**Including and Importing**

**Topics**

* Including and Importing
  + Includes vs. Imports
  + Importing Playbooks
  + Including and Importing Task Files
  + Including and Importing Roles

**Includes vs. Imports**

As noted in Creating Reusable Playbooks, include and import statements are very similar, however the Ansible executor engine treats them very differently.

* All import\* statements are pre-processed at the time playbooks are parsed.
* All include\* statements are processed as they encountered during the execution of the playbook.

Please refer to Creating Reusable Playbooks for documentation concerning the trade-offs one may encounter when using each type.

Also be aware that this behaviour changed in 2.4; prior to that Ansible version only include was available, and it behaved differently depending on context.

New in version 2.4.

**Importing Playbooks**

It is possible to include playbooks inside a master playbook. For example:

---

- import\_playbook: webservers.yml

- import\_playbook: databases.yml

The plays and tasks in each playbook listed will be run in the order they are listed, just as if they had been defined here directly.

Prior to 2.4 only include was available and worked for both playbooks and tasks as both import and include.

New in version 2.4.

**Including and Importing Task Files**

Use of included task lists is a great way to define a role that system is going to fulfill. A task include file simply contains a flat list of tasks:

*# common\_tasks.yml*

---

- name: placeholder foo

command: /bin/foo

- name: placeholder bar

command: /bin/bar

You can then use import\_tasks or include\_tasks to include this file in your main task list:

tasks:

- import\_tasks: common\_tasks.yml

*# or*

- include\_tasks: common\_tasks.yml

You can also pass variables into imports and includes:

tasks:

- import\_tasks: wordpress.yml wp\_user=timmy

- import\_tasks: wordpress.yml wp\_user=alice

- import\_tasks: wordpress.yml wp\_user=bob

Variables can also be passed to include files using an alternative syntax, which also supports structured variables like dictionaries and lists:

tasks:

- include\_tasks: wordpress.yml

vars:

wp\_user: timmy

ssh\_keys:

- "**{{** lookup**(**'file'**,** 'keys/one.pub'**)** **}}**"

- "**{{** lookup**(**'file'**,** 'keys/two.pub'**)** **}}**"

Using either syntax, variables passed in can then be used in the included files. These variables will only be available to tasks within the included file. See Variable Precedence: Where Should I Put A Variable? for more details on variable inheritance and precedence.

Task include statements can be used at arbitrary depth.

**Note**

Static and dynamic can be mixed, however this is not recommended as it may lead to difficult-to-diagnose bugs in your playbooks.

Includes and imports can also be used in the handlers: section; for instance, if you want to define how to restart apache, you only have to do that once for all of your playbooks. You might make a handlers.yml that looks like:

*# more\_handlers.yml*

---

- name: restart apache

service: name=apache state=restarted

And in your main playbook file:

handlers:

- include\_tasks: more\_handlers.yml

*# or*

- import\_tasks: more\_handlers.yml