Play Argument Validation

- Specification Format
- Sample Specification

Beginning in version 2.20, you can enable argument validation using the play keyword `validate_argspec`. This adds a `validate_argument_spec <ansible.builtin.validate_argument_spec#module>` task following play-level fact gathering. This feature is tech preview.

Play argument validation has two main parts:

- The `validate_argspec` keyword.
- A `.meta.yml` file that defines argument specifications.

To enable play argument validation you:

1. Define the argument specification identifier by setting a value for the `validate_argspec` keyword. You can set the value to `True` to use the play `name` or you can set the value to a string.
2. Provide a valid argument specification in a `<playbook_name>.meta.yml` file in the same directory as the playbook.

The following example provides a valid, empty argument specification named `setup webserver` for the playbook `create_webserver.yml`:

```
# create_webserver.meta.yml
argument_specs:
  setup webserver:
    options: {}
```

In the following playbook, both plays validate the play arguments against the `setup webserver` arguments:

```
# create_webserver.yml
- name: setup webserver
  hosts: all
```

```
  validate_argspec: True

- hosts: all
  validate_argspec: setup webserver
```

Specification Format

The play argument specification must be defined in a top-level `argument_specs` block within the playbook's `.meta.yml` file. All fields are lowercase.

argument-spec-name

> • The name of the play or argument specification.
>
> description
>
> > • A description of the play that may contain multiple lines.
> > • This can be a single string or a list of strings.
>
> options
>
> > • This section defines the dictionary of play arguments.
> > • For each play option (argument), you may include:
> >
> > option-name
> >
> > > • The name of the option/argument (required).
> >
> > description

- Detailed explanation of what this option does. It should be written in full sentences.
- This can be a single string or a list of strings.

type

- The data type of the option. See `Argument spec <argument_spec>` for allowed values for `type`. The default is `str`.
- If an option is of type `list`, `elements` should be specified.

required

- Only needed if `true`.
- If missing, the option is not required.

choices

- List of option values.
- Should be absent if empty.

elements

- Specifies the data type for list elements when the type is `list`.

options

- If this option takes a dict or list of dicts, you can define the structure here.

Sample Specification

```
# create_webservers.meta.yml
description: Set up basic HTTPS-enabled webserver to serve content from a specified document
root.
argument_specs:
  setup webserver:
    options:
      document_root:
        description: Path to the directory containing static web content to be served.
        type: str
        required: True
      port:
        description:
          - Port number on which the webserver listens for incoming HTTPS connections.
          - When unspecified, the port is 443.
        type: int
      ssl_cert_path:
        description: Path to the SSL certificate.
        type: str
        required: True
      ssl_key_path:
        description: Path to the private key corresponding to the SSL certificate.
        type: str
        required: True
```

□□□ #2
□ □□□ □□□ 2026-10-31 14:41:06 UTC
□ □□□ □□□ 2026-10-31 14:41:06 UTC