Day two highlights

Facts

- Information gathered about the managed hosts
- Specific to the host being managed (IP, memory, disk info, date/time and etc)
- Gathered by the **setup** module

```
$ ansible all  -m setup
```

- Use debug module to show the **ansible_facts** variable contents
- Facts are gathered automatically by default when running **playbooks**
- Turn off using **gather_facts: false**
- Custom facts can be set on each managed host:

/etc/ansible/facts.d/factfile.fact

Magic variables

- hostvars
- inventory_hostname
- groups
- group_names

- *You can use debug module to explore these variables*

Task Control - loops

```
-   name: First Play

    hosts: all

    tasks:

        - service:

            name: “{{ item }}"

            state: started

          loop:

            - httpd

            - firewalld
```

Older loop structures as follows::

- with_nested
- with_file
- with_fileglob
- with_random_choice
- with_sequence

and others

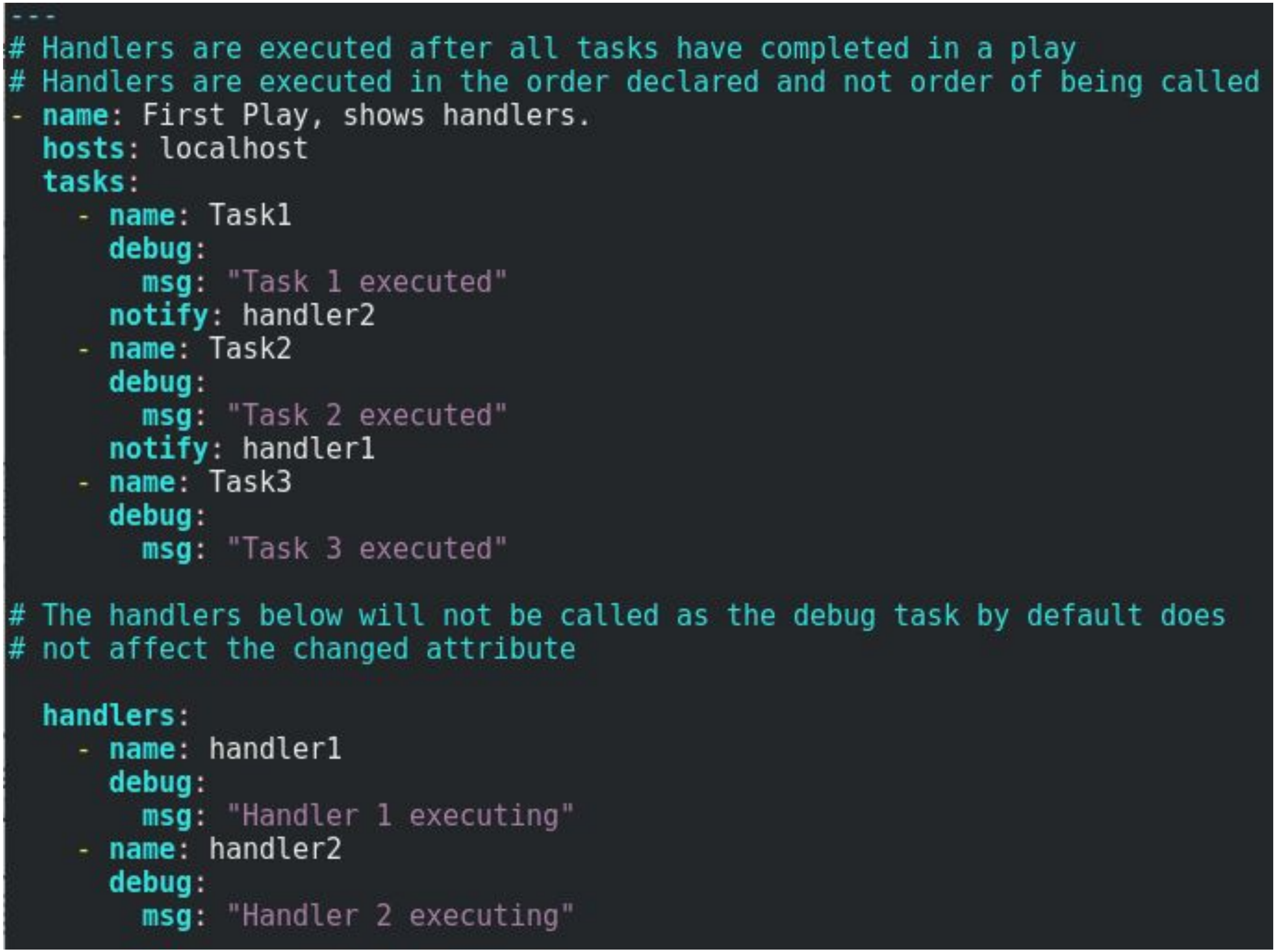
Task Control - conditions

```
-   name: First Play
    hosts: all
    tasks:
       - service:
            name: httpd
            state: started
         when: CONDITION1 [ and|or  CONDITIONX]...
```

CONDITIONS can also be specified as a **list of items**. In that case, ALL conditions in the list must be met in order for the module to be executed (equivalent to logical **and**)

Handlers

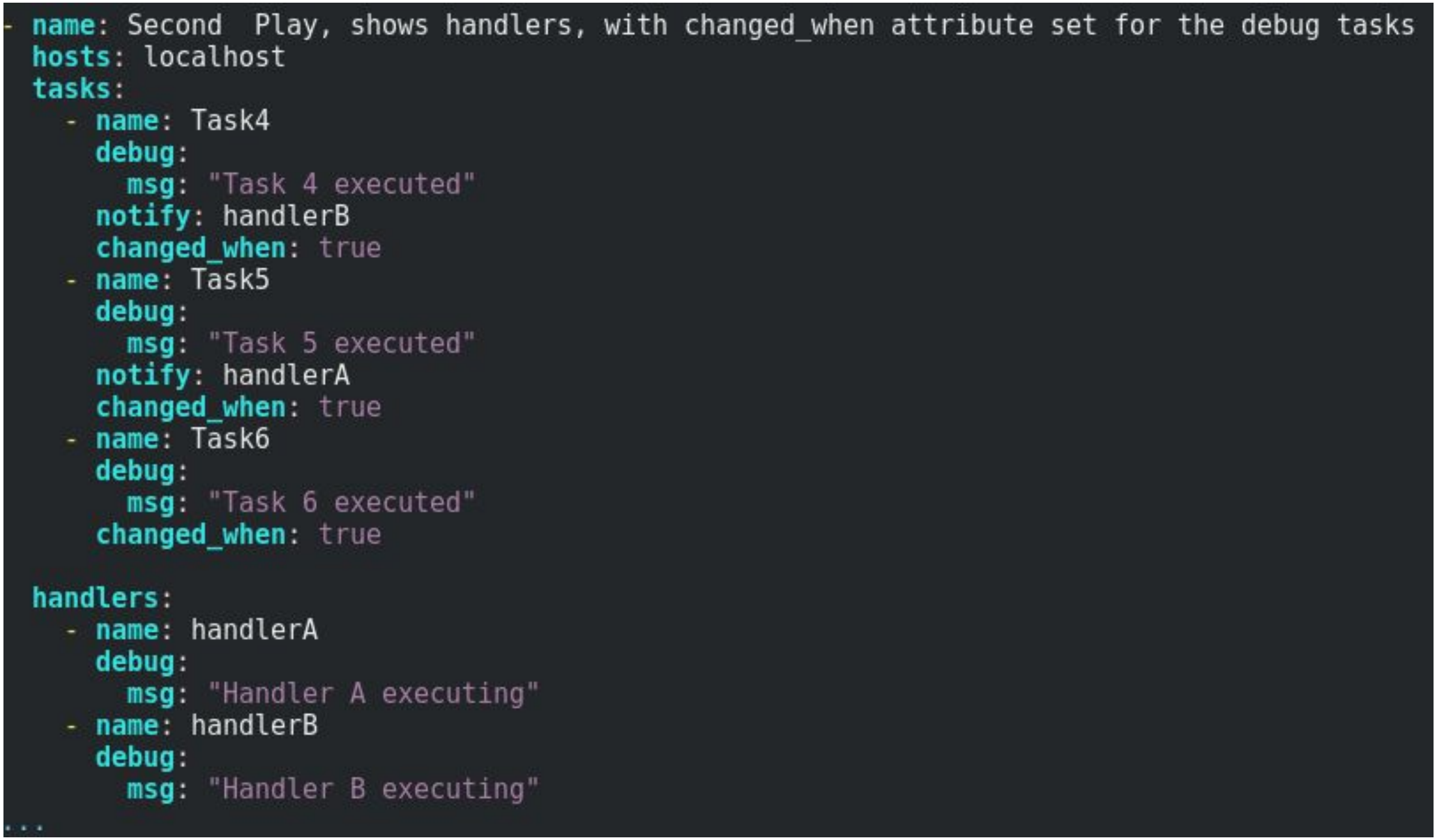
- Will only be executed if a **module has a changed status and notifies the handler**
- If to be executed, it executes after all tasks have executed
- If multiple handlers are notified, the order or execution depends on the order the handlers are declared in the playbook and not the order of notification
- Handlers need names as we need to **notify** the handlers by name
- The **handlers** key is at the same indentation level as the **tasks** key



```yaml
---
# Handlers are executed after all tasks have completed in a play
# Handlers are executed in the order declared and not order of being called
- name: First Play, shows handlers.
  hosts: localhost
  tasks:
    - name: Task1
      debug:
        msg: "Task 1 executed"
      notify: handler2
    - name: Task2
      debug:
        msg: "Task 2 executed"
      notify: handler1
    - name: Task3
      debug:
        msg: "Task 3 executed"

# The handlers below will not be called as the debug task by default does
# not affect the changed attribute

  handlers:
    - name: handler1
      debug:
        msg: "Handler 1 executing"
    - name: handler2
      debug:
        msg: "Handler 2 executing"
```



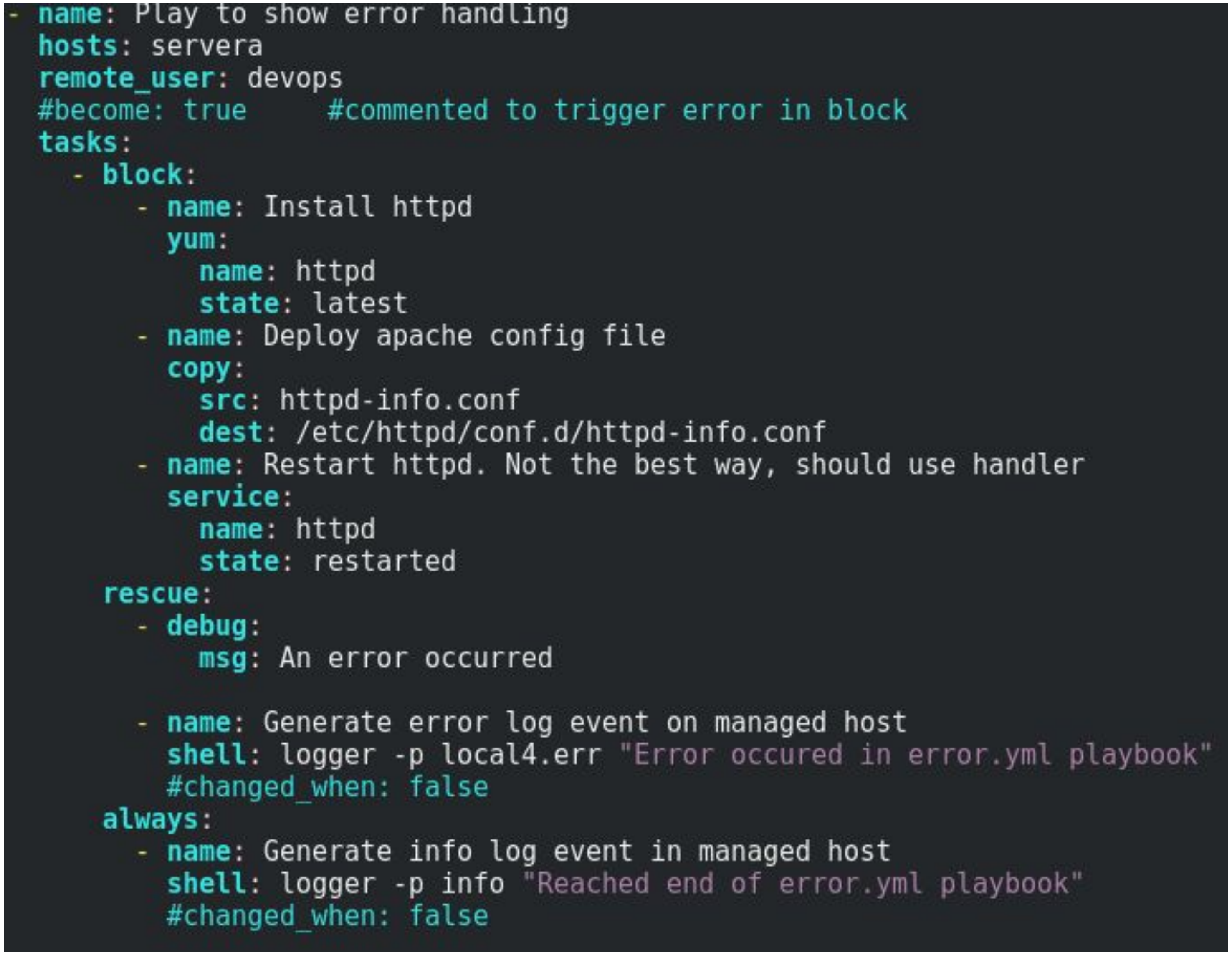
```yaml
- name: Second  Play, shows handlers, with changed_when attribute set for the debug tasks
  hosts: localhost
  tasks:
    - name: Task4
      debug:
        msg: "Task 4 executed"
      notify: handlerB
      changed_when: true
    - name: Task5
      debug:
        msg: "Task 5 executed"
      notify: handlerA
      changed_when: true
    - name: Task6
      debug:
        msg: "Task 6 executed"
      changed_when: true

  handlers:
    - name: handlerA
      debug:
        msg: "Handler A executing"
    - name: handlerB
      debug:
        msg: "Handler B executing"
...
```

Handling Errors

```
-   name: First Play
    hosts: all
    force_handlers: true|false
    tasks:
        - service:
              name: httpd
              state: started
          ignore_errors: true|false
          failed_when: CONDITION
          changed_when: CONDITION
```

Blocks



```
- name: Play to show error handling
  hosts: servera
  remote_user: devops
  #become: true     #commented to trigger error in block
  tasks:
    - block:
        - name: Install httpd
          yum:
            name: httpd
            state: latest
        - name: Deploy apache config file
          copy:
            src: httpd-info.conf
            dest: /etc/httpd/conf.d/httpd-info.conf
        - name: Restart httpd. Not the best way, should use handler
          service:
            name: httpd
            state: restarted
      rescue:
        - debug:
            msg: An error occurred

        - name: Generate error log event on managed host
          shell: logger -p local4.err "Error occured in error.yml playbook"
          #changed_when: false
      always:
        - name: Generate info log event in managed host
          shell: logger -p info "Reached end of error.yml playbook"
          #changed_when: false
```