Intro to playbooks

Ansible Playbooks offer a repeatable, re-usable, simple configuration management and multi-machine deployment system, one that is well suited to deploying complex applications. If you need to execute a task with Ansible more than once, write a playbook and put it under source control. Then you can use the playbook to push out new configuration or confirm the configuration of remote systems. The playbooks in the ansible-examples repository illustrate many useful techniques. You may want to look at these in another tab as you read the documentation.

Playbooks can:

- declare configurations
- orchestrate steps of any manual ordered process, on multiple sets of machines, in a defined order
- launch tasks synchronously or :ref:`asynchronously <playbooks_async>`

> **System Message: ERROR/3** (`D:\onboarding-resources\sample-onboarding-resources\ansible-devel\docs\docsite\rst\user_guide\(ansible-devel)(docs)(docsite)(rst)(user_guide)playbooks_intro.rst`, **line 14**); *backlink*
>
> Unknown interpreted text role "ref".

Playbook syntax

Playbooks are expressed in YAML format with a minimum of syntax. If you are not familiar with YAML, look at our overview of :ref:`yaml_syntax` and consider installing an add-on for your text editor (see :ref:`other_tools_and_programs`) to help you write clean YAML syntax in your playbooks.

> **System Message: ERROR/3** (`D:\onboarding-resources\sample-onboarding-resources\ansible-devel\docs\docsite\rst\user_guide\(ansible-devel)(docs)(docsite)(rst)(user_guide)playbooks_intro.rst`, **line 24**); *backlink*
>
> Unknown interpreted text role "ref".

> **System Message: ERROR/3** (`D:\onboarding-resources\sample-onboarding-resources\ansible-devel\docs\docsite\rst\user_guide\(ansible-devel)(docs)(docsite)(rst)(user_guide)playbooks_intro.rst`, **line 24**); *backlink*
>
> Unknown interpreted text role "ref".

A playbook is composed of one or more 'plays' in an ordered list. The terms 'playbook' and 'play' are sports analogies. Each play executes part of the overall goal of the playbook, running one or more tasks. Each task calls an Ansible module.

Playbook execution

A playbook runs in order from top to bottom. Within each play, tasks also run in order from top to bottom. Playbooks with multiple 'plays' can orchestrate multi-machine deployments, running one play on your webservers, then another play on your database servers, then a third play on your network infrastructure, and so on. At a minimum, each play defines two things:

- the managed nodes to target, using a :ref:`pattern <intro_patterns>`

> **System Message: ERROR/3** (`D:\onboarding-resources\sample-onboarding-resources\ansible-devel\docs\docsite\rst\user_guide\(ansible-devel)(docs)(docsite)(rst)(user_guide)playbooks_intro.rst`, **line 33**); *backlink*
>
> Unknown interpreted text role "ref".

- at least one task to execute

Note

In Ansible 2.10 and later, we recommend you use the fully-qualified collection name in your playbooks to ensure the correct module is selected, because multiple collections can contain modules with the same name (for example, `user`). See :ref:`collections_using_playbook`.

> **System Message: ERROR/3** (`D:\onboarding-resources\sample-onboarding-resources\ansible-devel\docs\docsite\rst\user_guide\(ansible-devel)(docs)(docsite)(rst)(user_guide)playbooks_intro.rst`, line 38); *backlink*
>
> Unknown interpreted text role "ref".

In this example, the first play targets the web servers; the second play targets the database servers.

```
---
- name: Update web servers
  hosts: webservers
  remote_user: root

  tasks:
  - name: Ensure apache is at the latest version
    ansible.builtin.yum:
      name: httpd
      state: latest
  - name: Write the apache config file
    ansible.builtin.template:
      src: /srv/httpd.j2
      dest: /etc/httpd.conf

- name: Update db servers
  hosts: databases
  remote_user: root

  tasks:
  - name: Ensure postgresql is at the latest version
    ansible.builtin.yum:
      name: postgresql
      state: latest
  - name: Ensure that postgresql is started
    ansible.builtin.service:
      name: postgresql
      state: started
```

Your playbook can include more than just a hosts line and tasks. For example, the playbook above sets a `remote_user` for each play. This is the user account for the SSH connection. You can add other :ref:`playbook_keywords` at the playbook, play, or task level to influence how Ansible behaves. Playbook keywords can control the :ref:`connection plugin <connection_plugins>`, whether to use :ref:`privilege escalation <become>`, how to handle errors, and more. To support a variety of environments, Ansible lets you set many of these parameters as command-line flags, in your Ansible configuration, or in your inventory. Learning the :ref:`precedence rules <general_precedence_rules>` for these sources of data will help you as you expand your Ansible ecosystem.

> **System Message: ERROR/3** (`D:\onboarding-resources\sample-onboarding-resources\ansible-devel\docs\docsite\rst\user_guide\(ansible-devel)(docs)(docsite)(rst)(user_guide)playbooks_intro.rst`, line 73); *backlink*
>
> Unknown interpreted text role "ref".

> **System Message: ERROR/3** (`D:\onboarding-resources\sample-onboarding-resources\ansible-devel\docs\docsite\rst\user_guide\(ansible-devel)(docs)(docsite)(rst)(user_guide)playbooks_intro.rst`, line 73); *backlink*
>
> Unknown interpreted text role "ref".

> **System Message: ERROR/3** (`D:\onboarding-resources\sample-onboarding-resources\ansible-devel\docs\docsite\rst\user_guide\(ansible-devel)(docs)(docsite)(rst)(user_guide)playbooks_intro.rst`, line 73); *backlink*
>
> Unknown interpreted text role "ref".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\ansible-devel\docs\docsite\rst\user_guide\(ansible-devel)(docs)(docsite)(rst)(user_guide)playbooks_intro.rst, line 73); *backlink*

Unknown interpreted text role "ref".

Task execution

By default, Ansible executes each task in order, one at a time, against all machines matched by the host pattern. Each task executes a module with specific arguments. When a task has executed on all target machines, Ansible moves on to the next task. You can use :ref:`strategies <playbooks_strategies>` to change this default behavior. Within each play, Ansible applies the same task directives to all hosts. If a task fails on a host, Ansible takes that host out of the rotation for the rest of the playbook.

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\ansible-devel\docs\docsite\rst\user_guide\(ansible-devel)(docs)(docsite)(rst)(user_guide)playbooks_intro.rst, line 80); *backlink*

Unknown interpreted text role "ref".

When you run a playbook, Ansible returns information about connections, the `name` lines of all your plays and tasks, whether each task has succeeded or failed on each machine, and whether each task has made a change on each machine. At the bottom of the playbook execution, Ansible provides a summary of the nodes that were targeted and how they performed. General failures and fatal "unreachable" communication attempts are kept separate in the counts.

Desired state and 'idempotency'

Most Ansible modules check whether the desired final state has already been achieved, and exit without performing any actions if that state has been achieved, so that repeating the task does not change the final state. Modules that behave this way are often called 'idempotent.' Whether you run a playbook once, or multiple times, the outcome should be the same. However, not all playbooks and not all modules behave this way. If you are unsure, test your playbooks in a sandbox environment before running them multiple times in production.

Running playbooks

To run your playbook, use the :ref:`ansible-playbook` command.

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\ansible-devel\docs\docsite\rst\user_guide\(ansible-devel)(docs)(docsite)(rst)(user_guide)playbooks_intro.rst, line 96); *backlink*

Unknown interpreted text role "ref".

```
ansible-playbook playbook.yml -f 10
```

Use the `--verbose` flag when running your playbook to see detailed output from successful modules as well as unsuccessful ones.

Ansible-Pull

Should you want to invert the architecture of Ansible, so that nodes check in to a central location, instead of pushing configuration out to them, you can.

The `ansible-pull` is a small script that will checkout a repo of configuration instructions from git, and then run `ansible-playbook` against that content.

Assuming you load balance your checkout location, `ansible-pull` scales essentially infinitely.

Run `ansible-pull --help` for details.

There's also a clever playbook available to configure `ansible-pull` via a crontab from push mode.

Verifying playbooks

You may want to verify your playbooks to catch syntax errors and other problems before you run them. The :ref:`ansible-playbook` command offers several options for verification, including `--check`, `--diff`, `--list-hosts`, `--list-tasks`, and `--syntax-check`. The :ref:`validate-playbook-tools` describes other tools for validating and testing playbooks.

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\ansible-devel\docs\docsite\rst\user_guide\(ansible-devel)(docs)(docsite)(rst)(user_guide)playbooks_intro.rst, line 124); *backlink*

Unknown interpreted text role "ref".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\ansible-devel\docs\docsite\rst\user_guide\(ansible-devel)(docs)(docsite)(rst)(user_guide)playbooks_intro.rst, line 124); *backlink*

Unknown interpreted text role "ref".

ansible-lint

You can use ansible-lint for detailed, Ansible-specific feedback on your playbooks before you execute them. For example, if you run `ansible-lint` on the playbook called `verify-apache.yml` near the top of this page, you should get the following results:

```
$ ansible-lint verify-apache.yml
[403] Package installs should not use latest
verify-apache.yml:8
Task/Handler: ensure apache is at the latest version
```

The ansible-lint default rules page describes each error. For `[403]`, the recommended fix is to change `state: latest` to `state: present` in the playbook.

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\ansible-devel\docs\docsite\rst\user_guide\(ansible-devel)(docs)(docsite)(rst)(user_guide)playbooks_intro.rst, line 142)

Unknown directive type "seealso".

```
.. seealso::

   `ansible-lint <https://docs.ansible.com/ansible-lint/index.html>`_
       Learn how to test Ansible Playbooks syntax
   :ref:`yaml_syntax`
       Learn about YAML syntax
   :ref:`playbooks_best_practices`
       Tips for managing playbooks in the real world
   :ref:`list_of_collections`
       Browse existing collections, modules, and plugins
   :ref:`developing_modules`
       Learn to extend Ansible by writing your own modules
   :ref:`intro_patterns`
       Learn about how to select hosts
   `GitHub examples directory <https://github.com/ansible/ansible-examples>`_
       Complete end-to-end playbook examples
   `Mailing List <https://groups.google.com/group/ansible-project>`_
       Questions? Help? Ideas?  Stop by the list on Google Groups
```