Ansible Handlers

Eyes On Cloud

What are Handlers?

- Handlers are commonly used in Ansible to start, reload, restart, and stop services.
- If your playbook calls for changing configuration files, you'll almost certainly need to restart a service to see the changes take effect.
- In this case, you must define a handler for that service and include the notify directive in any tasks that require it.
- Handlers are executed last by default, regardless of where they are in the playbook.
- In a playbook, you can define and call one or more handlers based on the tasks to be completed.

Why use ansible handlers?

- Handlers, as previously stated, are similar to other tasks in a playbook, with the exception that they are triggered using the notify directive and are only executed when the state changes.

Points on handlers

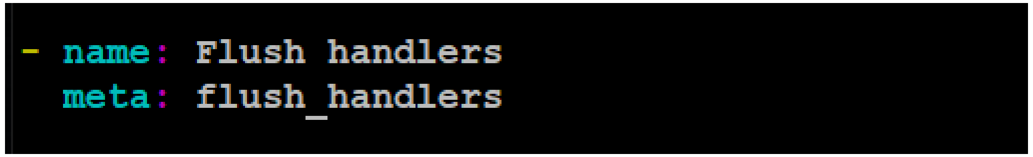
- Within the playbook, a handler should have a globally unique name.
- Only the first handler will be called if more than one with the same name are defined. All other handlers will be disregarded.
- Handlers are always executed in the order they are defined in the handler’s section, not in the notify section.
- A handler task only runs if the state changes; otherwise, it does not.
- The notify and handlers directives are used to define a handler. The notify directive causes the task(s) specified in the handler's section to be executed.

Using the ‘listen’ Directive to Group Handlers

- You can use the listen keyword to group handlers and call them all with a single notify statement.
- The notify directive is set to restart services in the following playbook, which triggers all handler tasks with the listen directive.

Determine When Handlers Should Run

- If you want handlers to run before the end of the play or between tasks, use the meta module to add a meta task to flush them.
- The following is the definition of the meta task.



```
- name: Flush handlers
  meta: flush_handlers
```