Configuration Management. Lection 3.

ANSIBLE and more ansible

- Roles
- Troubleshooting
- Vault

ANSIBLE. Roles

Roles provide a framework for fully independent, or interdependent collections of variables, tasks, files, templates, and modules.

In Ansible, the ***role is the primary mechanism for breaking a playbook into multiple files***. This simplifies writing complex playbooks, and it makes them easier to reuse. The breaking of playbook allows you to logically break the playbook into reusable components.

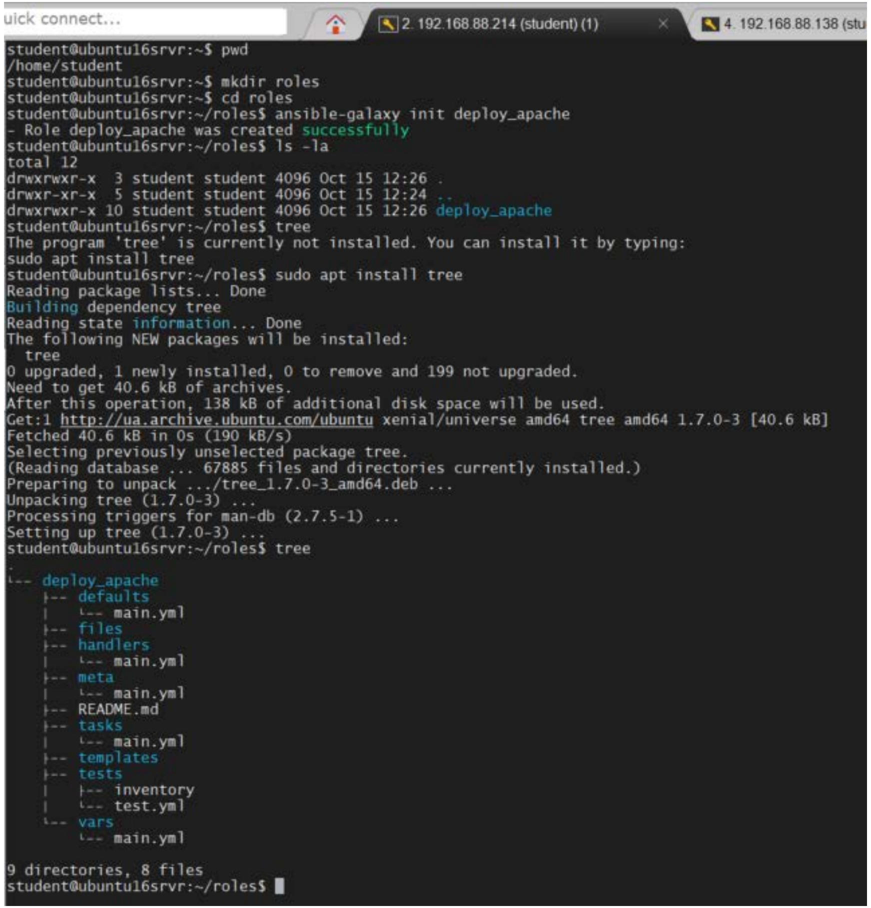
Each role is basically limited to a particular functionality or desired output, with all the necessary steps to provide that result either within that role itself or in other roles listed as dependencies.

Roles are not playbooks. Roles are small functionality which can be independently used but have to be used within playbooks. There is no way to directly execute a role. Roles have no explicit setting for which host the role will apply to.

Top-level playbooks are the bridge holding the hosts from your inventory file to roles that should be applied to those hosts.

https://www.tutorialspoint.com/ansible/ansible_quick_guide.htm

ANSIBLE. Roles

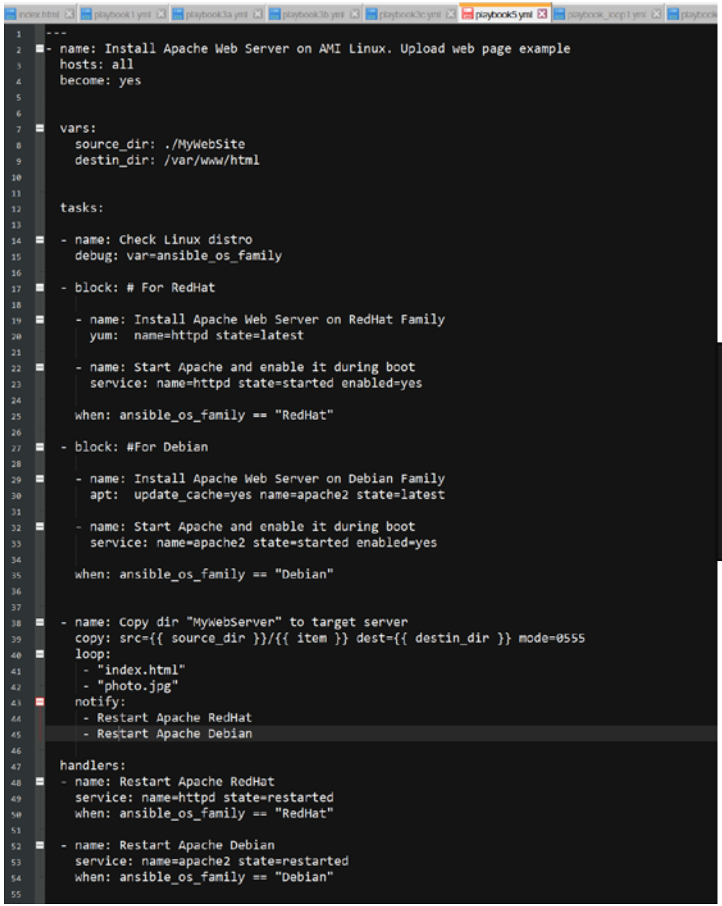


```
student@ubuntu16srvr:~$ pwd
/home/student
student@ubuntu16srvr:~$ mkdir roles
student@ubuntu16srvr:~$ cd roles
student@ubuntu16srvr:~/roles$ ansible-galaxy init deploy_apache
- Role deploy_apache was created successfully
student@ubuntu16srvr:~/roles$ ls -la
total 12
drwxrwxr-x  3 student student 4096 Oct 15 12:26 .
drwxr-xr-x  5 student student 4096 Oct 15 12:24 ..
drwxrwxr-x 10 student student 4096 Oct 15 12:26 deploy_apache
student@ubuntu16srvr:~/roles$ tree
The program 'tree' is currently not installed. You can install it by typing:
sudo apt install tree
student@ubuntu16srvr:~/roles$ sudo apt install tree
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following NEW packages will be installed:
  tree
0 upgraded, 1 newly installed, 0 to remove and 199 not upgraded.
Need to get 40.6 kB of archives.
After this operation, 138 kB of additional disk space will be used.
Get:1 http://ua.archive.ubuntu.com/ubuntu xenial/universe amd64 tree amd64 1.7.0-3 [40.6 kB]
Fetched 40.6 kB in 0s (190 kB/s)
Selecting previously unselected package tree.
(Reading database ... 67885 files and directories currently installed.)
Preparing to unpack .../tree_1.7.0-3_amd64.deb ...
Unpacking tree (1.7.0-3) ...
Processing triggers for man-db (2.7.5-1) ...
Setting up tree (1.7.0-3) ...
student@ubuntu16srvr:~/roles$ tree
.
└── deploy_apache
    ├── defaults
    │   └── main.yml
    ├── files
    ├── handlers
    │   └── main.yml
    ├── meta
    │   └── main.yml
    ├── README.md
    ├── tasks
    │   └── main.yml
    ├── templates
    ├── tests
    │   ├── inventory
    │   └── test.yml
    └── vars
        └── main.yml

9 directories, 8 files
student@ubuntu16srvr:~/roles$
```

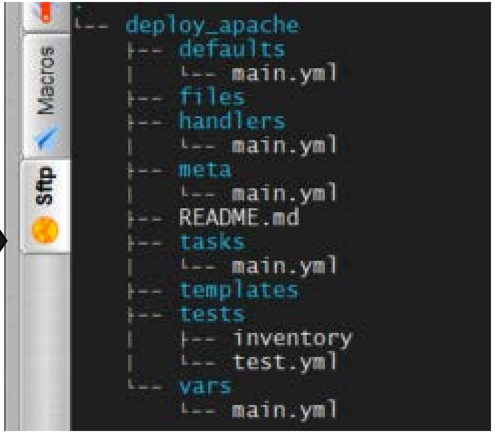
We have got the clean directory structure with the ***ansible-galaxy*** command. Each directory must contain a ***main.yml*** file, which contains the relevant content.

tasks - contains the main list of tasks to be executed by the role.
handlers - contains handlers, which may be used by this role or even anywhere outside this role.
defaults - default variables for the role.
vars - other variables for the role. Vars has the higher priority than defaults.
files - contains files required to transfer or deployed to the target machines via this role.
templates - contains templates which can be deployed via this role.
meta - defines some data / information about this role (author, dependency, versions, examples, etc,.)

ANSIBLE. Roles

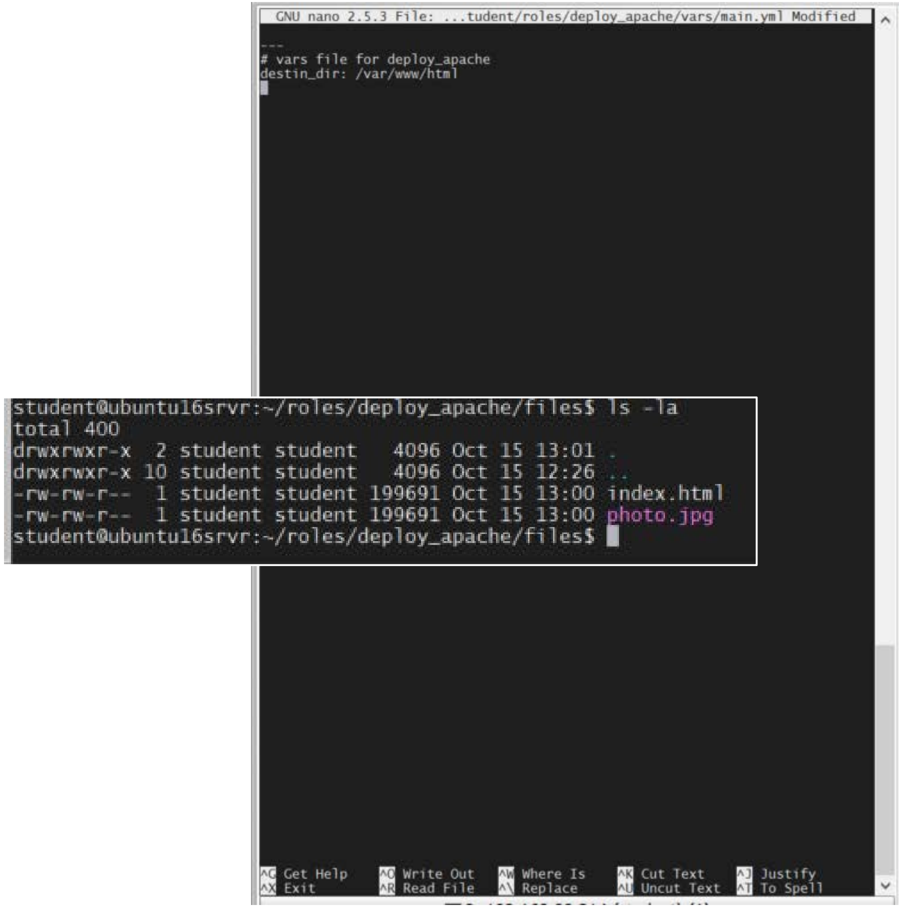


```
---
- name: Install Apache Web Server on AMI Linux. Upload web page example
  hosts: all
  become: yes


  vars:
    source_dir: ./MyWebSite
    destin_dir: /var/www/html


  tasks:

  - name: Check Linux distro
    debug: var=ansible_os_family

  - block: # For RedHat

      - name: Install Apache Web Server on RedHat Family
        yum:  name=httpd state=latest

      - name: Start Apache and enable it during boot
        service: name=httpd state=started enabled=yes

    when: ansible_os_family == "RedHat"

  - block: #For Debian

      - name: Install Apache Web Server on Debian Family
        apt:  update_cache=yes name=apache2 state=latest

      - name: Start Apache and enable it during boot
        service: name=apache2 state=started enabled=yes

    when: ansible_os_family == "Debian"


  - name: Copy dir "MyWebServer" to target server
    copy: src={{ source_dir }}/{{ item }} dest={{ destin_dir }} mode=0555
    loop:
     - "index.html"
     - "photo.jpg"
    notify:
     - Restart Apache RedHat
     - Restart Apache Debian

  handlers:
  - name: Restart Apache RedHat
    service: name=httpd state=restarted
    when: ansible_os_family == "RedHat"

  - name: Restart Apache Debian
    service: name=apache2 state=restarted
    when: ansible_os_family == "Debian"

```



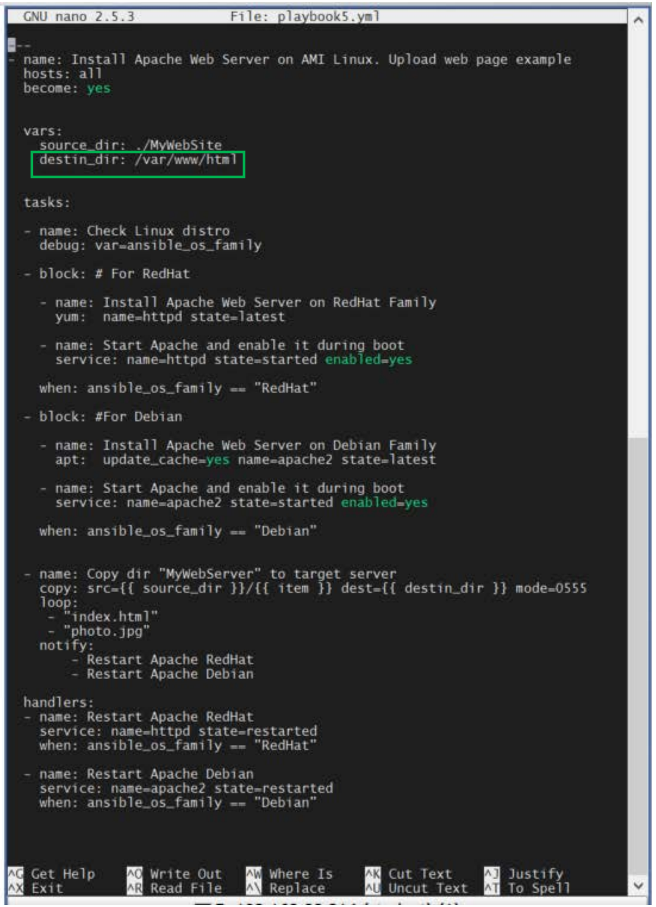
```
.
└── deploy_apache
    ├── defaults
    │   └── main.yml
    ├── files
    ├── handlers
    │   └── main.yml
    ├── meta
    │   └── main.yml
    ├── README.md
    ├── tasks
    │   └── main.yml
    ├── templates
    ├── tests
    │   ├── inventory
    │   └── test.yml
    └── vars
        └── main.yml
```

ANSIBLE. Roles



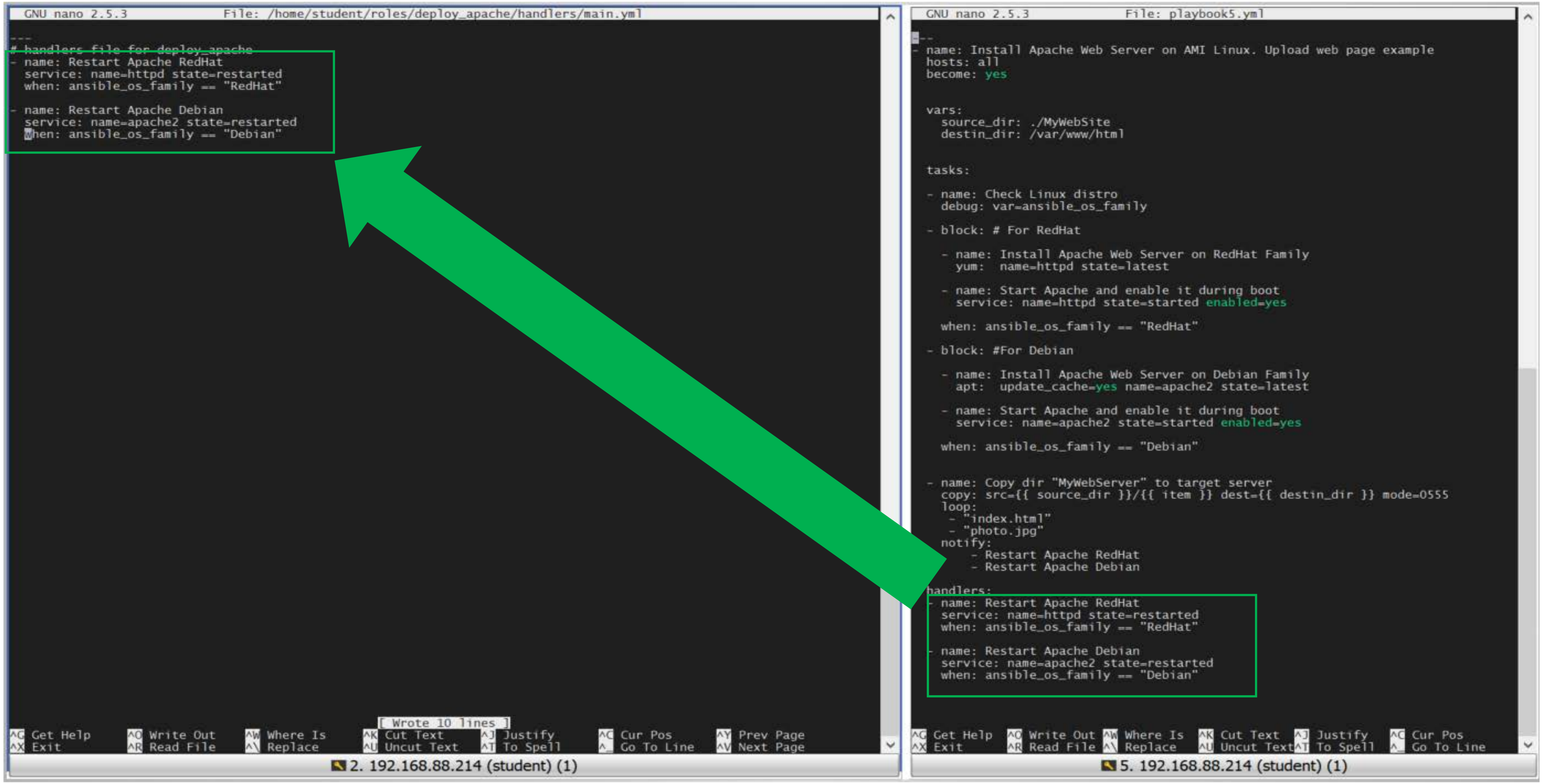
```
---
# vars file for deploy_apache
destin_dir: /var/www/html
```

```
student@ubuntu16srvr:~/roles/deploy_apache/files$ ls -la
total 400
drwxrwxr-x  2 student student   4096 Oct 15 13:01 .
drwxrwxr-x 10 student student   4096 Oct 15 12:26 ..
-rw-rw-r--  1 student student 199691 Oct 15 13:00 index.html
-rw-rw-r--  1 student student 199691 Oct 15 13:00 photo.jpg
student@ubuntu16srvr:~/roles/deploy_apache/files$
```



```
---
- name: Install Apache Web Server on AMI Linux. Upload web page example
  hosts: all
  become: yes


  vars:
    source_dir: ./MyWebSite
    destin_dir: /var/www/html


  tasks:

  - name: Check Linux distro
    debug: var=ansible_os_family

  - block: # For RedHat

    - name: Install Apache Web Server on RedHat Family
      yum:  name=httpd state=latest

    - name: Start Apache and enable it during boot
      service: name=httpd state=started enabled=yes

    when: ansible_os_family == "RedHat"

  - block: #For Debian

    - name: Install Apache Web Server on Debian Family
      apt:  update_cache=yes name=apache2 state=latest

    - name: Start Apache and enable it during boot
      service: name=apache2 state=started enabled=yes

    when: ansible_os_family == "Debian"


  - name: Copy dir "MyWebServer" to target server
    copy: src={{ source_dir }}/{{ item }} dest={{ destin_dir }} mode=0555
    loop:
     - "index.html"
     - "photo.jpg"
    notify:
        - Restart Apache RedHat
        - Restart Apache Debian

  handlers:
  - name: Restart Apache RedHat
    service: name=httpd state=restarted
    when: ansible_os_family == "RedHat"

  - name: Restart Apache Debian
    service: name=apache2 state=restarted
    when: ansible_os_family == "Debian"
```

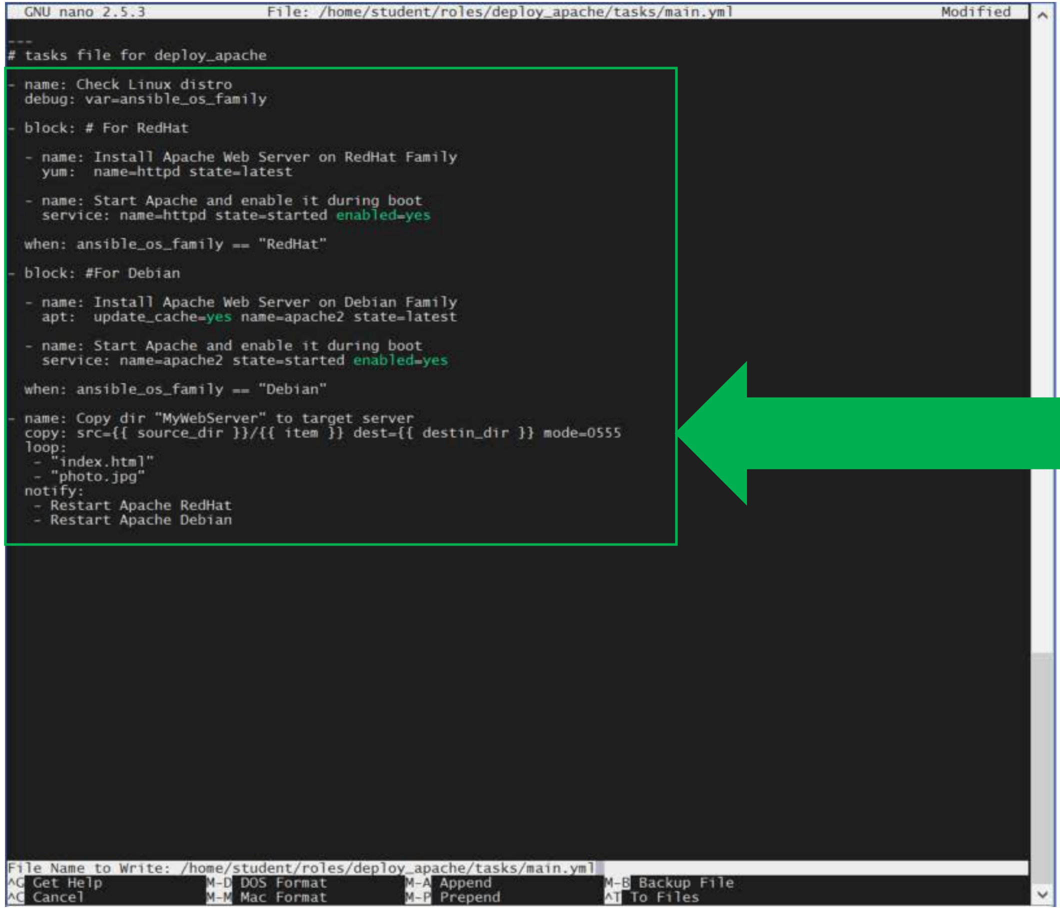
ANSIBLE. Roles



```
GNU nano 2.5.3              File: /home/student/roles/deploy_apache/handlers/main.yml

---
# handlers file for deploy_apache
- name: Restart Apache RedHat
  service: name=httpd state=restarted
  when: ansible_os_family == "RedHat"

- name: Restart Apache Debian
  service: name=apache2 state=restarted
  when: ansible_os_family == "Debian"

                              [ Wrote 10 lines ]
```

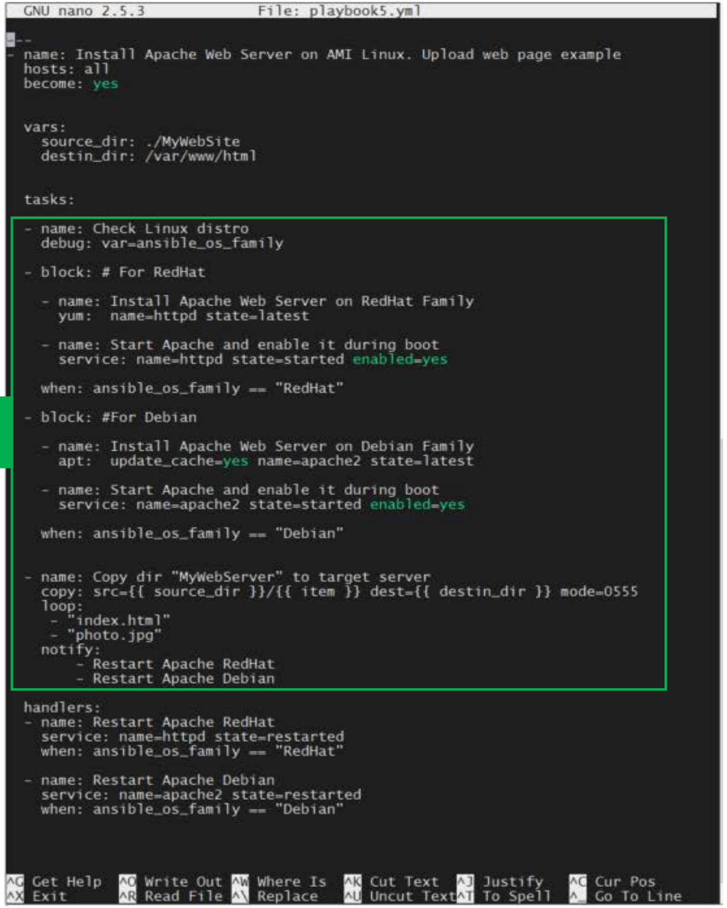
```
GNU nano 2.5.3              File: playbook5.yml

---
- name: Install Apache Web Server on AMI Linux. Upload web page example
  hosts: all
  become: yes


  vars:
    source_dir: ./MyWebSite
    destin_dir: /var/www/html


  tasks:

  - name: Check Linux distro
    debug: var=ansible_os_family

  - block: # For RedHat

    - name: Install Apache Web Server on RedHat Family
      yum:  name=httpd state=latest

    - name: Start Apache and enable it during boot
      service: name=httpd state=started enabled=yes

    when: ansible_os_family == "RedHat"

  - block: #For Debian

    - name: Install Apache Web Server on Debian Family
      apt:  update_cache=yes name=apache2 state=latest

    - name: Start Apache and enable it during boot
      service: name=apache2 state=started enabled=yes

    when: ansible_os_family == "Debian"


  - name: Copy dir "MyWebServer" to target server
    copy: src={{ source_dir }}/{{ item }} dest={{ destin_dir }} mode=0555
    loop:
     - "index.html"
     - "photo.jpg"
    notify:
        - Restart Apache RedHat
        - Restart Apache Debian

  handlers:
  - name: Restart Apache RedHat
    service: name=httpd state=restarted
    when: ansible_os_family == "RedHat"

  - name: Restart Apache Debian
    service: name=apache2 state=restarted
    when: ansible_os_family == "Debian"
```

ANSIBLE. Roles



```
  GNU nano 2.5.3          File: /home/student/roles/deploy_apache/tasks/main.yml          Modified

---
# tasks file for deploy_apache

- name: Check Linux distro
  debug: var=ansible_os_family

- block: # For RedHat

  - name: Install Apache Web Server on RedHat Family
    yum:  name=httpd state=latest

  - name: Start Apache and enable it during boot
    service: name=httpd state=started enabled=yes

  when: ansible_os_family == "RedHat"

- block: #For Debian

  - name: Install Apache Web Server on Debian Family
    apt:  update_cache=yes name=apache2 state=latest

  - name: Start Apache and enable it during boot
    service: name=apache2 state=started enabled=yes

  when: ansible_os_family == "Debian"

- name: Copy dir "MyWebServer" to target server
  copy: src={{ source_dir }}/{{ item }} dest={{ destin_dir }} mode=0555
  loop:
   - "index.html"
   - "photo.jpg"
  notify:
   - Restart Apache RedHat
   - Restart Apache Debian
```

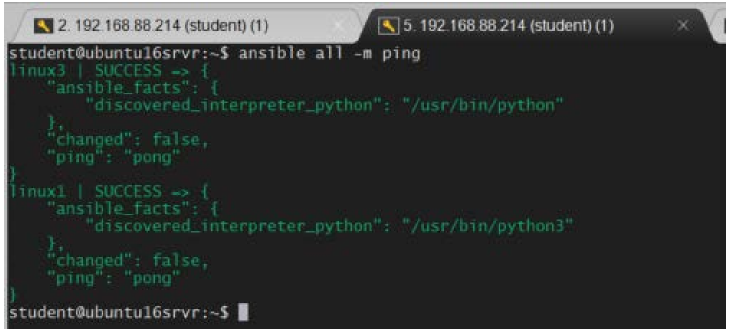
```
File Name to Write: /home/student/roles/deploy_apache/tasks/main.yml
^G Get Help          M-D DOS Format       M-A Append           M-B Backup File
^C Cancel            M-M Mac Format       M-P Prepend          ^T To Files
```



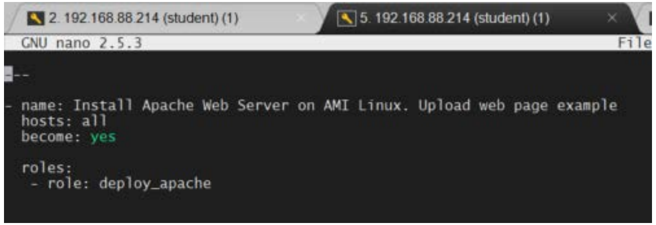
```
  GNU nano 2.5.3              File: playbook5.yml

---
- name: Install Apache Web Server on AMI Linux. Upload web page example
  hosts: all
  become: yes


  vars:
    source_dir: ./MyWebSite
    destin_dir: /var/www/html


  tasks:

  - name: Check Linux distro
    debug: var=ansible_os_family

  - block: # For RedHat

    - name: Install Apache Web Server on RedHat Family
      yum:  name=httpd state=latest

    - name: Start Apache and enable it during boot
      service: name=httpd state=started enabled=yes

    when: ansible_os_family == "RedHat"

  - block: #For Debian

    - name: Install Apache Web Server on Debian Family
      apt:  update_cache=yes name=apache2 state=latest

    - name: Start Apache and enable it during boot
      service: name=apache2 state=started enabled=yes

    when: ansible_os_family == "Debian"


  - name: Copy dir "MyWebServer" to target server
    copy: src={{ source_dir }}/{{ item }} dest={{ destin_dir }} mode=0555
    loop:
     - "index.html"
     - "photo.jpg"
    notify:
        - Restart Apache RedHat
        - Restart Apache Debian

  handlers:
  - name: Restart Apache RedHat
    service: name=httpd state=restarted
    when: ansible_os_family == "RedHat"

  - name: Restart Apache Debian
    service: name=apache2 state=restarted
    when: ansible_os_family == "Debian"
```

```
^G Get Help   ^O Write Out  ^W Where Is   ^K Cut Text   ^J Justify    ^C Cur Pos
^X Exit       ^R Read File  ^\ Replace    ^U Uncut Text ^T To Spell   ^_ Go To Line
```

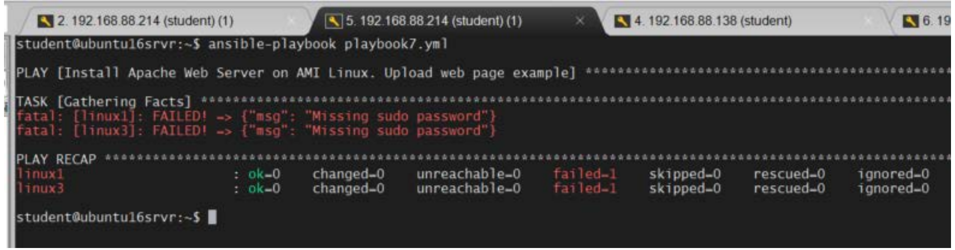
ANSIBLE. Roles


2. 192.168.88.214 (student) (1)
5. 192.168.88.214 (student) (1)

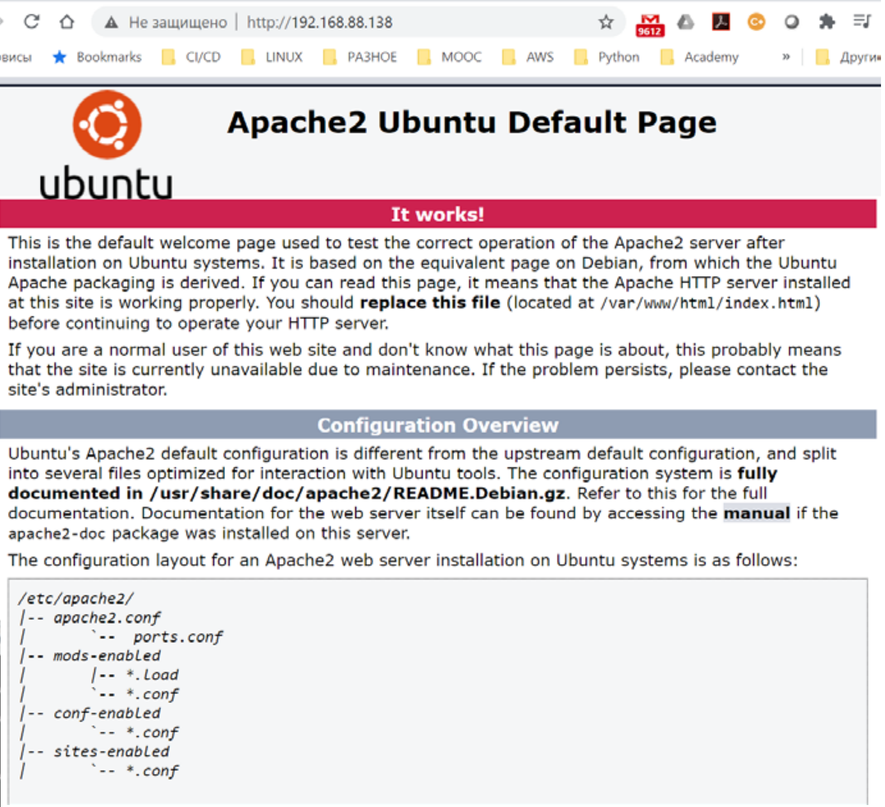
```
student@ubuntu16srvr:~$ ansible all -m ping
linux3 | SUCCESS => {
    "ansible_facts": {
        "discovered_interpreter_python": "/usr/bin/python"
    },
    "changed": false,
    "ping": "pong"
}
linux1 | SUCCESS => {
    "ansible_facts": {
        "discovered_interpreter_python": "/usr/bin/python3"
    },
    "changed": false,
    "ping": "pong"
}
student@ubuntu16srvr:~$
```


2. 192.168.88.214 (student) (1)
5. 192.168.88.214 (student) (1)
GNU nano 2.5.3
File

```
---

- name: Install Apache Web Server on AMI Linux. Upload web page example
  hosts: all
  become: yes

  roles:
   - role: deploy_apache
```


2. 192.168.88.214 (student) (1)
5. 192.168.88.214 (student) (1)
4. 192.168.88.138 (student)
6. 19

```
student@ubuntu16srvr:~$ ansible-playbook playbook7.yml

PLAY [Install Apache Web Server on AMI Linux. Upload web page example] ************************************************

TASK [Gathering Facts] ************************************************************************************************
fatal: [linux1]: FAILED! => {"msg": "Missing sudo password"}
fatal: [linux3]: FAILED! => {"msg": "Missing sudo password"}

PLAY RECAP ************************************************************************************************************
linux1                     : ok=0    changed=0    unreachable=0    failed=1    skipped=0    rescued=0    ignored=0
linux3                     : ok=0    changed=0    unreachable=0    failed=1    skipped=0    rescued=0    ignored=0

student@ubuntu16srvr:~$
```



Apache2 Ubuntu Default Page

It works!

This is the default welcome page used to test the correct operation of the Apache2 server after installation on Ubuntu systems. It is based on the equivalent page on Debian, from which the Ubuntu Apache packaging is derived. If you can read this page, it means that the Apache HTTP server installed at this site is working properly. You should **replace this file** (located at `/var/www/html/index.html`) before continuing to operate your HTTP server.

If you are a normal user of this web site and don't know what this page is about, this probably means that the site is currently unavailable due to maintenance. If the problem persists, please contact the site's administrator.

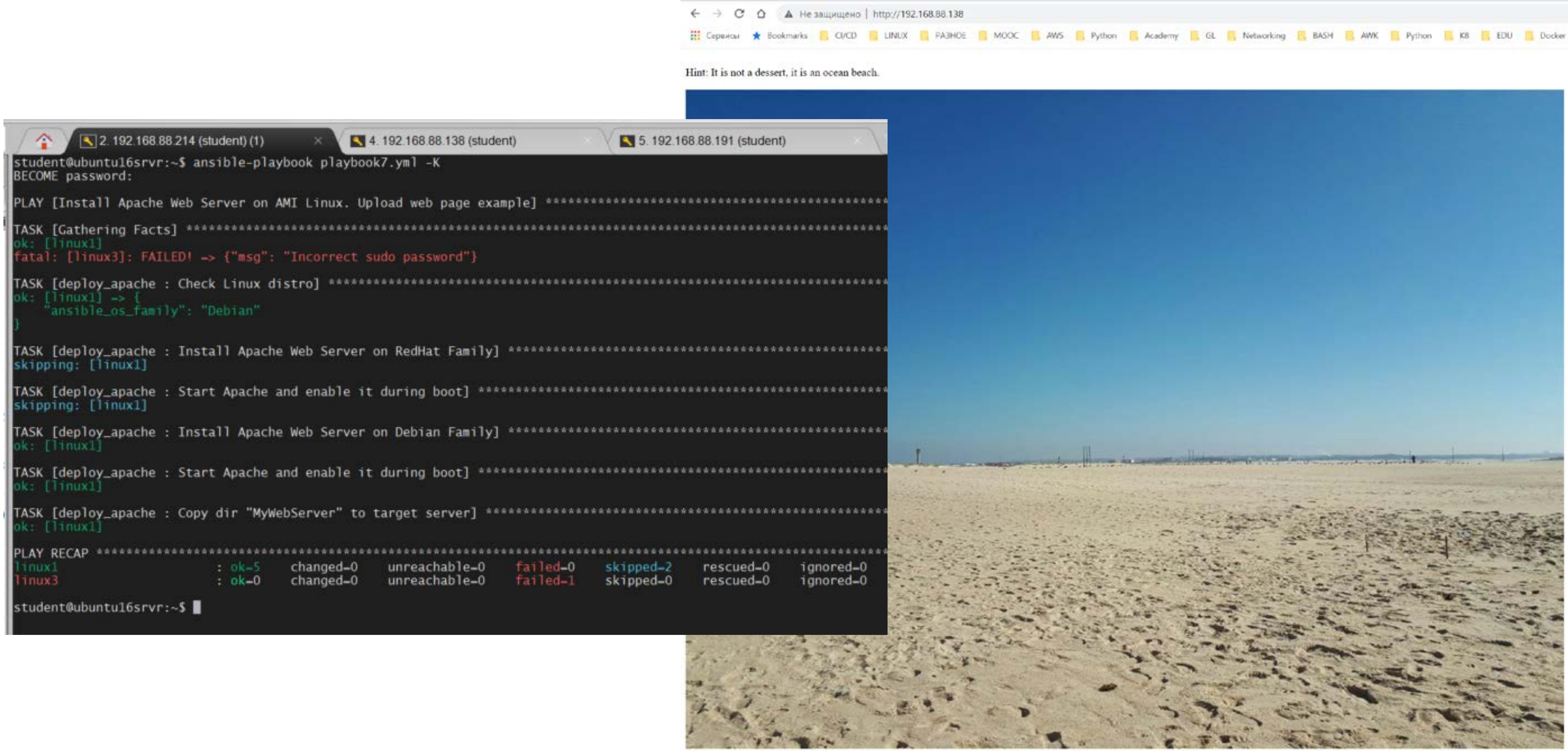
Configuration Overview

Ubuntu's Apache2 default configuration is different from the upstream default configuration, and split into several files optimized for interaction with Ubuntu tools. The configuration system is **fully documented in /usr/share/doc/apache2/README.Debian.gz**. Refer to this for the full documentation. Documentation for the web server itself can be found by accessing the **manual** if the apache2-doc package was installed on this server.

The configuration layout for an Apache2 web server installation on Ubuntu systems is as follows:

```
/etc/apache2/
|-- apache2.conf
|       `--  ports.conf
|-- mods-enabled
|       |-- *.load
|       `-- *.conf
|-- conf-enabled
|       `-- *.conf
|-- sites-enabled
|       `-- *.conf
```

ANSIBLE. Roles


2. 192.168.88.214 (student) (1)
4. 192.168.88.138 (student)
5. 192.168.88.191 (student)
Не защищено | http://192.168.88.138
Сервисы
Bookmarks
CI/CD
LINUX
РАЗНОЕ
MOOC
AWS
Python
Academy
GL
Networking
BASH
AWK
Python
K8
EDU
Docker

```
student@ubuntu16srvr:~$ ansible-playbook playbook7.yml -K
BECOME password:

PLAY [Install Apache Web Server on AMI Linux. Upload web page example] *******************************************

TASK [Gathering Facts] ********************************************************************************************
ok: [linux1]
fatal: [linux3]: FAILED! => {"msg": "Incorrect sudo password"}

TASK [deploy_apache : Check Linux distro] *************************************************************************
ok: [linux1] => {
    "ansible_os_family": "Debian"
}

TASK [deploy_apache : Install Apache Web Server on RedHat Family] *************************************************
skipping: [linux1]

TASK [deploy_apache : Start Apache and enable it during boot] *****************************************************
skipping: [linux1]

TASK [deploy_apache : Install Apache Web Server on Debian Family] *************************************************
ok: [linux1]

TASK [deploy_apache : Start Apache and enable it during boot] *****************************************************
ok: [linux1]

TASK [deploy_apache : Copy dir "MyWebServer" to target server] ****************************************************
ok: [linux1]

PLAY RECAP ********************************************************************************************************
linux1                     : ok=5    changed=0    unreachable=0    failed=0    skipped=2    rescued=0    ignored=0
linux3                     : ok=0    changed=0    unreachable=0    failed=1    skipped=0    rescued=0    ignored=0

student@ubuntu16srvr:~$
```

Hint: It is not a dessert, it is an ocean beach.

ANSIBLE. Troubleshooting

The most common strategies for debugging Ansible playbooks are using the modules given below:

Debug and ***Register***

These two are the modules available in Ansible. For debugging purpose, we need to use the two modules judiciously.

Use ***Verbosity***

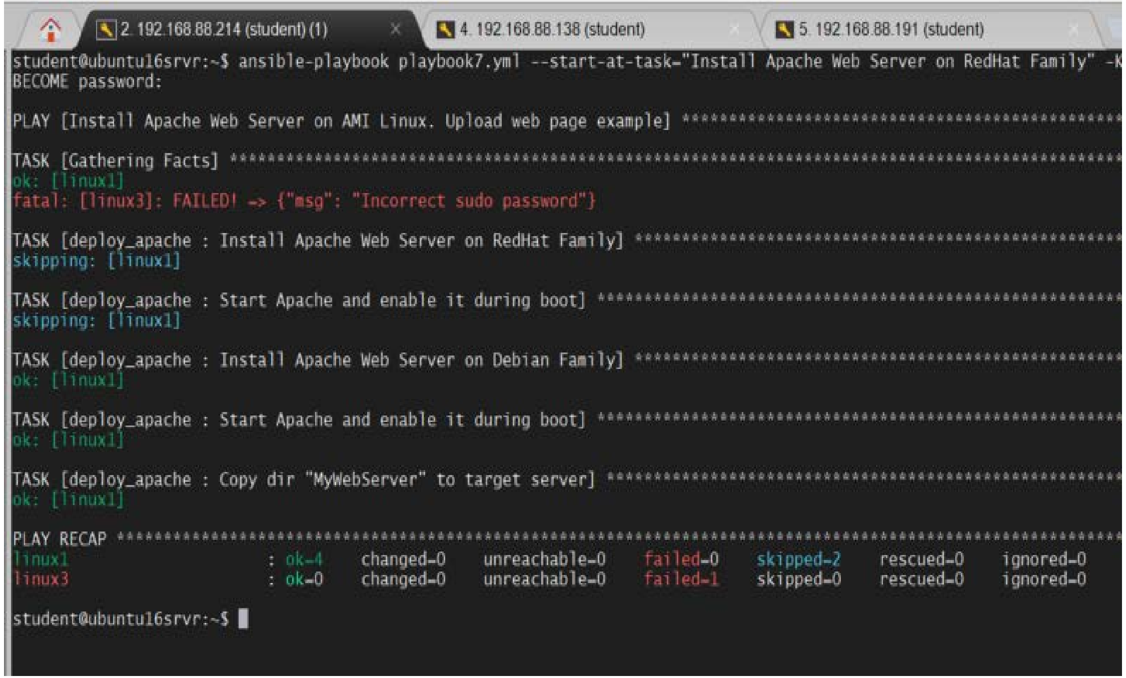
With the Ansible command, one can provide the verbosity level. You can run the commands with verbosity level one (-v) or two (-vv).

ANSIBLE. Troubleshooting

When you are testing new plays or debugging playbooks, you may need to run the same play multiple times. To make this more efficient, Ansible offers two alternative ways to execute a playbook: ***start-at-task*** and ***step mode.***

start-at-task

To start executing your playbook at a particular task (usually the task that failed on the previous run), use the ***--start-at-task*** option:

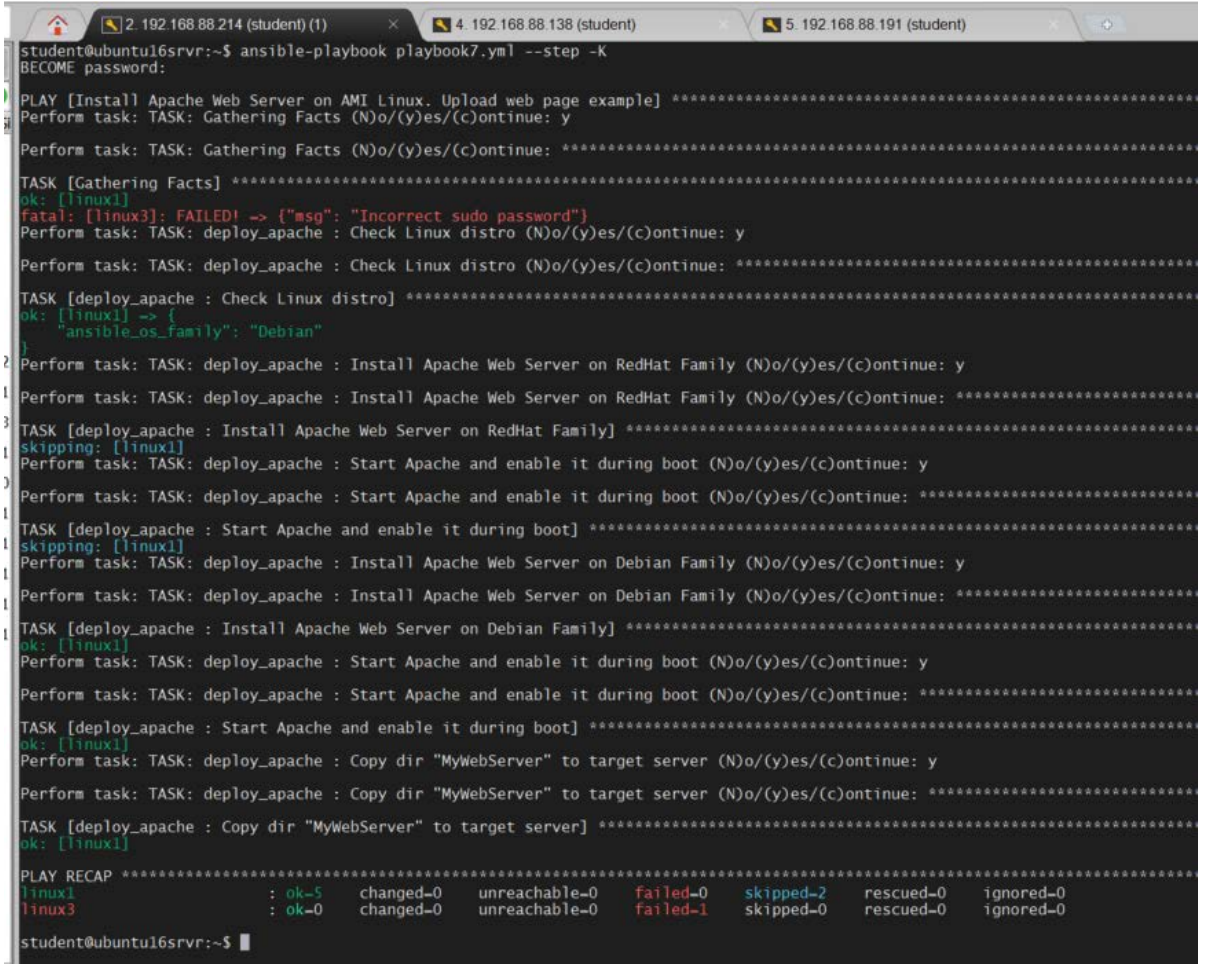

2. 192.168.88.214 (student) (1)
4. 192.168.88.138 (student)
5. 192.168.88.191 (student)

```
student@ubuntu16srvr:~$ ansible-playbook playbook7.yml --start-at-task="Install Apache Web Server on RedHat Family" -K
BECOME password:

PLAY [Install Apache Web Server on AMI Linux. Upload web page example] ************************************************

TASK [Gathering Facts] ************************************************************************************************
ok: [linux1]
fatal: [linux3]: FAILED! => {"msg": "Incorrect sudo password"}

TASK [deploy_apache : Install Apache Web Server on RedHat Family] *****************************************************
skipping: [linux1]

TASK [deploy_apache : Start Apache and enable it during boot] *********************************************************
skipping: [linux1]

TASK [deploy_apache : Install Apache Web Server on Debian Family] *****************************************************
ok: [linux1]

TASK [deploy_apache : Start Apache and enable it during boot] *********************************************************
ok: [linux1]

TASK [deploy_apache : Copy dir "MyWebServer" to target server] ********************************************************
ok: [linux1]

PLAY RECAP ************************************************************************************************************
linux1                     : ok=4    changed=0    unreachable=0    failed=0    skipped=2    rescued=0    ignored=0
linux3                     : ok=0    changed=0    unreachable=0    failed=1    skipped=0    rescued=0    ignored=0

student@ubuntu16srvr:~$ █
```

ANSIBLE. Troubleshooting

Step mode

To execute a playbook interactively, use ***--step***:


2. 192.168.88.214 (student) (1)
4. 192.168.88.138 (student)
5. 192.168.88.191 (student)

```
student@ubuntu16srvr:~$ ansible-playbook playbook7.yml --step -K
BECOME password:

PLAY [Install Apache Web Server on AMI Linux. Upload web page example] ***********************************************
Perform task: TASK: Gathering Facts (N)o/(y)es/(c)ontinue: y

Perform task: TASK: Gathering Facts (N)o/(y)es/(c)ontinue: ******************************************************************

TASK [Gathering Facts] ************************************************************************************************************
ok: [linux1]
fatal: [linux3]: FAILED! => {"msg": "Incorrect sudo password"}
Perform task: TASK: deploy_apache : Check Linux distro (N)o/(y)es/(c)ontinue: y

Perform task: TASK: deploy_apache : Check Linux distro (N)o/(y)es/(c)ontinue: ************************************************

TASK [deploy_apache : Check Linux distro] *****************************************************************************************
ok: [linux1] => {
    "ansible_os_family": "Debian"
}
Perform task: TASK: deploy_apache : Install Apache Web Server on RedHat Family (N)o/(y)es/(c)ontinue: y

Perform task: TASK: deploy_apache : Install Apache Web Server on RedHat Family (N)o/(y)es/(c)ontinue: *************************

TASK [deploy_apache : Install Apache Web Server on RedHat Family] *****************************************************************
skipping: [linux1]
Perform task: TASK: deploy_apache : Start Apache and enable it during boot (N)o/(y)es/(c)ontinue: y

Perform task: TASK: deploy_apache : Start Apache and enable it during boot (N)o/(y)es/(c)ontinue: *****************************

TASK [deploy_apache : Start Apache and enable it during boot] *********************************************************************
skipping: [linux1]
Perform task: TASK: deploy_apache : Install Apache Web Server on Debian Family (N)o/(y)es/(c)ontinue: y

Perform task: TASK: deploy_apache : Install Apache Web Server on Debian Family (N)o/(y)es/(c)ontinue: *************************

TASK [deploy_apache : Install Apache Web Server on Debian Family] *****************************************************************
ok: [linux1]
Perform task: TASK: deploy_apache : Start Apache and enable it during boot (N)o/(y)es/(c)ontinue: y

Perform task: TASK: deploy_apache : Start Apache and enable it during boot (N)o/(y)es/(c)ontinue: *****************************

TASK [deploy_apache : Start Apache and enable it during boot] *********************************************************************
ok: [linux1]
Perform task: TASK: deploy_apache : Copy dir "MyWebServer" to target server (N)o/(y)es/(c)ontinue: y

Perform task: TASK: deploy_apache : Copy dir "MyWebServer" to target server (N)o/(y)es/(c)ontinue: ****************************

TASK [deploy_apache : Copy dir "MyWebServer" to target server] ********************************************************************
ok: [linux1]

PLAY RECAP ************************************************************************************************************************
linux1                     : ok=5    changed=0    unreachable=0    failed=0    skipped=2    rescued=0    ignored=0
linux3                     : ok=0    changed=0    unreachable=0    failed=1    skipped=0    rescued=0    ignored=0

student@ubuntu16srvr:~$ █
```

ANSIBLE. Vault

Encrypting content with Ansible Vault

Ansible Vault encrypts variables and files so you can protect sensitive content such as passwords or keys rather than leaving it visible as plaintext in playbooks or roles.
To use Ansible Vault you need one or more passwords to encrypt and decrypt content. If you store your vault passwords in a third-party tool such as a secret manager, you need a script to access them.
Use the passwords with the ***ansible-vault*** command-line tool to create and view encrypted variables, ***create*** encrypted files, ***encrypt*** existing files, or ***edit***, ***re-key***, or ***decrypt*** files. You can then place encrypted content under source control and share it more safely.

ANSIBLE. Vault

Managing vault passwords

Managing your encrypted content is easier if you develop a strategy for managing your vault passwords. A vault password can be any string you choose. There is no special command to create a vault password. However, you need to keep track of your vault passwords. Each time you encrypt a variable or file with ***Ansible Vault***, you must provide a password. When you use an encrypted variable or file in a command or playbook, you must provide the same password that was used to encrypt it. To develop a strategy for managing vault passwords, start with two questions:

Do you want to encrypt all your content with the same password, or use different passwords for different needs?

Where do you want to store your password or passwords?

ANSIBLE. Vault

Choosing between a single password and multiple passwords
If you have a small team or few sensitive values, you can use a single password for everything you encrypt with Ansible Vault. Store your vault password securely in a file or a secret manager as described below.

If you have a larger team or many sensitive values, you can use multiple passwords. For example, you can use different passwords for different users or different levels of access. Depending on your needs, you might want a different password for each encrypted file, for each directory, or for each environment. For example, you might have a playbook that includes two vars files, one for the dev environment and one for the production environment, encrypted with two different passwords. When you run the playbook, select the correct vault password for the environment you are targeting, using a vault ID.

Managing multiple passwords with vault IDs
If you use multiple vault passwords, you can differentiate one password from another with vault IDs. You use the vault ID in three ways:

- Pass it with **--vault-id** to the ansible-vault command when you create encrypted content
- Include it wherever you store the password for that vault ID (see Storing and accessing vault passwords)
- Pass it with **--vault-id** to the ansible-playbook command when you run a playbook that uses content you encrypted with that vault ID

ANSIBLE. Vault

Once you have a strategy for managing and storing vault passwords, you can start encrypting content. You can encrypt two types of content with Ansible Vault: variables and files. Encrypted content always includes the !vault tag, which tells Ansible and YAML that the content needs to be decrypted, and a | character, which allows multi-line strings. Encrypted content created with --vault-id also contains the vault ID label. This table shows the main differences between encrypted variables and encrypted files.

	Encrypted variables	**Encrypted files**
How much is encrypted?	Variables within a plaintext file	The entire file
When is it decrypted?	On demand, only when needed	Whenever loaded or referenced
What can be encrypted?	Only variables	Any structured data file

Q&A

Thank you!