DevOps Tools Day14

Prakash Ramamurthy
prakash.ramamurthy@wipro.com

Agenda

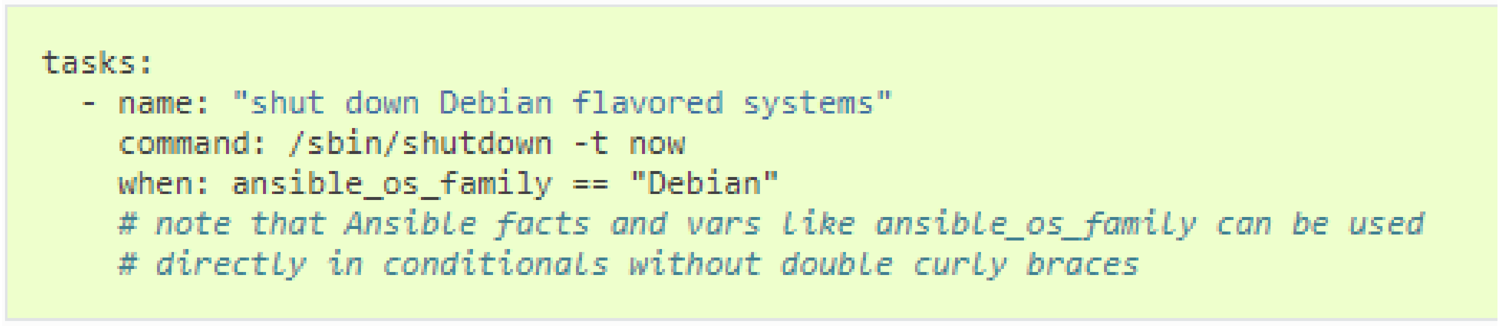
- Conditionals and Loops
- Blocks and Error Handling
- Ansible Vaults
- Hands-On Demonstration

Conditionals and Loops

Conditionals and Loops

When Statement

- Uses host specific environment collected by facts
- Helps to skip a particular steop while playbook is running on a particular host
- Skip installing a particular package not suitable for a host
- Install specific version of a package considering the host environment

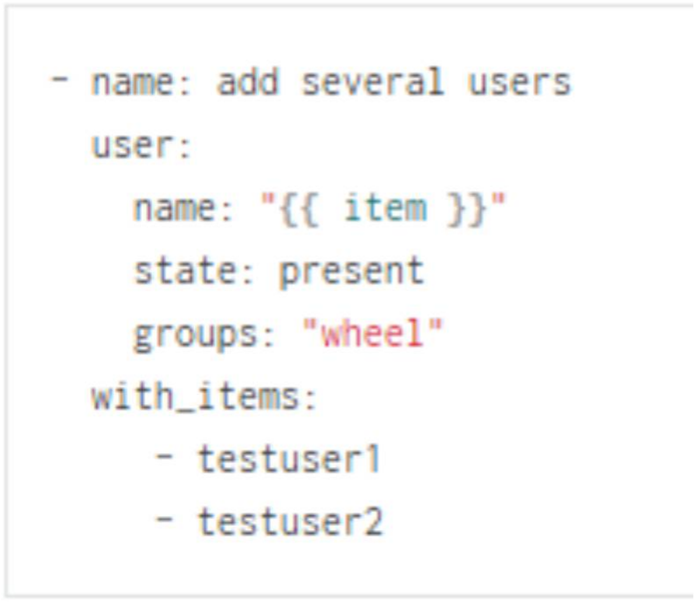


```
tasks:
  - name: "shut down Debian flavored systems"
    command: /sbin/shutdown -t now
    when: ansible_os_family == "Debian"
    # note that Ansible facts and vars like ansible_os_family can be used
    # directly in conditionals without double curly braces
```

Conditionals and Loops

Loops

- Loops do many things in a single task
- To create multiple users
- To install multiple packages
- Keep polling until a certain state is reached
- Loops can also be nested



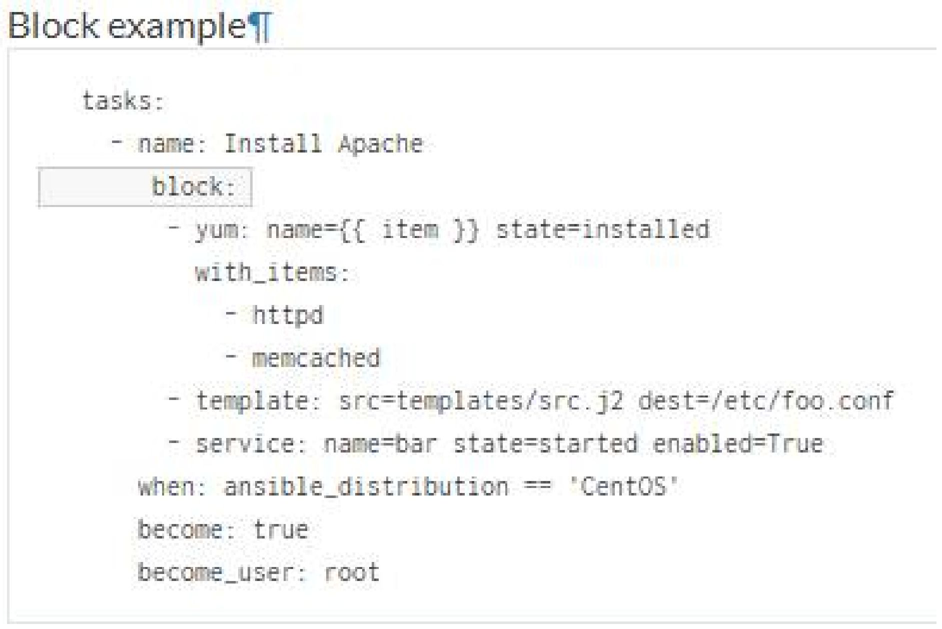
```
- name: add several users
  user:
    name: "{{ item }}"
    state: present
    groups: "wheel"
  with_items:
     - testuser1
     - testuser2
```

Blocks and Error Handling

Conditionals and Loops

Blocks

- Allow logical grouping of tasks
- Allow error handling during the "play"
- Allows applying a data or directive to entire block
- Tasks inside the block make use of them individually



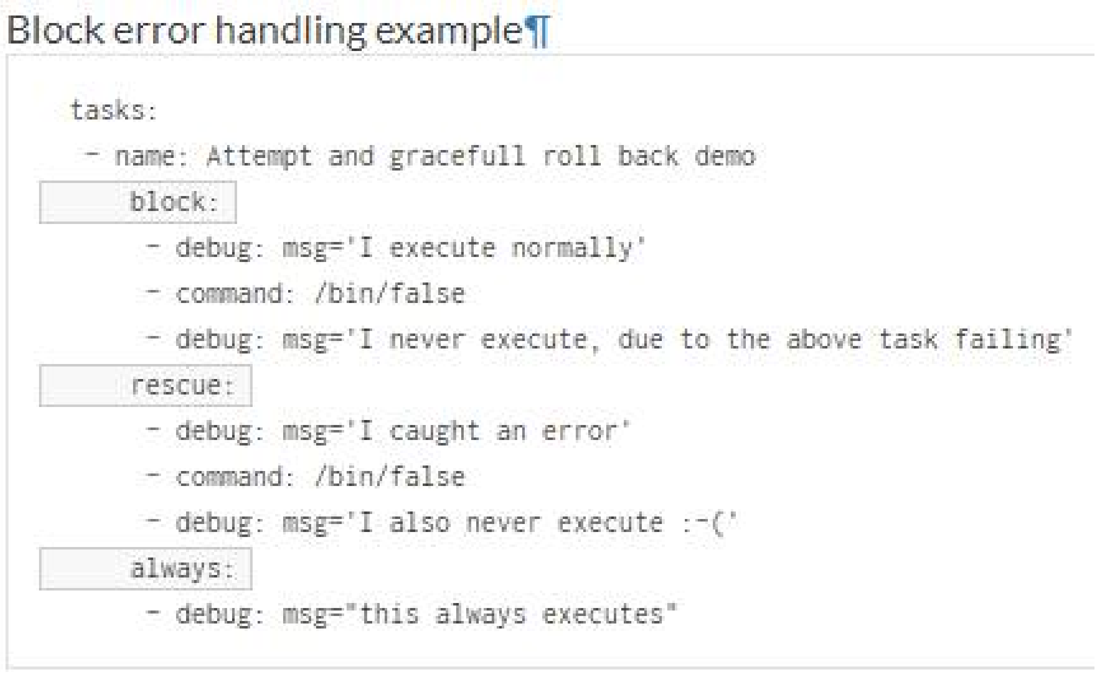
Block example¶

```
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_distribution == 'CentOS'
    become: true
    become_user: root
```

Conditionals and Loops

Error Handling

- Blocks provide the ability to handle errors
- Works similar to exceptions in programming languages


Block error handling 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"
```

Ansible Vaults

Ansible Vaults

Vault

Ansible Vaults

Vault

- Helps to protect passwords or keys kept in files in encrypted way
- Vault files (encrypted files), can be either distributed as is or can also be pushed in source control repos.
- Can encrypt any structured data file
- If vault files are used as “src” argument with template, unarchive, modules then it will be decrypted automatically while placing in “dest”. Vault password must be supplied.

Hands-On Demonstration

Hands-On Demonstration

Setup Module

```
$ ansible localhost -m setup >> facts

$ ansible localhost -m setup | grep "os_family"
$ ansible localhost -m setup | grep "ansible_architecture“

$ ansible localhost -m setup | grep "ansible_distribution"

$ ansible localhost -m setup | grep "ansible_nodename“

$ ansible localhost -m setup | grep "ansible_pkg_mgr“

$ ansible localhost -m setup | grep "ansible_processor“

$ ansible localhost -m setup | grep "ansible_user“

$ ansible localhost -m setup | grep
"ansible_python_version“

$ ansible localhost -m setup | grep "ansible_product_name“

$ ansible localhost -m setup | grep "ansible_virtualization"
```

Hands-On Demonstration

When Condition

- Playbook using When Condition:
- $ cat when.yaml

```
---
 - hosts: localhost
   tasks:
   - name: print the platform family
     shell: echo $ansible_os_family
     when: ansible_os_family == "Debian"

   - name: Non Debian Machine
     shell: echo "some other family"
     when: ansible_os_family == "RedHat"
```

- Run the Playbook

```
$ ansible-playbook -v when.yaml
```

Hands-On Demonstration

Sudo Privilege

- To create users
- `$ cat user.yaml`

```
- hosts: localhost
  become: yes
  tasks:
  - name: add several users
    user:
      name: testuser
      state: present
      groups: "docker"
```

- Run the Playbook

```
$ ansible-playbook user.yaml
```

Hands-On Demonstration

Loops - 1

- Using Loops to add several users

```
$ cat loop.yaml
- hosts: localhost
  become: yes
  tasks:
  - name: add several users
    user:
      name: "{{ item }}"
      state: present
      groups: "docker"
    with_items:
      - testuser1
      - testuser2
```

- Run the Playbook

```
$ ansible-playbook loop.yaml
```

Hands-On Demonstration

Loops - 2

- Using Loops in variables

```
$ cat loopvar.yaml
- hosts: localhost
  become: yes

  vars:
    users_with_items:
      - name: "testuser3"
      - name: "testuser4"

  tasks:
  - name: add several users
    user:
      name: "{{ item.name }}"
      state: present
      groups: "docker"
    with_items: " {{users_with_items }}"
```

Hands-On Demonstration

Loops - 3

- Using Loops to create personal directories for users:

```
$ cat loopdir.yaml
- hosts: localhost
  become: yes
  become_user: osgdev

  vars:
    users_with_items:
      - name: "testuser5"
        personal_directories:
        - "tu05"
      - name: "testuser6"
        personal_directories:
        - "tu06"
```

```
tasks:
  - name: User with directories
    user:
      name: "{{ item.name }}"
    with_items: " {{users_with_items }}"
```

Hands-On Demonstration

Loops - 4

- Using Loops to create folders:

```
$ cat loopcomdir.yaml
- hosts: localhost
  become: yes
  become_user: osgdev

  vars:
    users_with_items:
      - name: "testuser5"
        personal_directories:
        - "tu05"
      - name: "testuser6"
        personal_directories:
        - "tu06"

    common_directories:
      - ".ssh"
      - "loops"
```

```
tasks:
  - name: Create common directories
    file:
      dest: "/home/{{ item.0.name }}/{{ item.1 }}"
      owner: "{{ item.0.name }}"
      group: "{{ item.0.name }}"
      state: directory
    with_nested:
      - "{{ users_with_items }}"
      - "{{ common_directories }}
```

Hands-On Demonstration

Loops - 5

- To Create Folder

```
$ cat dir1.yaml
- hosts: localhost

  tasks:
  - name: To Create a folder
    file:
      path: "/home/osgdev/ansilab/trialdir/trial"
      owner: osgdev
      group: docker
      state: directory
      mode: 0755
```

Hands-On Demonstration

Loops - 5

- Using Loops to create multiple folders:

```
$ cat dir2.yaml
- hosts: localhost

  tasks:
  - name: To Create a folder
    file:
      path: "/home/osgdev/ansilab/trialdir/{{ item }}"
      owner: osgdev
      group: docker
      state: directory
      mode: 0755
    with_items:
    - trial1
    - trial2
```

Hands-On Demonstration

Nested Loops

- Using Nested Loops to create multiple folders:

```
$ cat dir4.yaml
- hosts: localhost

  tasks:
  - name: To Create a folder
    file:
      path: "/home/osgdev/ansilab/trialdir/{{ item[0] }}/{{ item[1] }}"
      owner: osgdev
      group: docker
      state: touch
      mode: 0755
    with_nested:
      - ['trial1', 'trial2']
      - ['file1', 'file2', 'file3']
```

Hands-On Demonstration

Blocks:

- Normal Playbook with folder and file creation:

```
$ cat block.yaml
- hosts: localhost

  vars:
    folder_path: /home/osgdev/ansilab/block

  tasks:
  - name: To Create a folder
    file:
      path: "{{folder_path}}/NEWSAMPLE2"
      owner: osgdev
      group: docker
      state: directory
      mode: 0755
```

```
  - name: To Create a file
    file:
      path: "{{folder_path}}/NEWSAMPLE2/new_file2"
      state: touch
```

Hands-On Demonstration

Blocks:

- Bring them into a block

```
$ cat block1.yaml
- hosts: localhost

  vars:
    folder_path: /home/osgdev/ansilab/block

  tasks:
  - block:
    - name: To Create a folder
      file:
        path: "{{folder_path}}/NEWSAMPLE2"
        owner: osgdev
        group: docker
        state: directory
        mode: 0755
```

```
- name: To Create a file
    file:
      path: "{{folder_path}}/NEWSAMPLE2/new_file2"
      state: touch
```

Hands-On Demonstration

Error Handling in Blocks:

- Error Handling in Blocks:
- $ cat block2.yaml

```
- hosts: localhost

  vars:
    folder_path: /home/osgdev/ansilab/block

  tasks:
  - name: Folder and File Creation
    block:
    - name: Debug activity
      debug: msg='working on creating folder and file'
```

Hands-On Demonstration

Error Handling in Blocks:

```
- name: To Create a folder
  file:
    path: "{{folder_path}}/NEWSAMPLE2"
    owner: osgdev
    group: docker
    state: directory
    mode: 0755

- name: To Create a file
  file:
    path: "{{folder_path}}/NEWSAMPLE1/new_file2"
    state: touch
```

Hands-On Demonstration

Error Handling in Blocks:

```
rescue:
- name: Need to correct error
  debug: msg='Caught an error'

- name: To Create a file
  file:
    path: "{{folder_path}}/NEWSAMPLE2/new_file2"
    state: touch

- name: Error corrected
  debug: msg='Error Rectified'

always:
- name: Executing always
  debug: msg='Happy ending'
```

Hands-On Demonstration

Vaults

- Creating Encrypted Files

```
$ ansible-vault create foo.yml
$ cat foo.yml
```

- Editing Encrypted Files:

```
$ ansible-vault edit foo.yml
```

- Encrypting Unencrypted Files

```
$ cat info.yml
- information
  name: important
  content: 100
$ ansible-vault encrypt info.yml
$ cat info.yml
```

Hands-On Demonstration

Vaults

- Decrypting Encrypted Files

```
$ ansible-vault decrypt info.yml
$ cat info.yml
- information
  name: important
  content: 100
```

- Encrypt the file again and view it with password

```
$ ansible-vault encrypt foo.yml

$ cat foo.yml

$ ansible-vault view foo.yml
```

- Changing the passwords

```
$ ansible-vault rekey foo.yml
```