISSIONS | NOTIFICATIONS | COMPLETED JOBS

* NAME: Parameters	DESCRIPTION:	* JOB TYPE (□ PROMPT ON LAUNCH): Run
* INVENTORY (□ PROMPT ON LAUNCH): Demo Inventory	* PROJECT: Robot	* PLAYBOOK: playbook_with_params.yml
* CREDENTIAL (□ PROMPT ON LAUNCH): MACHINE: Demo Credential	FORKS: DEFAULT	LIMIT (□ PROMPT ON LAUNCH):
* VERBOSITY (□ PROMPT ON LAUNCH): 0 (Normal)	INSTANCE GROUPS:	JOB TAGS (□ PROMPT ON LAUNCH):
SKIP TAGS (□ PROMPT ON LAUNCH):	LABELS:	SHOW CHANGES (□ PROMPT ON LAUNCH): OFF

OPTIONS

- □ Enable Privilege Escalation
- □ Allow Provisioning Callbacks
- □ Enable Concurrent Jobs
- □ Use Fact Cache

EXTRA VARIABLES YAML JSON ☑ PROMPT ON LAUNCH

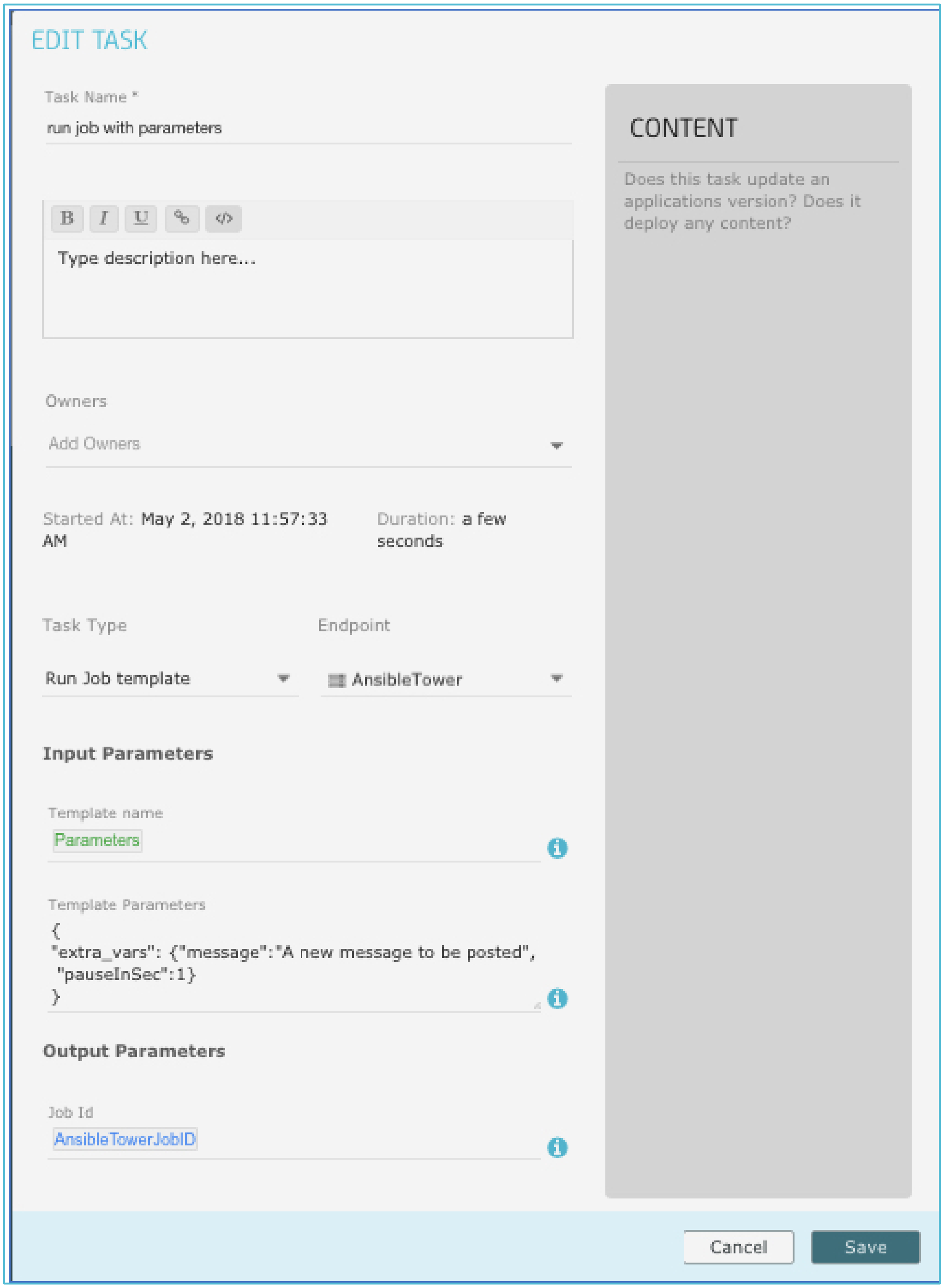
```
---
```

CANCEL SAVE

Note: When the Ansible Tower job template was configured, the **PROMPT ON LAUNCH** option was configured for **Extra Variables**: ☑ PROMPT ON LAUNCH

A task is configured in CDD with the following template parameters:

- Message: “message to be posted”
- Pause in seconds: ‘‘1”



EDIT TASK

Task Name *
run job with parameters

Type description here...

Owners
Add Owners

Started At: May 2, 2018 11:57:33 AM

Duration: a few seconds

Task Type: Run Job template

Endpoint: AnsibleTower

Input Parameters

Template name: Parameters

Template Parameters:

```
{
"extra_vars": {"message":"A new message to be posted",
 "pauseInSec":1}
}
```

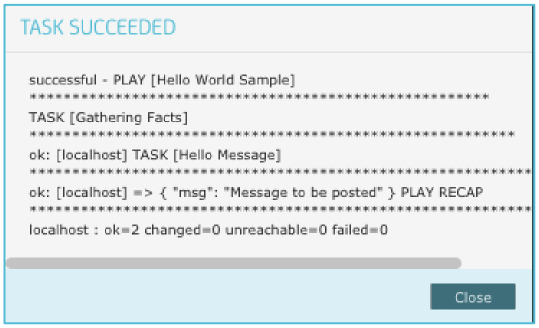
Output Parameters

Job Id: AnsibleTowerJobID

CONTENT

Does this task update an applications version? Does it deploy any content?

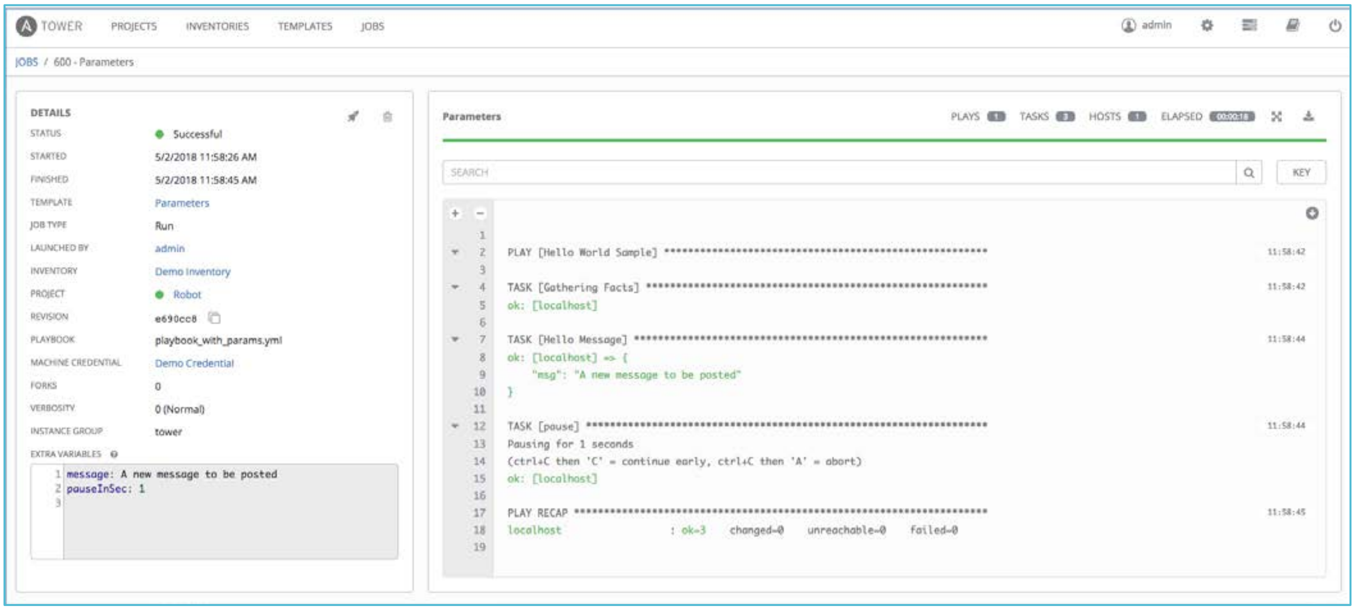
The task runs successfully, and the parameters and job tags are passed to Ansible Tower as indicated in the following success message:



TASK SUCCEEDED

```
successful - PLAY [Hello World Sample]
**************************************************************
TASK [Gathering Facts]
******************************************************************
ok: [localhost] TASK [Hello Message]
*********************************************************************
ok: [localhost] => { "msg": "Message to be posted" } PLAY RECAP
*********************************************************************
localhost : ok=2 changed=0 unreachable=0 failed=0
```

In Ansible Tower, the successful job is displayed.



DETAILS

STATUS	● Successful
STARTED	5/2/2018 11:58:26 AM
FINISHED	5/2/2018 11:58:45 AM
TEMPLATE	Parameters
JOB TYPE	Run
LAUNCHED BY	admin
INVENTORY	Demo Inventory
PROJECT	● Robot
REVISION	e690cc8
PLAYBOOK	playbook_with_params.yml
MACHINE CREDENTIAL	Demo Credential
FORKS	0
VERBOSITY	0 (Normal)
INSTANCE GROUP	tower

EXTRA VARIABLES

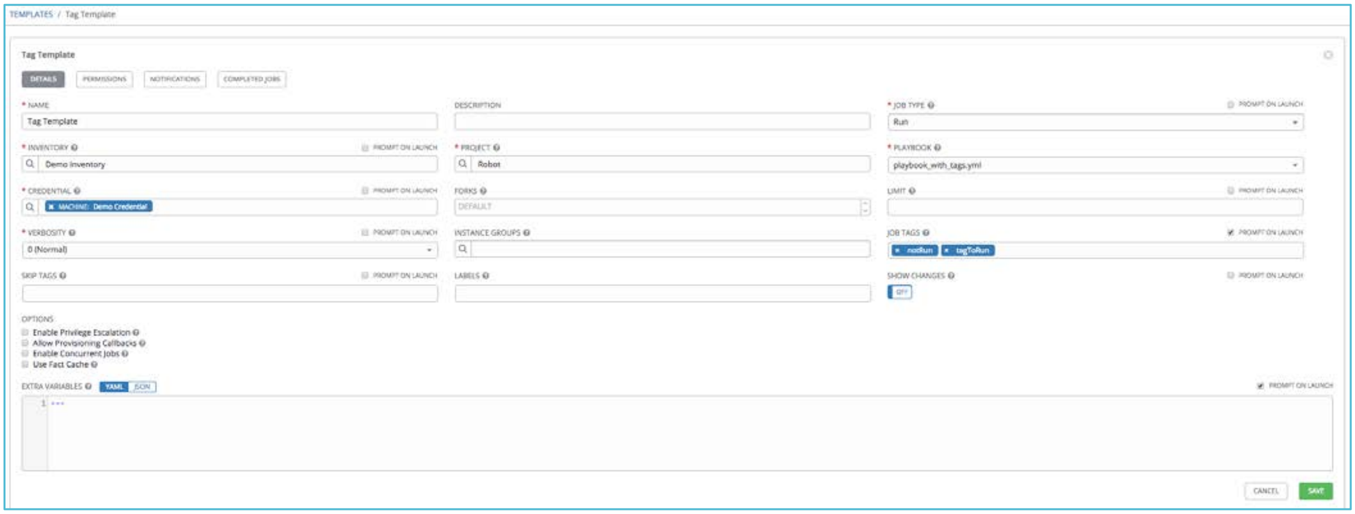
```
message: A new message to be posted
pauseInSec: 1
```

Parameters

```

PLAY [Hello World Sample] ****************************************************    11:58:42

TASK [Gathering Facts] *******************************************************    11:58:42
ok: [localhost]

TASK [Hello Message] *********************************************************    11:58:44
ok: [localhost] => {
    "msg": "A new message to be posted"
}

TASK [pause] *****************************************************************    11:58:44
Pausing for 1 seconds
(ctrl+C then 'C' = continue early, ctrl+C then 'A' = abort)
ok: [localhost]

PLAY RECAP *******************************************************************    11:58:45
localhost                  : ok=3    changed=0    unreachable=0    failed=0

```

Example 2: Running an Ansible Tower Job with Parameters that are used under a specific Tag

A job template is configured in Ansible Tower named Tag Template:



TEMPLATES / Tag Template

Tag Template

* NAME: Tag Template

DESCRIPTION:

* JOB TYPE: Run — □ PROMPT ON LAUNCH

* INVENTORY: Demo Inventory — □ PROMPT ON LAUNCH

* PROJECT: Robot

* PLAYBOOK: playbook_with_tags.yml

* CREDENTIAL: MACHINE: Demo Credential — □ PROMPT ON LAUNCH

FORKS: DEFAULT

LIMIT: — □ PROMPT ON LAUNCH

* VERBOSITY: 0 (Normal) — □ PROMPT ON LAUNCH

INSTANCE GROUPS:

JOB TAGS: notRun, tagToRun — ☑ PROMPT ON LAUNCH

SKIP TAGS: — □ PROMPT ON LAUNCH

LABELS:

SHOW CHANGES: OFF — □ PROMPT ON LAUNCH

OPTIONS

□ Enable Privilege Escalation
□ Allow Provisioning Callbacks
□ Enable Concurrent Jobs
□ Use Fact Cache

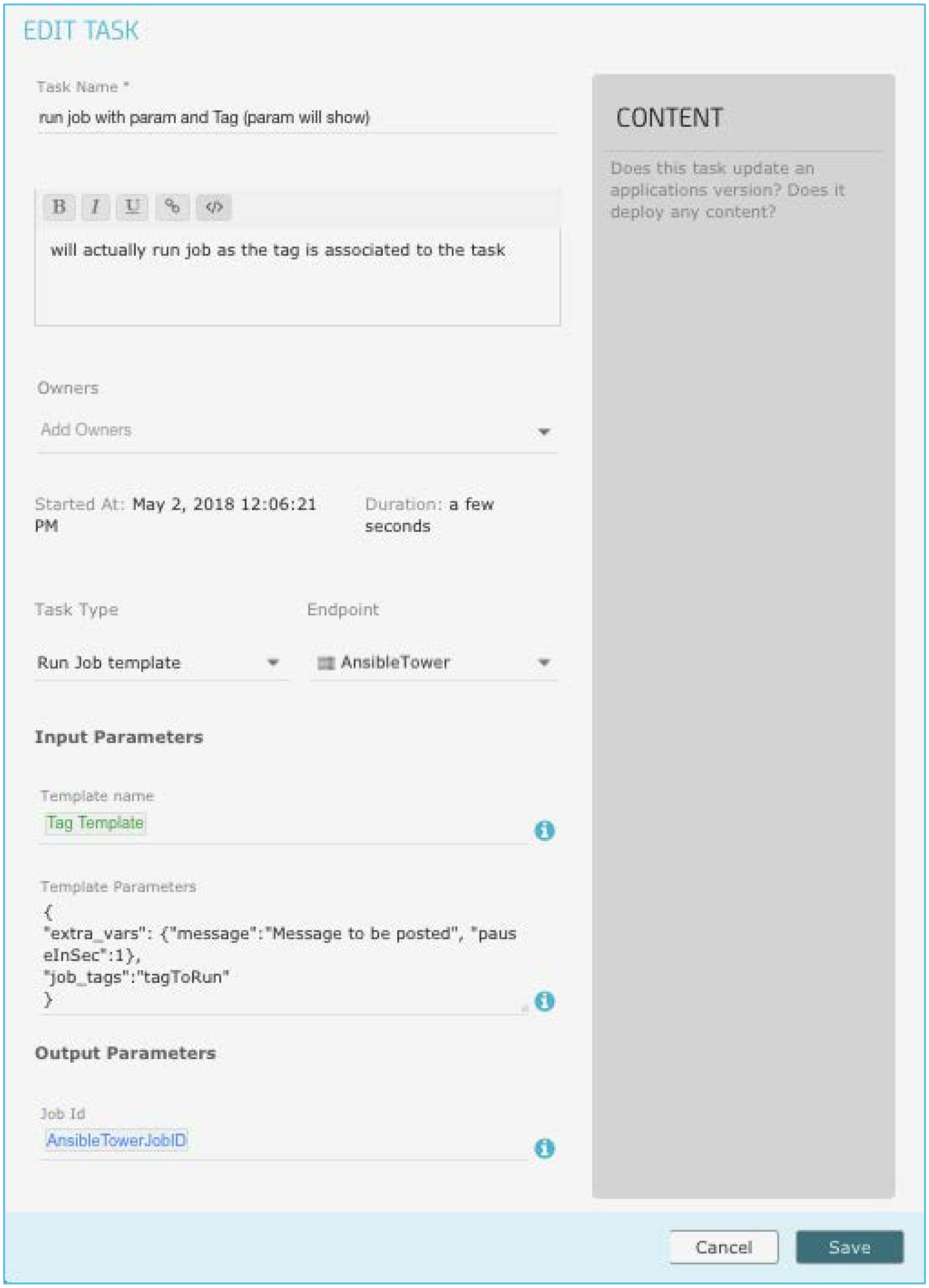
EXTRA VARIABLES: YAML JSON — ☑ PROMPT ON LAUNCH

```
---
```

Note: When the Ansible Tower job template was configured, the **PROMPT ON LAUNCH** option was configured for **Extra Variables** and **Job Tags**: ☑ PROMPT ON LAUNCH

A task is configured in CDD with the following template parameters:

- Message: “message to be posted”
- Pause in seconds: ‘‘1”
- Job tags: “tagToRun”



EDIT TASK

Task Name *

run job with param and Tag (param will show)

will actually run job as the tag is associated to the task

Owners

Add Owners

Started At: May 2, 2018 12:06:21 PM

Duration: a few seconds

Task Type

Run Job template

Endpoint

AnsibleTower

Input Parameters

Template name

Tag Template

Template Parameters

```
{
"extra_vars": {"message":"Message to be posted", "pauseInSec":1},
"job_tags":"tagToRun"
}
```

Output Parameters

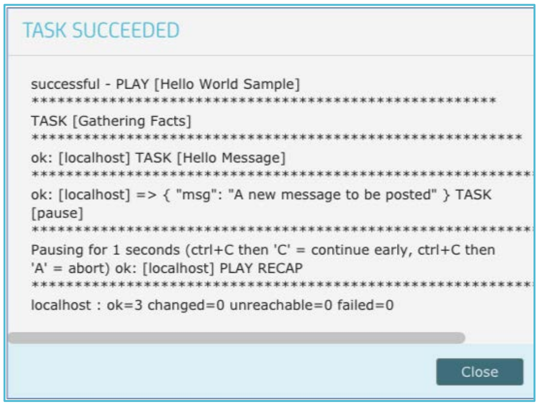
Job Id

AnsibleTowerJobID

CONTENT

Does this task update an applications version? Does it deploy any content?

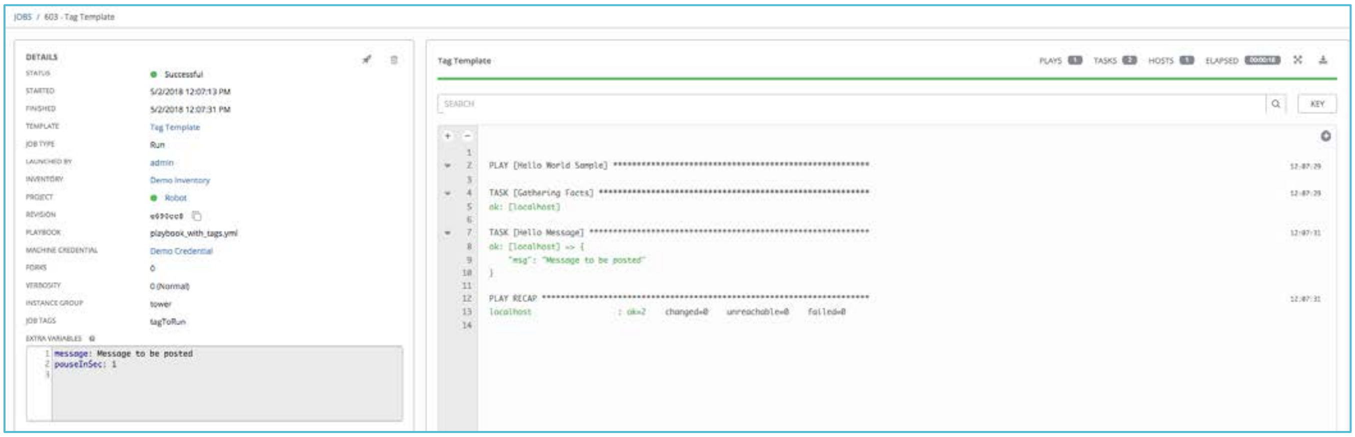
The task runs successfully, and the parameters and job tags are passed to Ansible Tower as indicated in the following success message:



TASK SUCCEEDED

```
successful - PLAY [Hello World Sample]
*******************************************************
TASK [Gathering Facts]
**********************************************************
ok: [localhost] TASK [Hello Message]
***********************************************************
ok: [localhost] => { "msg": "A new message to be posted" } TASK
[pause]
***********************************************************
Pausing for 1 seconds (ctrl+C then 'C' = continue early, ctrl+C then
'A' = abort) ok: [localhost] PLAY RECAP
***********************************************************
localhost : ok=3 changed=0 unreachable=0 failed=0
```

In Ansible Tower, the successful job is displayed.



DETAILS

STATUS	● Successful
STARTED	5/2/2018 12:07:13 PM
FINISHED	5/2/2018 12:07:31 PM
TEMPLATE	Tag Template
JOB TYPE	Run
LAUNCHED BY	admin
INVENTORY	Demo Inventory
PROJECT	● Robot
REVISION	e690cc8
PLAYBOOK	playbook_with_tags.yml
MACHINE CREDENTIAL	Demo Credential
FORKS	0
VERBOSITY	0 (Normal)
INSTANCE GROUP	tower
JOB TAGS	tagToRun

EXTRA VARIABLES

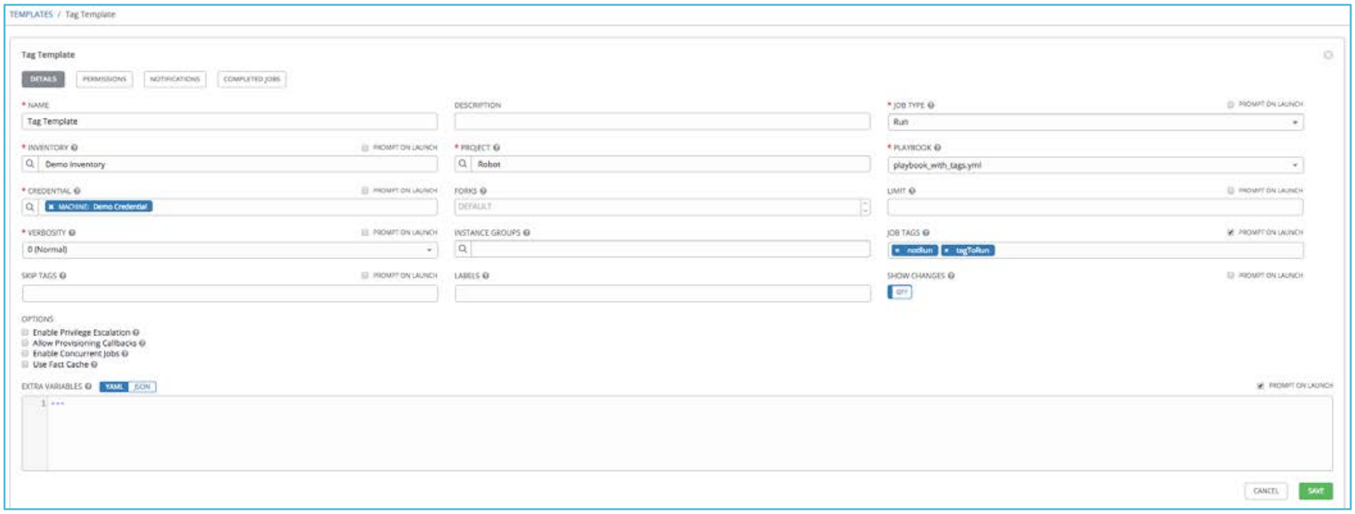
```
message: Message to be posted
pauseInSec: 1
```

Tag Template

```

PLAY [Hello World Sample] ******************************************************     12:07:29

TASK [Gathering Facts] *********************************************************     12:07:29
ok: [localhost]

TASK [Hello Message] ***********************************************************     12:07:31
ok: [localhost] => {
    "msg": "Message to be posted"
}

PLAY RECAP *********************************************************************     12:07:31
localhost                  : ok=2    changed=0    unreachable=0    failed=0

```

Example 3: Running an Ansible Tower Job with Parameters that are NOT used under a specific Tag

A job template is configured in Ansible Tower named Tag Template:



Tag Template

* NAME: Tag Template	DESCRIPTION:	* JOB TYPE (□ PROMPT ON LAUNCH): Run
* INVENTORY (□ PROMPT ON LAUNCH): Demo Inventory	* PROJECT: Robot	* PLAYBOOK: playbook_with_tags.yml
* CREDENTIAL (□ PROMPT ON LAUNCH): MACHINE: Demo Credential	FORKS: DEFAULT	LIMIT (□ PROMPT ON LAUNCH):
* VERBOSITY (□ PROMPT ON LAUNCH): 0 (Normal)	INSTANCE GROUPS:	JOB TAGS (☑ PROMPT ON LAUNCH): notRun, tagToRun
SKIP TAGS (□ PROMPT ON LAUNCH):	LABELS:	SHOW CHANGES (□ PROMPT ON LAUNCH): OFF

OPTIONS
- □ Enable Privilege Escalation
- □ Allow Provisioning Callbacks
- □ Enable Concurrent Jobs
- □ Use Fact Cache

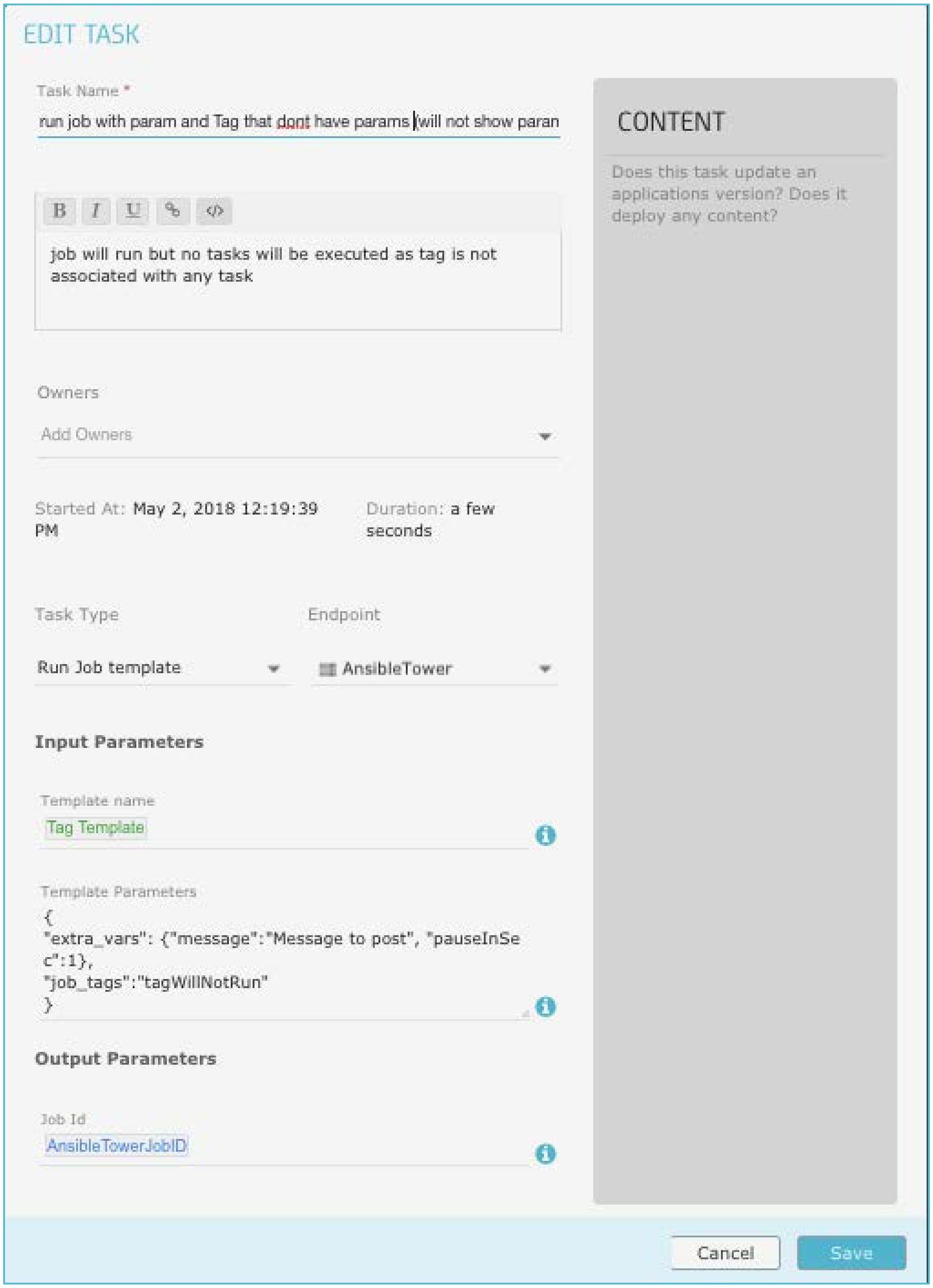
EXTRA VARIABLES [✓ YAML] JSON ☑ PROMPT ON LAUNCH

```
---
```

Note: When the Ansible Tower job template was configured, the **PROMPT ON LAUNCH** option was configured for **Extra Variables** and **Job Tags**: ☑ PROMPT ON LAUNCH

A task is configured in CDD with the following template parameters:

- Message: “message to be posted”
- Pause in seconds: ‘‘1”
- Job tags: “tagWillNotRun”



Task Name *

run job with param and Tag that dont have params will not show paran

job will run but no tasks will be executed as tag is not associated with any task

Owners

Started At: May 2, 2018 12:19:39 PM

Duration: a few seconds

Task Type: Run Job template

Endpoint: AnsibleTower

Input Parameters

Template name: Tag Template

Template Parameters

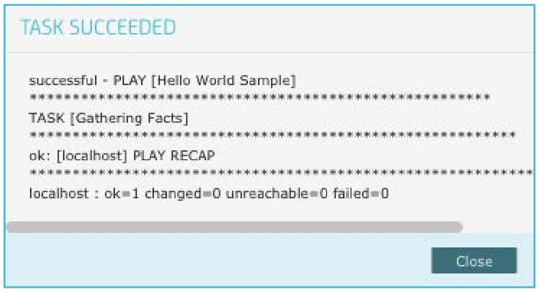
```
{
"extra_vars": {"message":"Message to post", "pauseInSe
c":1},
"job_tags":"tagWillNotRun"
}
```

Output Parameters

Job Id: AnsibleTowerJobID

CONTENT
Does this task update an applications version? Does it deploy any content?
Cancel
Save

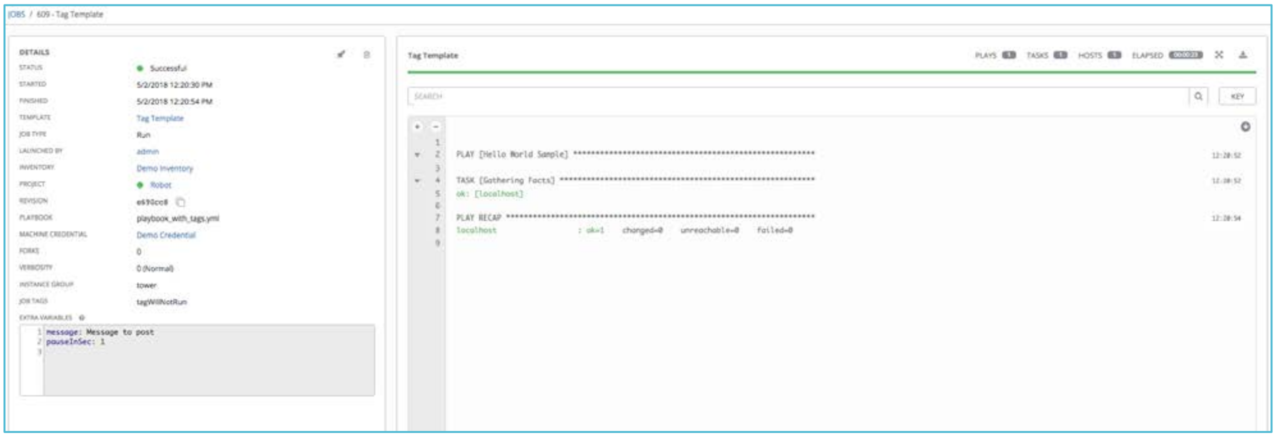
The task runs successfully, and the parameters and job tags are passed to Ansible Tower as indicated in the following success message:



TASK SUCCEEDED

```
successful - PLAY [Hello World Sample]
*********************************************************
TASK [Gathering Facts]
*************************************************************
ok: [localhost] PLAY RECAP
**************************************************************
localhost : ok=1 changed=0 unreachable=0 failed=0
```

In Ansible Tower, the successful job is displayed.



DETAILS

STATUS	● Successful
STARTED	5/2/2018 12:20:30 PM
FINISHED	5/2/2018 12:20:54 PM
TEMPLATE	Tag Template
JOB TYPE	Run
LAUNCHED BY	admin
INVENTORY	Demo Inventory
PROJECT	● Robot
REVISION	e690cc8
PLAYBOOK	playbook_with_tags.yml
MACHINE CREDENTIAL	Demo Credential
FORKS	0
VERBOSITY	0 (Normal)
INSTANCE GROUP	tower
JOB TAGS	tagWillNotRun

EXTRA VARIABLES

```
message: Message to post
pauseInSec: 1
```

Tag Template

```

PLAY [Hello World Sample] *********************************************************   12:20:52

TASK [Gathering Facts] ************************************************************   12:20:52
ok: [localhost]

PLAY RECAP ************************************************************************   12:20:54
localhost                  : ok=1    changed=0    unreachable=0    failed=0

```