Ansible Workshop - Exercises

Automation Platform

Learn to manage and run your Ansible content in AAP.

6 - Workflows

Objective

The basic idea of a workflow is to link multiple Job Templates together. They may or may not share inventory, playbooks or even permissions. The links can be conditional:

- if job template A succeeds, job template B is automatically executed afterwards
- but in case of failure, job template C will be run.

And the workflows are not even limited to Job Templates, but can also include project or inventory updates.

This enables new applications for Ansible automation controller: different Job Templates can build upon each other. E.g. the networking team creates playbooks with their own content, in their own Git repository and even targeting their own inventory, while the operations team also has their own repos, playbooks and inventory.

In this lab you'll learn how to setup a workflow.

Guide

Lab scenario

You have two departments in your organization:

- The web operations team that is developing playbooks in their own Git branch named `webops`
- The web developers team that is developing playbooks in their own Git branch named `webdev`.

When there is a new Node.js server to deploy, two things need to happen:

Web operations team

- `httpd`, `firewalld`, and `node.js` need to be installed, `SELinux` settings configured, the firewall needs to be opened, and `httpd` and `node.js` should get started.

Web developers team

- The most recent version of the web application needs to be deployed and `node.js` needs to be restarted.

In other words, the Web operations team prepares a server for application deployment, and the Web developers team deploys the application on the server.

To make things somewhat easier for you, everything needed already exists in a Github repository: playbooks, JSP-files etc. You just need to glue it together.

Note

In this example we use two different branches of the same repository for the content of the separate teams. In reality, the structure of your Source Control repositories depends on a lot of factors and could be different.

Set up projects

First you have to set up the Git repo as a Project like you normally would.

Warning

If you are still logged in as user **wweb**, log out of and log in as user **admin**.

Within **Resources** -> **Projects**, click the **Add** button to create a project for the web operations team. Fill out the form as follows:

Parameter	Value
Name	`Webops Git Repo`
Organization	`Default`
Default Execution Environment	`Default Execution Environment`
Source Control Credential Type	`Git`
Source Control URL	`https://github.com/ansible/workshop-examples.git`
Source Control Branch/Tag/Commit	`webops`
Options	☑ Clean ☑ Delete ☑ Update Revision on Launch

Click **Save**

Within **Resources** -> **Projects**, click the **Add** button to create a project for the web developers team. Fill out the form as follows:

Parameter	Value
Name	`Webdev Git Repo`

Default Execution Environment	`Default Execution Environment`
Source Control Credential Type	`Git`
Source Control URL	`https://github.com/ansible/workshop-examples.git`
Source Control Branch/Tag/Commit	`webdev`
Options	☑ Clean ☑ Delete ☑ Update Revision on Launch

Click **Save**

Set up job templates

Now you have to create two Job Templates like you would for "normal" Jobs.

Within **Resources** -> **Templates**, click the **Add** button and choose **Add job template**:

Parameter	**Value**
Name	`Web App Deploy`
Job Type	`Run`
Inventory	`Workshop Inventory`
Project	`Webops Git Repo`
Execution Environment	`Default execution environment`
Playbook	`rhel/webops/web_infrastructure.yml`
Credentials	`Workshop Credential`
Limit	`web`
Options	☑ Privilege Escalation

Click **Save**

Within **Resources** -> **Templates**, click the **Add** button and choose **Add job template**:

Parameter	Value
Name	`Node.js Deploy`
Job Type	`Run`
Inventory	`Workshop Inventory`
Project	`Webdev Git Repo`
Execution Environment	`Default execution environment`
Playbook	`rhel/webdev/install_node_app.yml`
Credentials	`Workshop Credential`
Limit	`web`
Options	☑ Privilege Escalation

Click **Save**

> **Tip**
>
> If you want to know what the Ansible Playbooks look like, check out the Github URL and switch to the appropriate branches.

Set up the workflow

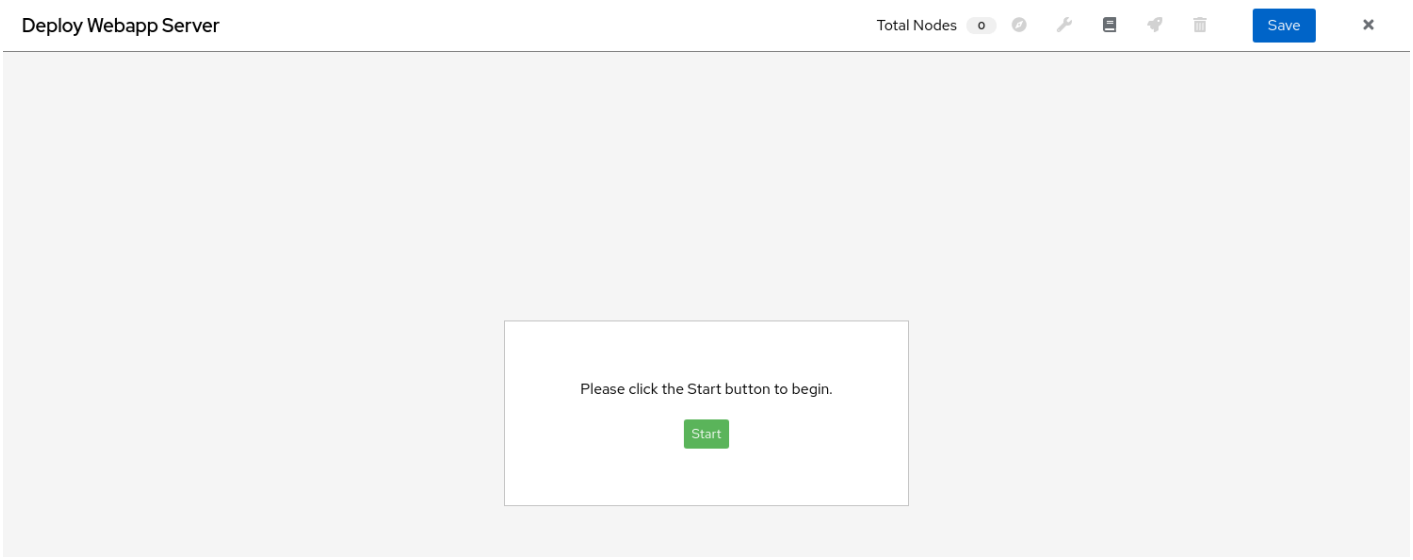
Workflows are configured in the **Templates** view, you might have noticed you can choose between **Add job template** and **Add workflow template** when adding a template.

Within **Resources** -> **Templates**, click the **Add** button and choose **Add workflow template**:

Parameter	Value
Name	`Deploy Webapp Server`
Organization	`Default`

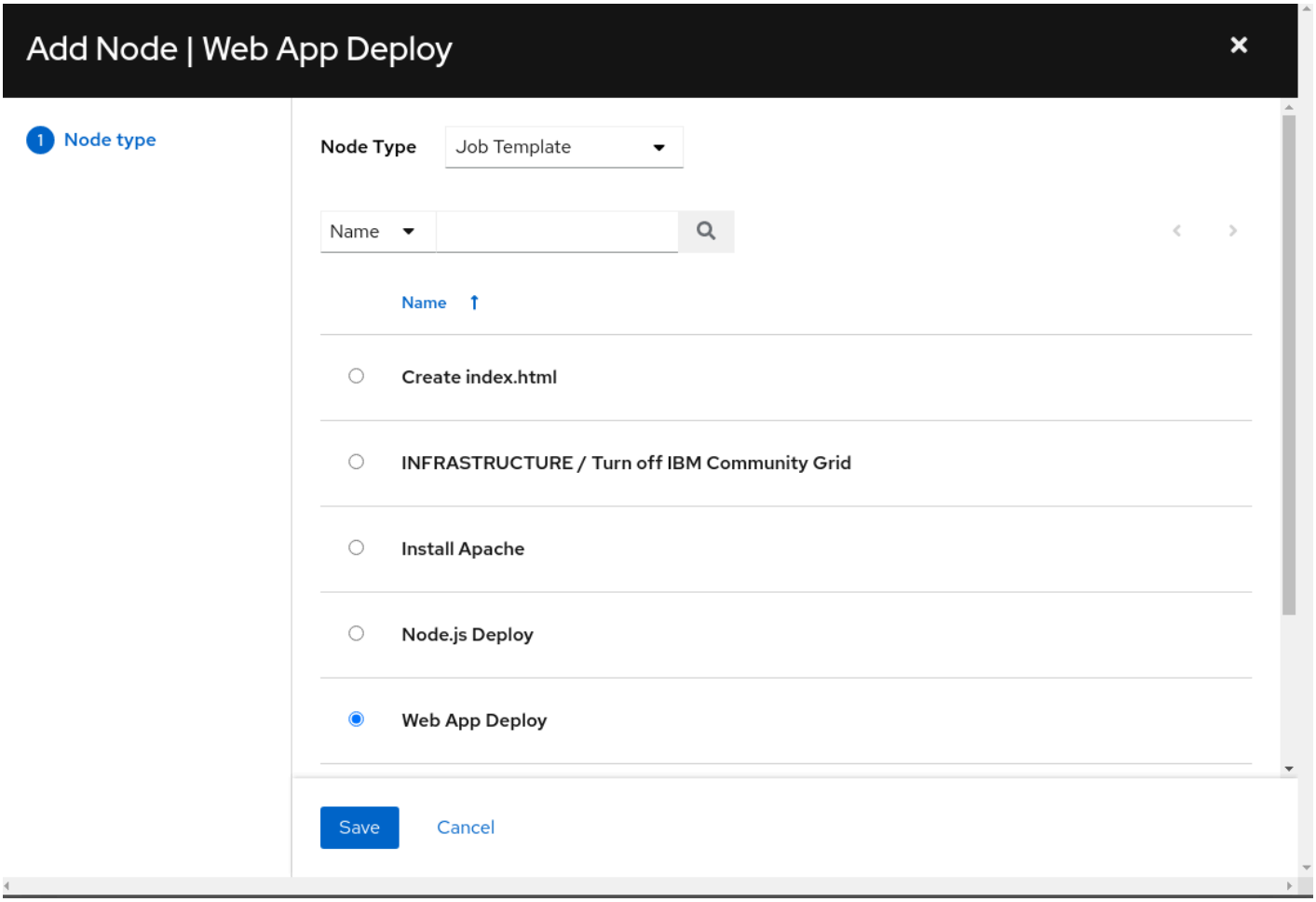
Click **Save**

After saving the template the **Workflow Visualizer** opens to allow you to build a workflow. You can later open the **Workflow Visualizer** again by using the button on the template details page and selecting **Visualizer** from the menu.

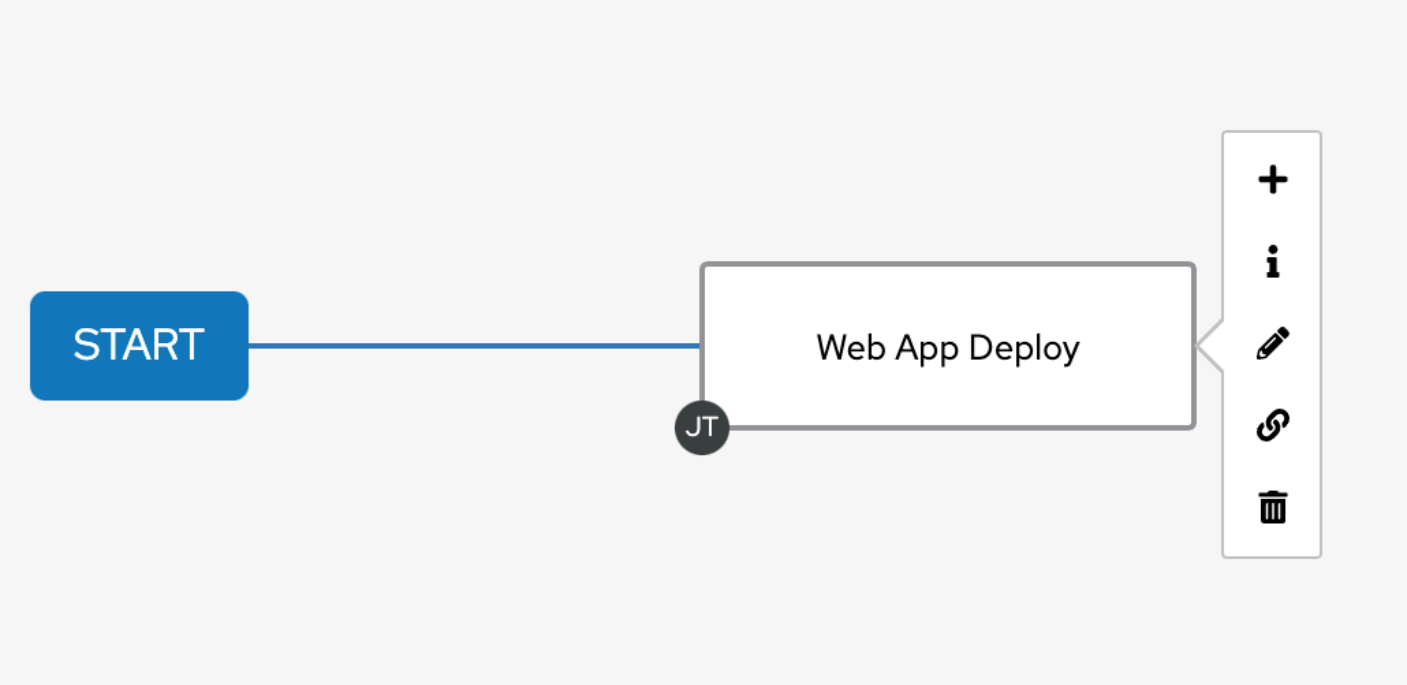


Click on the **Start** button, an **Add Node** window opens. Assign an action to the node, via node type by selecting **Job Template**.

Select the **Web App Deploy** job template and click **Save**.



A new node is shown, connected to the **START** button with the name of the job template. Hover the mouse pointer over the node, you'll see options to add a node (+), view node details (i), edit the node (pencil), link to an available node (chain), and delete the node (trash bin).

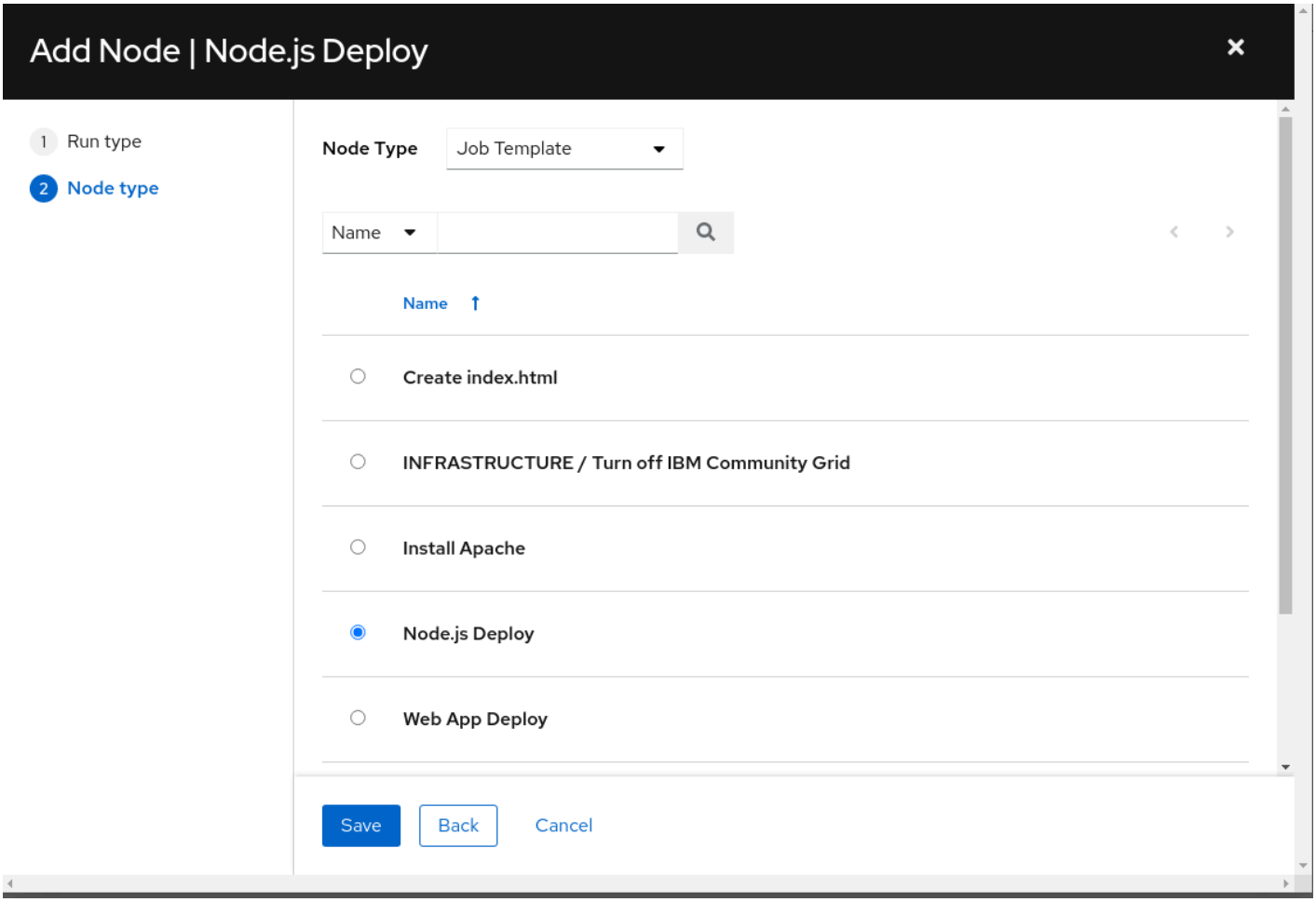


Hover over the node and click the (+) sign to add a new node.

- For the **Run Type** select **On Success** (default) and click **Next**.

> **Tip**
>
> The run type allows for more complex workflows. You could lay out different execution paths for successful and for failed playbook runs.

- For **Node Type** select **Job Template** (default) and choose the **Node.js Deploy** job template. Click **Save**.

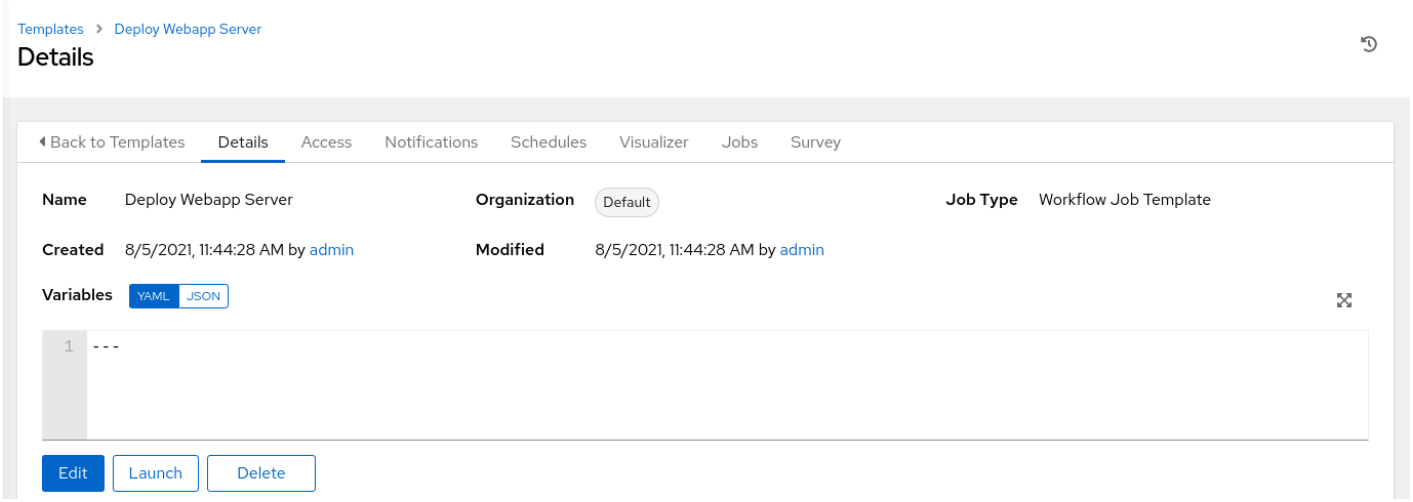


Click **Save** in the top right corner of the **Visualizier** view.

> **Tip**
>
> The **Visualizer** has options for setting up more advanced workflows, please refer to the documentation.

Launch workflow

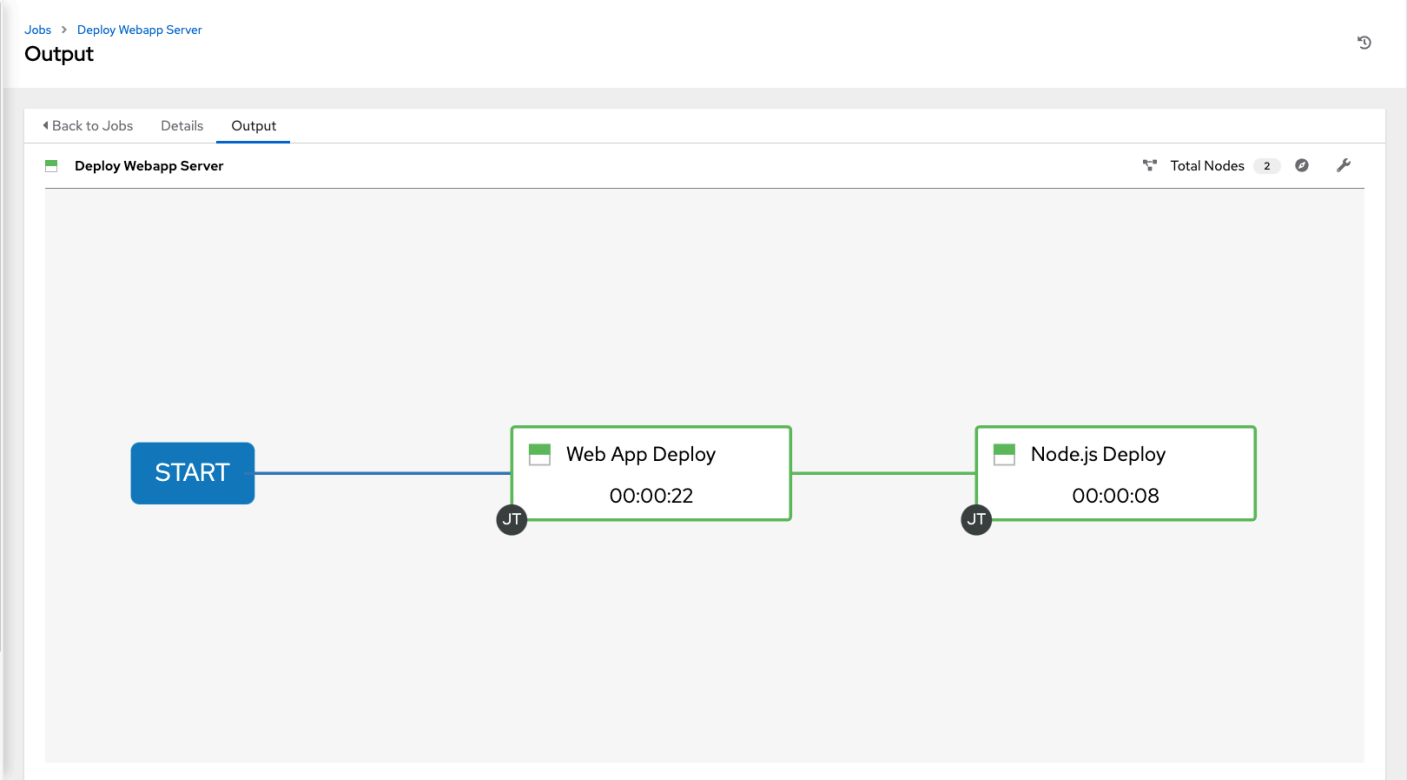
From within the **Deploy Webapp Server** Details page, **Launch** the workflow.



Note how the workflow run is shown in the Jobs > Deploy Webapp Server Output. In contrast to a normal job template job execution, there is no playbook output when the job completes but the time to complete the job is displayed. If you want to look at the actual playbook run, hover over the node you wish to see the details on and click it. Within the Details view of the job, select the **Output** menu to see the playbook output. If you want to get back the **Output** view of the **Deploy WebappServer** workflow, under Views -> Jobs -> **XX - Deploy Webapp Server** will take you back to the Output overview.

Note

`XX` is the number of the job run.



After the job was finished, check if everything worked fine: from your control host run the following curl command against `node1`, `node2` and `node3`. The output of each curl command should be `Hello World`.

```
[student@ansible-1 ansible-files]$ curl http://nodeX/nodejs
Hello World
```

Note

`X` should be replaced with the appropriate number of the node you are checking.

© Tim Grützmacher 2025