Ansible Blocks

What are Blocks?

- Block feature is used to allow for logical grouping of tasks and even in play error handling.
- Most of what you can apply to a single task can be applied at the block level, which also makes it much easier to set data or directives common to the tasks.
- This does not mean the directive affects the block itself, but is inherited by the tasks enclosed by a block. i.e. a when will be applied to the tasks, not the block itself.

Example

- In this example, each of the 3 tasks will be executed after appending the when condition from the block and evaluating it in the task's context.
- Also, they inherit the privilege escalation directives enabling "become to root" for all the enclosed tasks.

```
tasks:
  - name: Install Apache
    block:
      - yum: name={{ item }} state=installed
        with_items:
          - httpd
          - memcached
      - template: src=templates/src.j2 dest:/etc/foo.conf
      - service: name=bar state: started enabled: True
    when: ansible_facts['distribution'] == 'CentOS'
    become: true
    become_user: root
```

Error Handling

- Blocks also introduce the ability to handle errors in a way similar to exceptions in most programming languages.
- The tasks in the block would execute normally, if there is any error the rescue section would get executed with whatever you need to do to recover from the previous error.
- The always section runs no matter what previous error did or did not occur in the block and rescue sections.

Example


```
tasks:
  - name: Attempt and gracefull roll back demo
    block:
     - debug: msg='I execute normally'
     - command: /bin/false
     - debug: msg: 'I never execute, due to the above task failing, :-('
    rescue:
     - debug: msg='I caught an error'
     - command: /bin/false
     - debug: msg='i also never execute :-('
    always:
     - debug: msg="this always executes"
```