

Ansible Handlers – Explained in Detail

Handlers in Ansible are special tasks that run **only when notified** by another task. They're mainly used to **trigger actions after a change**, such as restarting a service after a config file is updated.



Why Use Handlers?

- **Efficiency:** Run only when needed.
 - **Idempotency:** Prevent unnecessary restarts or reloads.
 - **Clean Structure:** Keep logic for “post-change actions” separate.
-



Basic Structure of a Handler

Example: Restart Nginx if config changes

```
- name: Configure web server
hosts: web
tasks:
  - name: Copy nginx config
    copy:
      src: nginx.conf
      dest: /etc/nginx/nginx.conf
    notify: Restart nginx
```

handlers:

```
  - name: Restart nginx
    service:
      name: nginx
      state: restarted
```

- The handler `Restart nginx` only runs if the `copy` task reports a change.
-

What Triggers a Handler?

Any task that has:

```
notify: <handler-name>
```

Multiple tasks can notify the same handler, and **the handler runs only once at the end of the play**, regardless of how many times it's notified.

Multiple Handlers

You can define and notify multiple handlers:

```
notify:  
  - Restart nginx  
  - Reload firewall
```

Handlers Run Only If Changed

Handlers **will not run** if the notifying task did **not** change anything.

Example: Full Playbook with Handlers

```
- name: Install and configure Apache  
  hosts: web  
  tasks:  
    - name: Install httpd  
      yum:  
        name: httpd  
        state: present  
  
    - name: Deploy config file  
      template:  
        src: httpd.conf.j2  
        dest: /etc/httpd/conf/httpd.conf  
      notify: Restart Apache
```

```
handlers:  
  - name: Restart Apache  
    service:  
      name: httpd  
      state: restarted
```

Important Notes

- Handlers run **after all tasks are complete**, not immediately after being notified.
 - Use **unique names** for each handler.
 - A handler is **skipped** if no notifying task reports a change.
-

Advanced Tip: Force a Handler

Use the `meta` module if you ever want to force a handler run mid-playbook:

```
- meta: flush_handlers
```

This will cause **all pending handlers to run immediately** at that point in the play.

Use Cases for Handlers

Task	Handler Trigger
Update a config file	Restart or reload service
Add firewall rule	Reload firewall
Install a new cron job	Restart cron daemon
Change SSL cert	Reload web server or load balancer
