

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:

- **Web operations team:** developing playbooks in their Git branch `webops`.
- **Web developers team:** working in their branch `webdev`.

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

Web operations team

- Install `httpd`, `firewalld`, and `node.js`.
- Configure `SELinux` settings, open the firewall, and start `httpd` and `node.js`.

Web developers team

- Deploy the latest version of the web application and restart `node.js`.

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 **Automation Execution** → **Projects**, click **Create Project** to set up the web operations team's project: 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	<input checked="" type="checkbox"/> Clean <input checked="" type="checkbox"/> Delete <input checked="" type="checkbox"/> Update Revision on Launch

Click **Create project**.

Repeat the process to set up the **Webdev Git Repo**, using the branch `webdev`. Fill out the form as follows:

Parameter	Value
Name	Webdev 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	webdev
Options	<input checked="" type="checkbox"/> Clean <input checked="" type="checkbox"/> Delete <input checked="" type="checkbox"/> Update Revision on Launch

Click **Create project**.

Set up job templates

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

Within **Automation Execution** → **Templates** → **Create template** → **Create job template**, fill out the form with the following values:

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 Credentials
Limit	web
Options	<input checked="" type="checkbox"/> Privilege Escalation

Click **Create job template**.

The screenshot shows the 'Create job template' page in the Red Hat Ansible Automation Platform. The left sidebar is collapsed, and the main area is filled with configuration fields for a new job template. The 'Name' field is set to 'Web App Deploy'. The 'Job type' is selected as 'Run'. The 'Inventory' is set to 'Workshop Inventory'. The 'Project' is 'Webops Git Repo'. The 'Playbook' is 'rhel/webops/web_infrastructure.yml'. Other settings include 'Forks' (0), 'Limit' (0), 'Timeout' (0), and 'Labels' (Select or create labels). The 'Extra variables' section contains checkboxes for 'Privilege escalation' (checked), 'Concurrent jobs' (unchecked), 'Provisioning callback' (unchecked), 'Enable webhook' (unchecked), 'Enable fact storage' (unchecked), and 'Prevent instance group fallback' (unchecked). At the bottom, there are 'Create job template' and 'Cancel' buttons.

Click **Create job template**, and then repeat the process for the **Node.js Deploy** template, changing the project to **Webdev Git Repo** and the playbook to `rhel/webdev/install_node_app.yml`.

Parameter	Value
Name	Node.js Deploy
Job Type	Run
Inventory	Workshop Inventory
Project	Webdev Git Repo
Execution Environment	Default execution environment
Playbook	<code>rhel/webdev/install_node_app.yml</code>
Credentials	Workshop Credentials
Limit	web

Options

Privilege Escalation

Click **Create job template**.



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 **Create job template** and **Create workflow job template** when adding a template.

Within **Automation Execution** → **Templates** → **Create template** → **Create workflow job template**, fill in the details:

Parameter	Value
Name	Deploy Webapp Server
Organization	Default

Click **Create workflow job template** to open the **Workflow Visualizer**.

The screenshot shows the "Workflow Visualizer" interface. At the top, it displays the title "Workflow Visualizer Deploy Webapp Server". Below the title are three buttons: "Save" (disabled), "Add step" (highlighted in blue), and "Total nodes 0". The main area is a dark gray canvas with a large white circle containing a black plus sign in the center. Below the canvas, the text "There are currently no nodes in this workflow" is displayed, followed by the instruction "Add a node by clicking the button below". A final "Add step" button is located at the bottom of the interface.

Click the **Add Step** button and assign the **Web App Deploy** job template to the first node. Add a second node by clicking the 3 dot sign, selecting the "Add step and link" and assign the **Node.js Deploy** template with the **Run on**

success status type. Select **Next** and **Finish** to complete the workflow.

Workflow Visualizer Deploy Webapp Server

Save Add step Total nodes 0

Add step

1 Node details 2 Review

Node type * Job Template
Job template * Web App Deploy
Convergence * Any
Node alias

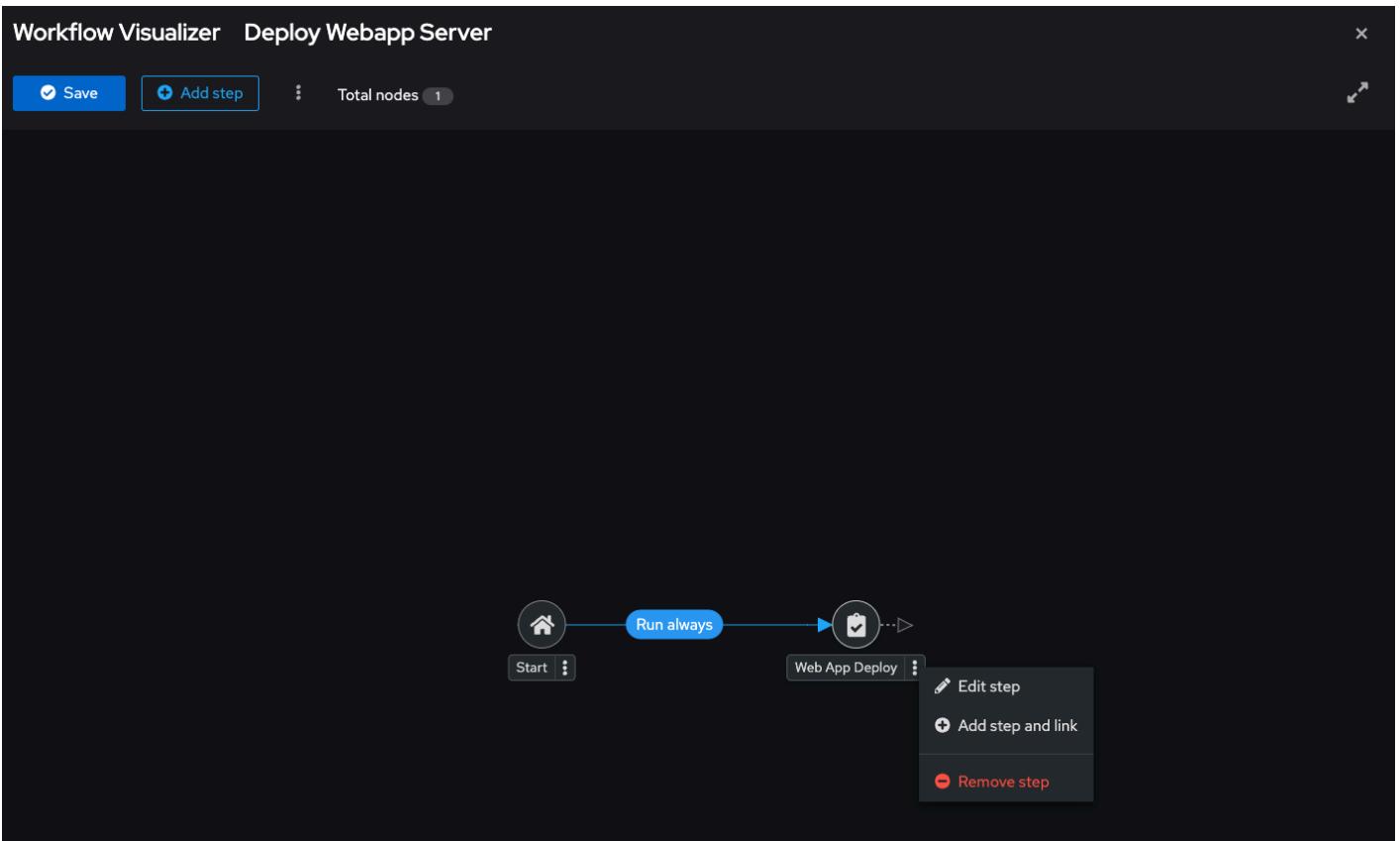
There are currently no nodes in this workflow

Add a node by clicking the button below

Add step

Next Back Cancel

The screenshot shows the 'Workflow Visualizer' interface with a dark theme. At the top, there's a header with 'Workflow Visualizer' and 'Deploy Webapp Server'. Below the header are buttons for 'Save' (disabled), 'Add step' (highlighted in blue), and 'Total nodes 0'. On the right side of the header is a close button (X). In the center, there's a large 'Add step' dialog box. The dialog has two tabs: 'Node details' (selected) and 'Review'. Under 'Node details', there are four dropdown menus: 'Node type' set to 'Job Template', 'Job template' set to 'Web App Deploy', 'Convergence' set to 'Any', and 'Node alias' which is empty. Below the dialog, a message says 'There are currently no nodes in this workflow' with a note to 'Add a node by clicking the button below'. A large blue 'Add step' button is centered. At the bottom of the screen, there are three buttons: 'Next' (highlighted in blue), 'Back', and 'Cancel'.



Workflow Visualizer Deploy Webapp Server

Save Add step Total nodes 1

Add step

1 Node details 2 Review

Node type *: Job Template

Job template *: Node.js Deploy

Status *: Run on success

Convergence *: Any

Node alias:

Next Back Cancel

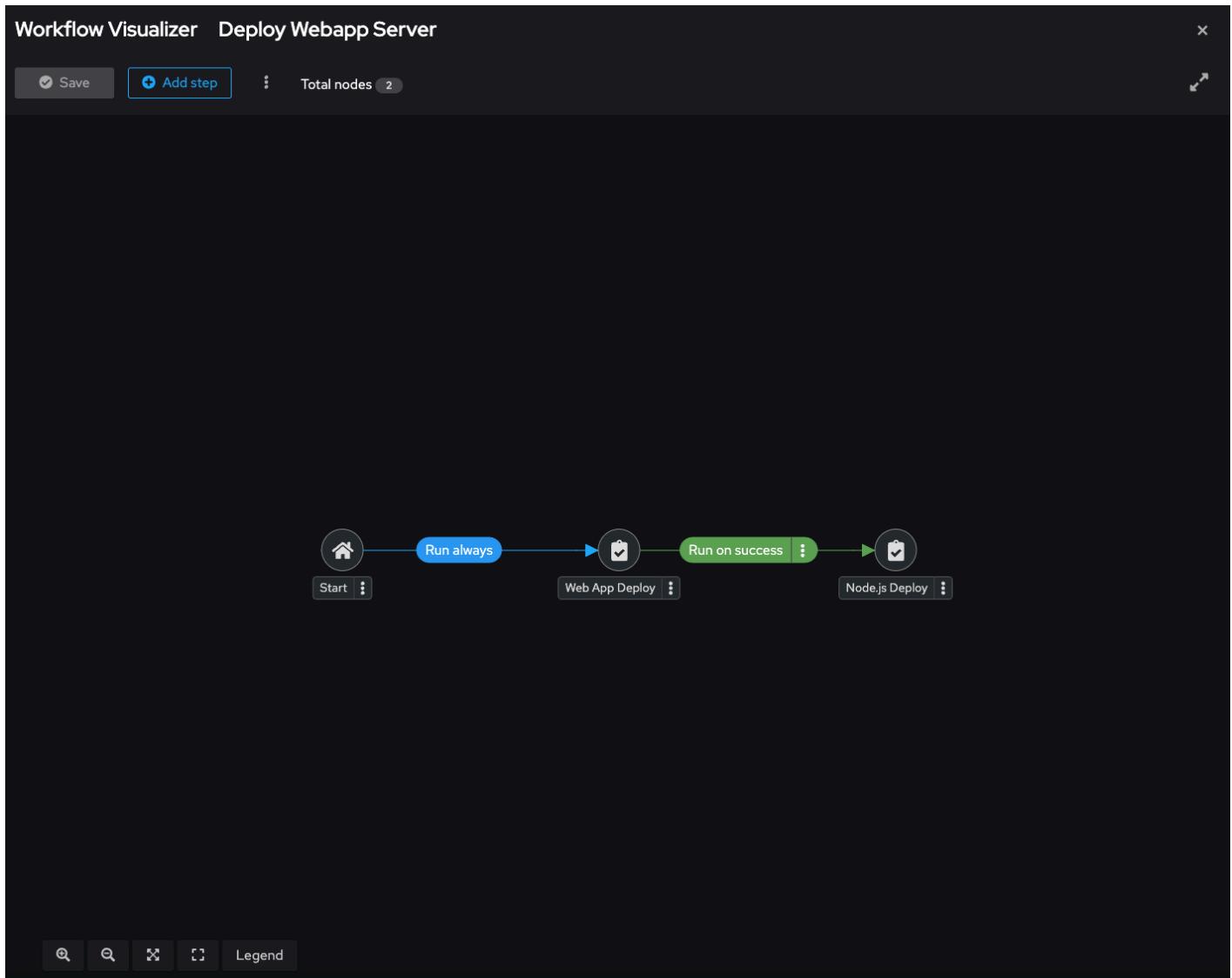
```
graph LR; Start((Start)) --> RunAlways[Run always]; RunAlways --> Deploy[Web App Deploy]
```

The same workflow diagram is shown, but now with an "Add step" dialog box overlaid on the right side. The dialog has two tabs: "Node details" (selected) and "Review". Under "Node details", there are fields for "Node type" (set to "Job Template"), "Job template" (set to "Node.js Deploy"), "Status" (set to "Run on success"), "Convergence" (set to "Any"), and "Node alias" (empty). At the bottom of the dialog are "Next" and "Cancel" buttons.

Click **Save** to finalize the workflow.

Tip

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



Launch workflow

Within the **Deploy Webapp Server** template, click **Launch template**.

The screenshot shows the 'Deploy Webapp Server' template details page. At the top, there are navigation links: 'Templates > Deploy Webapp Server > Details'. On the right, there are buttons for 'View workflow visualizer', 'Launch template', 'Edit template', and a more options menu. Below the header, there's a breadcrumb trail: 'Back to Templates' → 'Details'. A horizontal bar contains links for 'Team Access', 'User Access', 'Schedules', 'Jobs', 'Survey', and 'Notifications'. The main content area has sections for 'Name' (Deploy Webapp Server), 'Created' (10/17/2024, 4:01:58 PM by admin), and 'Last modified' (10/17/2024, 4:01:58 PM by admin). There's also a 'Extra vars' section with a file icon and buttons for 'YAML' and 'JSON'. The bottom of the page has a footer with links to 'Ansible Tower Documentation', 'Ansible Tower API Reference', and 'Ansible Tower Help'.

Once the workflow completes, verify the result.

- Go to **Automation Execution** → **Infrastructure** → **Inventories** → **Workshop Inventory**.
- Select the Hosts tab and select `node1` and click **Run Command**.
- Within the Details window, select Module `command`, in Arguments type `curl http://node1/nodejs` and click Next.
- Within the Execution Environment window, select `Default execution environment` and click Next.
- Within the Credential window, select `Workshop Credentials` and click Next.

Review your inputs and click Finish.

Verify that the output result shows `Hello World`

 **Note**

`xx` is the number of the job run.

© Tim Grützmacher 2025