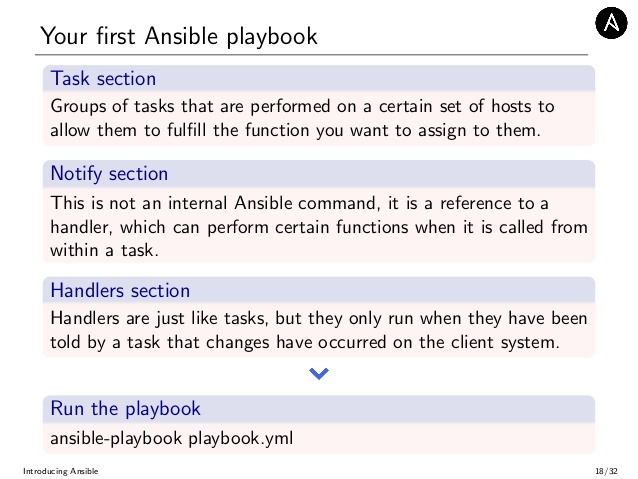
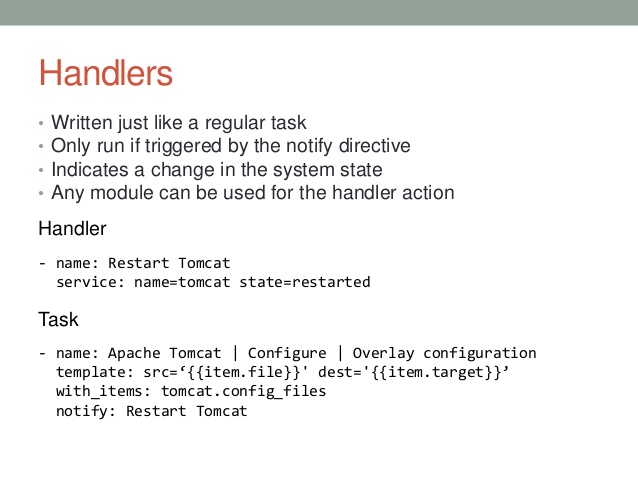
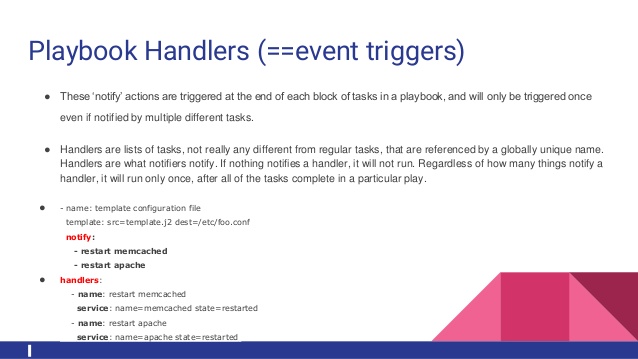
Handlers

A Handler is exactly the same as a Task, but it will run when called by another Task. A Handler will take an action when called by an event it listens for.

This is useful for secondary actions that might be required after running a Task, such as starting a new service after installation or reloading a service after a configuration change.







nginx-and-handler.yml

Here is our first Playbook with the Handler:

---

- hosts: aws

tasks:

- name: Install Nginx

apt: pkg=nginx state=installed update\_cache=true

notify:

- Start Nginx

handlers:

- name: Start Nginx

service: name=nginx state=started

The **aws** is defined in **/etc/ansible/hosts**:

[aws]

54.153.0.23

We can run it with the **ansible-playbook** command:

$ ansible-playbook -s nginx-and-handler.yml -u ubuntu

PLAY [aws] \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

GATHERING FACTS \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

ok: [54.153.0.23]

TASK: [Install Nginx] \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

changed: [54.153.0.23]

NOTIFIED: [Start Nginx] \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

changed: [54.153.0.23]

PLAY RECAP \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

54.153.0.23 : ok=3 changed=2 unreachable=0 failed=0

We used **-s** to tell Ansible to use **sudo**, **-u** to login as **ubuntu** user, and then pass the **Playbook** file.

We can add a **notify** directive to the installation Task. This notifies any Handler named "Start Nginx" after the Task is run. Then we can create the Handler called "Start Nginx". This Handler is the Task called when "Start Nginx" is notified.

This particular Handler uses the Service module, which can start, stop, restart, reload system services. Here we simply tell Ansible that we want Nginx to be started.

If we try to install again, we get **changed=0** since it's been already installed:

$ ansible-playbook -s nginx-and-handler.yml -u ubuntu

PLAY [aws] \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

GATHERING FACTS \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

ok: [54.153.0.23]

TASK: [Install Nginx] \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

ok: [54.153.0.23]

PLAY RECAP \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

54.153.0.23 : ok=2 changed=0 unreachable=0 failed=0

Note also this time the Handler did not run.

Adding more Tasks

We can add a few more Tasks to this Playbook:

---

- hosts: aws

vars:

- docroot: /var/www/bogotobogo.com/public

tasks:

- name: Add Nginx Repository

apt\_repository: repo='ppa:nginx/stable' state=present

register: ppastable

- name: Install Nginx

apt: pkg=nginx state=installed update\_cache=true

when: ppastable|success

register: nginxinstalled

notify:

- Start Nginx

- name: Create Web Root

when: nginxinstalled|success

file: dest=/var/www/bogotobogo.com/public mode=775 state=directory owner=www-data group=www-data

notify:

- Reload Nginx

handlers:

- name: Start Nginx

service: name=nginx state=started

- name: Reload Nginx

service: name=nginx state=reloaded

Run the Playbook:

$ ansible-playbook -s nginx-and-handler-with-more-tasks.yml -u ubuntu

PLAY [aws] \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

GATHERING FACTS \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

ok: [54.153.0.23]

TASK: [Add Nginx Repository] \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

ok: [54.153.0.23]

TASK: [Install Nginx] \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

ok: [54.153.0.23]

TASK: [Create Web Root] \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

changed: [54.153.0.23]

NOTIFIED: [Reload Nginx] \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

changed: [54.153.0.23]

PLAY RECAP \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

54.153.0.23 : ok=5 changed=2 unreachable=0 failed=0

In the Playbook we have the following Tasks:

1. Add Nginx Repository - Add the Nginx stable PPA to get the latest stable version of Nginx, using the apt\_repository module.
2. Install Nginx - Installs Nginx using the Apt module.
3. Create Web Root - Finally, create a web root directory.

Also new here are the **register** and **when** directives. These tell Ansible to run a Task when something else happens.

The "Add Nginx Repository" Task registers "ppastable". Then we use that to inform the Install Nginx Task to only run when the registered "ppastable" Task is successful. This allows us to conditionally stop Ansible from running a Task.