# Ansible - Advanced Execution

## How to Limit Execution by Tasks

This is a very important execution strategy where one needs to execute only one execution and not the entire playbook. **For example**, suppose you only want to stop a server (in case a production issue comes) and then post applying a patch you would like to only start the server.

Here in original playbook stop and start were a part of different roles in the same playbook but this can be handled with the usage of tags. We can provide different tags to different roles (which in turn will have tasks) and hence based on the tags provided by the executor only that specified role/task gets executed. So for the above example provided, we can add tags like the following −

- {role: start-tomcat, tags: ['install']}}

The following command helps in using tags −

ansible-playbook -i hosts <your yaml> --tags "install" -vvv

With the above command, only the start-tomcat role will be called. The tag provided is case-sensitive. Ensure exact match is being passed to the command.

## How to Limit Execution by Hosts

There are two ways to achieve the execution of specific steps on specific hosts. For a specific role, one defines the hosts - as to which specific hosts that specific role should be run.

### Example

- hosts: <A>

environment: "{{your env}}"

pre\_tasks:

- debug: msg = "Started deployment.

Current time is {{ansible\_date\_time.date}} {{ansible\_date\_time.time}} "

roles:

- {role: <your role>, tags: ['<respective tag>']}

post\_tasks:

- debug: msg = "Completed deployment.

Current time is {{ansible\_date\_time.date}} {{ansible\_date\_time.time}}"

- hosts: <B>

pre\_tasks:

- debug: msg = "started....

Current time is {{ansible\_date\_time.date}} {{ansible\_date\_time.time}} "

roles:

- {role: <your role>, tags: ['<respective tag>']}

post\_tasks:

- debug: msg = "Completed the task..

Current time is {{ansible\_date\_time.date}} {{ansible\_date\_time.time}}"

As per the above example, depending on the hosts provided, the respective roles will only be called. Now my hosts A and B are defined in the hosts (inventory file).

### Alternate Solution

A different solution might be defining the playbook's hosts using a variable, then passing in a specific host address via **--extra-vars** −

# file: user.yml (playbook) --- - hosts: '{{ target }}'

user: ...

playbook contd….

### Running the Playbook

ansible-playbook user.yml --extra-vars "target = "<your host variable>"

If {{ target }} isn't defined, the playbook does nothing. A group from the hosts file can also be passed through if need be. This does not harm if the extra vars is not provided.

### Playbook targeting a single host

$ ansible-playbook user.yml --extra-vars "target = <your hosts variable>" --listhosts