ATTACK DEFENSE

by PentesterAcademy

Name	Ansible: Deploying Apache and MySQL
URL	https://www.attackdefense.com/challengedetails?cid=2042
Type	DevOps Basics: Infrastructure as Code

Important Note: This document illustrates all the important steps required to complete this lab. This is by no means a comprehensive step-by-step solution for this exercise. This is only provided as a reference to various commands needed to complete this exercise and for your further research on this topic. Also, note that the IP addresses and domain names might be different in your lab.

Challenge Description

Ansible is open-source software for storing provisioning, configuration management, and application-deployment infrastructure as code (IaC).

A Kali CLI machine (kali-cli) is provided to the user with Ansible installed on it and a configuration file for installing Apache2 web server and MySQL is also provided. The tar archives for installation are stored in /tmp directory. A Ubuntu server (test-server) is also present on the same network. The credentials for this machine are:

Username	Password
ubuntu	password1

The multiview interface provides access to the Kali CLI interface and port 80 of the Ubuntu server.

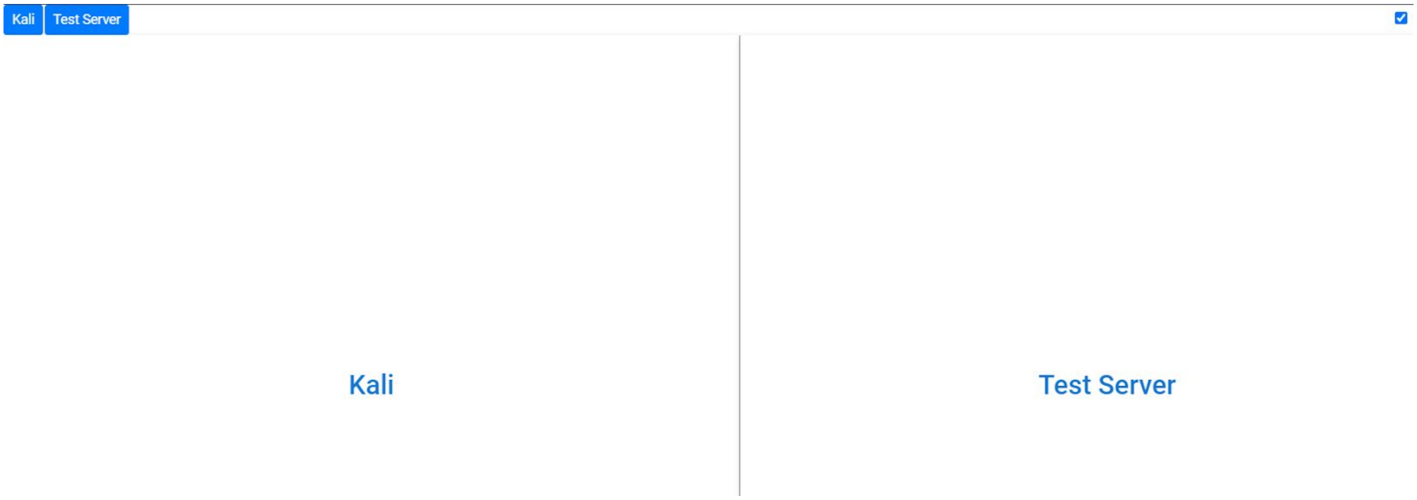
Objective: Deploy Apache and MySQL on the Ubuntu server machine using Ansible!

Instructions:

- The Ubuntu server is reachable by name "test-server"

Solution

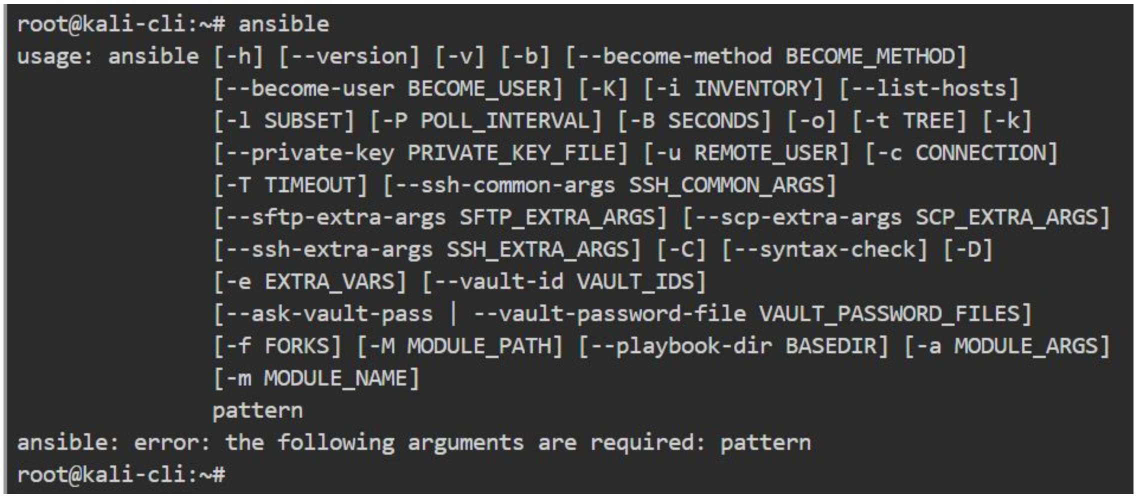
Step 1: The multi-view interface should appear like this.



As marked in the above image, we have a Kali CLI terminal and port 80 of Ubuntu server. Ubuntu server doesn’t have an Apache web server installed on it as of now, once that is installed on it, the iframe will change.

Step 2: Ensure that ansible is installed on the Kali machine.

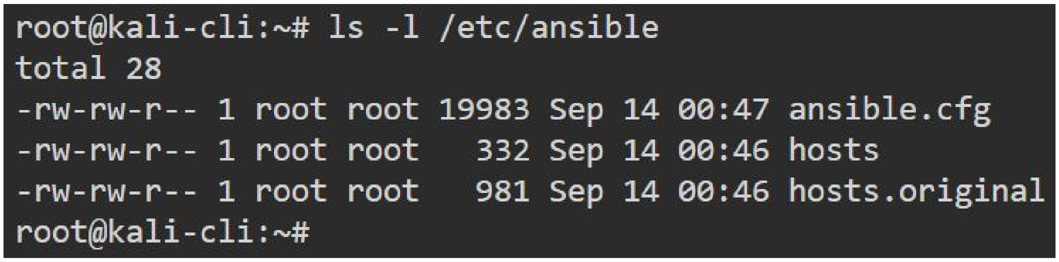
Command: ansible



```
root@kali-cli:~# ansible
usage: ansible [-h] [--version] [-v] [-b] [--become-method BECOME_METHOD]
               [--become-user BECOME_USER] [-K] [-i INVENTORY] [--list-hosts]
               [-l SUBSET] [-P POLL_INTERVAL] [-B SECONDS] [-o] [-t TREE] [-k]
               [--private-key PRIVATE_KEY_FILE] [-u REMOTE_USER] [-c CONNECTION]
               [-T TIMEOUT] [--ssh-common-args SSH_COMMON_ARGS]
               [--sftp-extra-args SFTP_EXTRA_ARGS] [--scp-extra-args SCP_EXTRA_ARGS]
               [--ssh-extra-args SSH_EXTRA_ARGS] [-C] [--syntax-check] [-D]
               [-e EXTRA_VARS] [--vault-id VAULT_IDS]
               [--ask-vault-pass | --vault-password-file VAULT_PASSWORD_FILES]
               [-f FORKS] [-M MODULE_PATH] [--playbook-dir BASEDIR] [-a MODULE_ARGS]
               [-m MODULE_NAME]
               pattern
ansible: error: the following arguments are required: pattern
root@kali-cli:~#
```

Step 3: Check Ansible configuration files.

Command: ls -l /etc/ansible



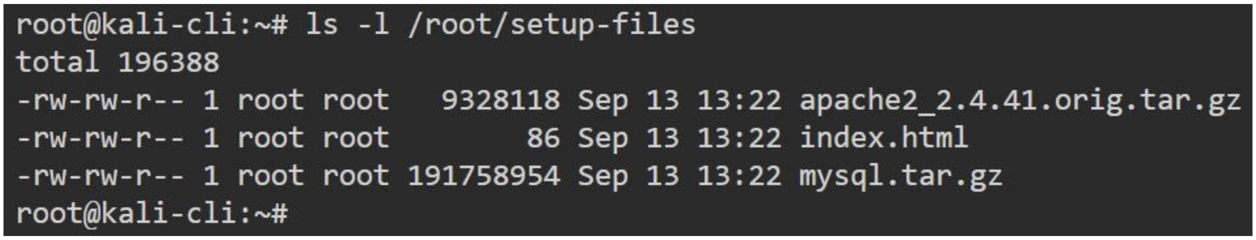
```
root@kali-cli:~# ls -l /etc/ansible
total 28
-rw-rw-r-- 1 root root 19983 Sep 14 00:47 ansible.cfg
-rw-rw-r-- 1 root root   332 Sep 14 00:46 hosts
-rw-rw-r-- 1 root root   981 Sep 14 00:46 hosts.original
root@kali-cli:~#
```

Here,

ansible.cfg	:	Ansible configuration file
hosts	:	list of remote machine to be configured using Ansible
hosts.original	:	Template hosts files

Step 4: Check archives and files provided for installation.

Command: ls -l /root/setup-files



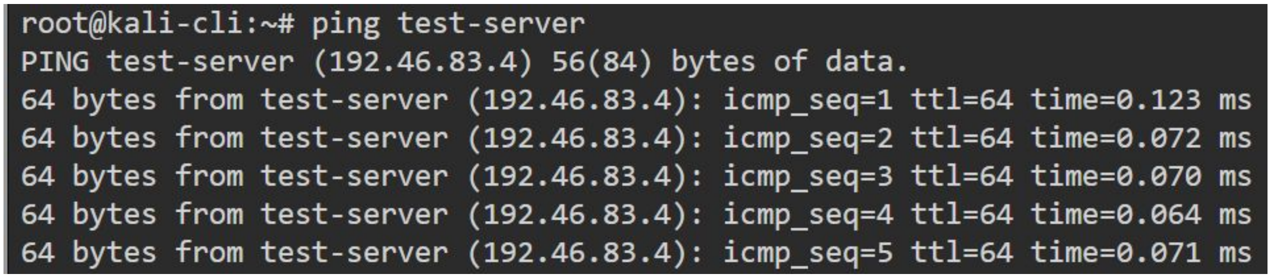
```
root@kali-cli:~# ls -l /root/setup-files
total 196388
-rw-rw-r-- 1 root root   9328118 Sep 13 13:22 apache2_2.4.41.orig.tar.gz
-rw-rw-r-- 1 root root        86 Sep 13 13:22 index.html
-rw-rw-r-- 1 root root 191758954 Sep 13 13:22 mysql.tar.gz
root@kali-cli:~#
```

Here,

apache2_2.4.41.orig.tar.gz	:	Apache2 web server source code
mysql.tar.gz	:	MySQL installation archive
index.html	:	Sample web page

Step 5: Ping the ubuntu server to check it is available and up.

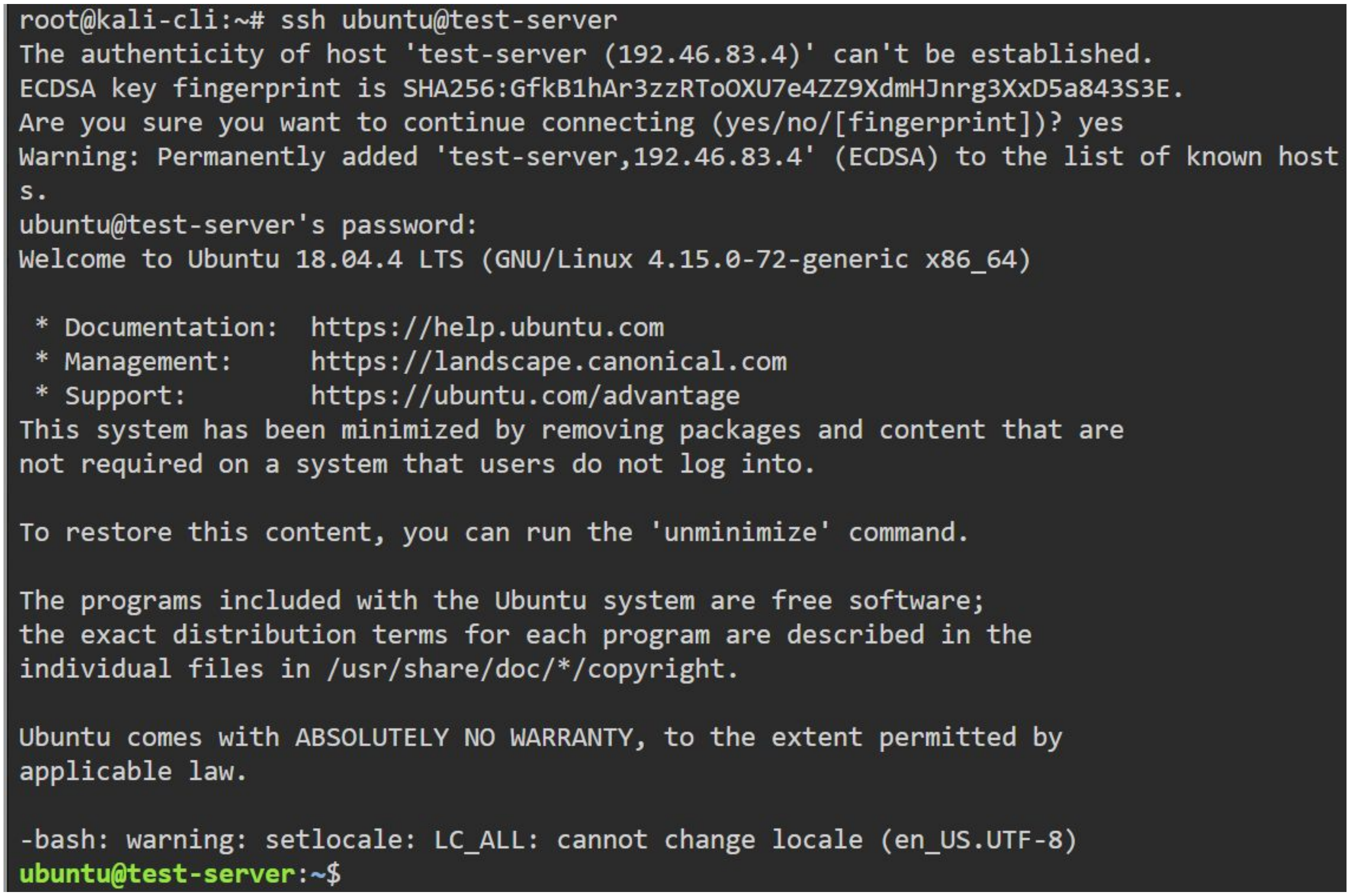
Command: ping test-server



```
root@kali-cli:~# ping test-server
PING test-server (192.46.83.4) 56(84) bytes of data.
64 bytes from test-server (192.46.83.4): icmp_seq=1 ttl=64 time=0.123 ms
64 bytes from test-server (192.46.83.4): icmp_seq=2 ttl=64 time=0.072 ms
64 bytes from test-server (192.46.83.4): icmp_seq=3 ttl=64 time=0.070 ms
64 bytes from test-server (192.46.83.4): icmp_seq=4 ttl=64 time=0.064 ms
64 bytes from test-server (192.46.83.4): icmp_seq=5 ttl=64 time=0.071 ms
```

Step 6: SSH into the ubuntu server as user ‘ubuntu’ and password ‘password1’

Command: ssh ubuntu@test-server



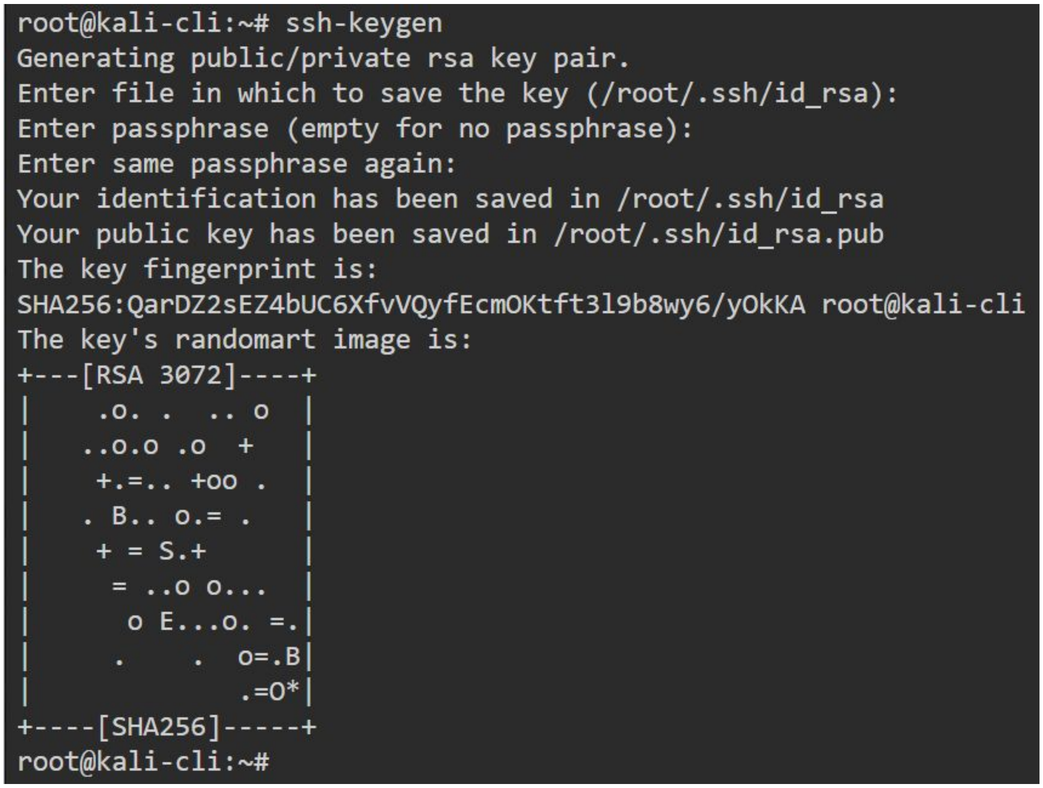
```
root@kali-cli:~# ssh ubuntu@test-server
The authenticity of host 'test-server (192.46.83.4)' can't be established.
ECDSA key fingerprint is SHA256:GfkB1hAr3zzRToOXU7e4ZZ9XdmHJnrg3XxD5a843S3E.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'test-server,192.46.83.4' (ECDSA) to the list of known host
s.
ubuntu@test-server's password:
Welcome to Ubuntu 18.04.4 LTS (GNU/Linux 4.15.0-72-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage
This system has been minimized by removing packages and content that are
not required on a system that users do not log into.

To restore this content, you can run the 'unminimize' command.

The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

-bash: warning: setlocale: LC_ALL: cannot change locale (en_US.UTF-8)
ubuntu@test-server:~$
```

Step 7: Ansible uses SSH connection to remote machine to do installation and configuration tasks. To make sure that Ansible can SSH into the Ubuntu server without prompting for password and showing warnings, one needs to create a SSH public-private key pair for the local machine and add it to authorized keys on the target remote machine.

Generate the SSH key pair.

Command: ssh-keygen

Leave the file path/name and key protection passphrase empty by pressing enter key.

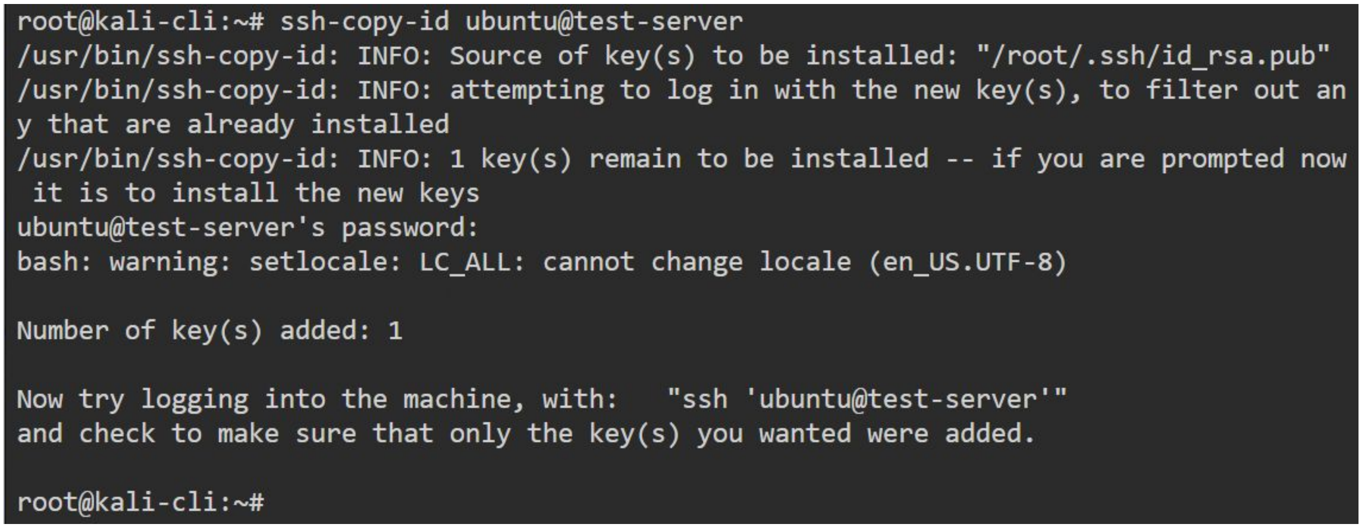


```
root@kali-cli:~# ssh-keygen
Generating public/private rsa key pair.
Enter file in which to save the key (/root/.ssh/id_rsa):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /root/.ssh/id_rsa
Your public key has been saved in /root/.ssh/id_rsa.pub
The key fingerprint is:
SHA256:QarDZ2sEZ4bUC6XfvVQyfEcmOKtft3l9b8wy6/yOkKA root@kali-cli
The key's randomart image is:
+---[RSA 3072]----+
|    .o. .  .. o  |
|   ..o.o .o  +   |
|    +.=.. +oo .  |
|   . B.. o.= .   |
|    + = S.+      |
|     = ..o o...  |
|      o E...o. =.|
|     .    .  o=.B|
|             .=O*|
+----[SHA256]-----+
root@kali-cli:~#
```

Step 8: Add the public key of the local machine to authorized keys of the remote machine. The SSH will prompt for a password. Use the same credentials as used before.

- **Username:** ubuntu
- **Password:** password1

Command: ssh-copy-id ubuntu@test-server



```
root@kali-cli:~# ssh-copy-id ubuntu@test-server
/usr/bin/ssh-copy-id: INFO: Source of key(s) to be installed: "/root/.ssh/id_rsa.pub"
/usr/bin/ssh-copy-id: INFO: attempting to log in with the new key(s), to filter out an
y that are already installed
/usr/bin/ssh-copy-id: INFO: 1 key(s) remain to be installed -- if you are prompted now
 it is to install the new keys
ubuntu@test-server's password:
bash: warning: setlocale: LC_ALL: cannot change locale (en_US.UTF-8)

Number of key(s) added: 1

Now try logging into the machine, with:   "ssh 'ubuntu@test-server'"
and check to make sure that only the key(s) you wanted were added.

root@kali-cli:~#
```

Now, the test-server is ready to be used with Ansible.

Step 9: Check the YAML file provided in the home directory of the root user.

Command: cat apache-mysql.yml

apache-mysql.yml

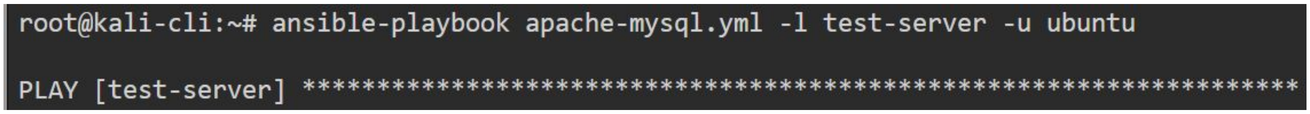
```
---
 - hosts: test-server
   become: yes
   tasks:
    - name: Extracting Apache2 web server source code archive on remote target machine
      unarchive:
       src: /root/setup-files/apache2_2.4.41.orig.tar.gz
       dest: /

    - name: Run configure on Apache2  (Build from source code step)
      command: ./configure -prefix=/usr/local/apache2
      args:
       chdir: "/httpd-2.4.41"

    - name: Building and Installing Apache2 (Build from source code step)
      command: "{{ item }} chdir=/httpd-2.4.41"
      with_items:
      - make
      - make install
    - name: Removing Apache2 Build Directory
      file:
       path: "/httpd-2.4.41"
       state: absent

    - name: Extracting MySQL server source code archive on remote target machine
      unarchive:
       src: /root/setup-files/mysql.tar.gz
       dest: /usr/local/

    - name: Making configuraton changes for MySQL server
      command: "{{ item }} chdir=/usr/local/mysql"
      args:
       warn: no
      with_items:
      - groupadd mysql
      - useradd -d /home/mysql -g mysql mysql
      - chown -R mysql:mysql /usr/local/mysql
      - scripts/mysql_install_db --user=mysql
      - chown -R root /usr/local/mysql
      - chown -R mysql /usr/local/mysql/data
      - cp /usr/local/mysql/support-files/my-medium.cnf /etc/my.cnf
      - ln -s /usr/local/mysql/bin/mysqld_safe /usr/local/bin/mysqld_safe
      - ln -s /usr/local/mysql/bin/mysql /usr/local/bin/mysql
      - ln -s /usr/local/mysql/bin/mysqladmin /usr/local/bin/mysqladmin
      - sed -i 's/\[mysqld\]/\[mysqld\]\nbind-address = 127.0.0.1/g' /etc/my.cnf

    - name: Starting Apache2 web server
      command: /usr/local/apache2/bin/apachectl start
      args:
       warn: no

    - name: Copying a sample webpage to Apache2 web server's web directory
      copy:
       src: /root/setup-files/index.html
       dest: /usr/local/apache2/htdocs/index.html

    - name: Starting MySQL server
      command: /usr/local/mysql/support-files/mysql.server start
      args:
       warn: no
```

This file will install and start MySQL/Apache2 combo on the Ubuntu machine.

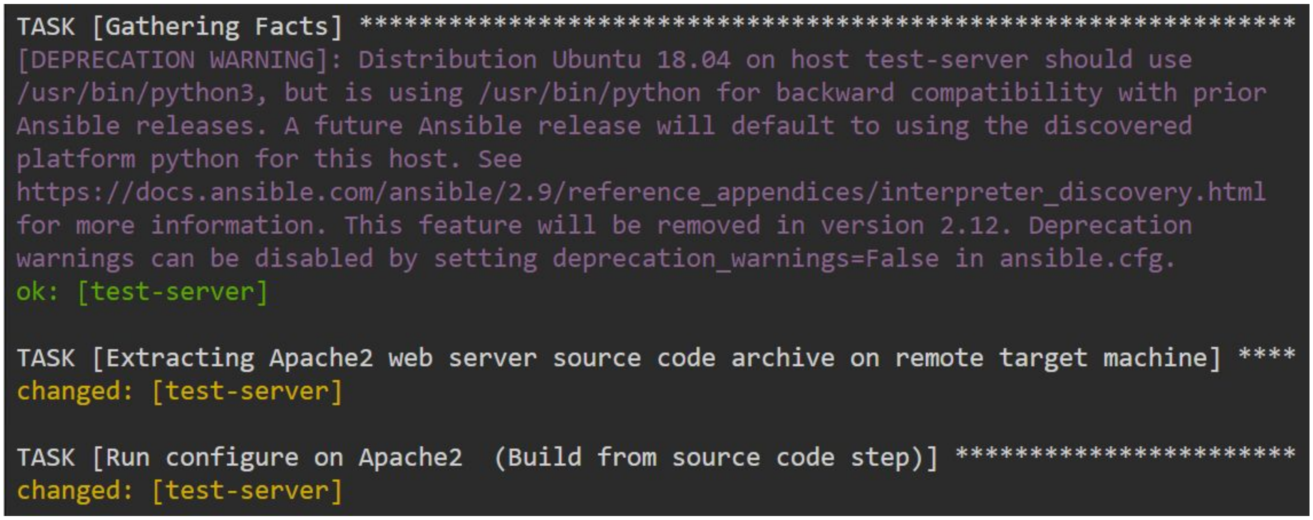
Step 10: Run the ansible-playbook with the following arguments:

YAML file name	:	apache-mysql.yml
Target machine name	:	test-server
User of target machine	:	ubuntu

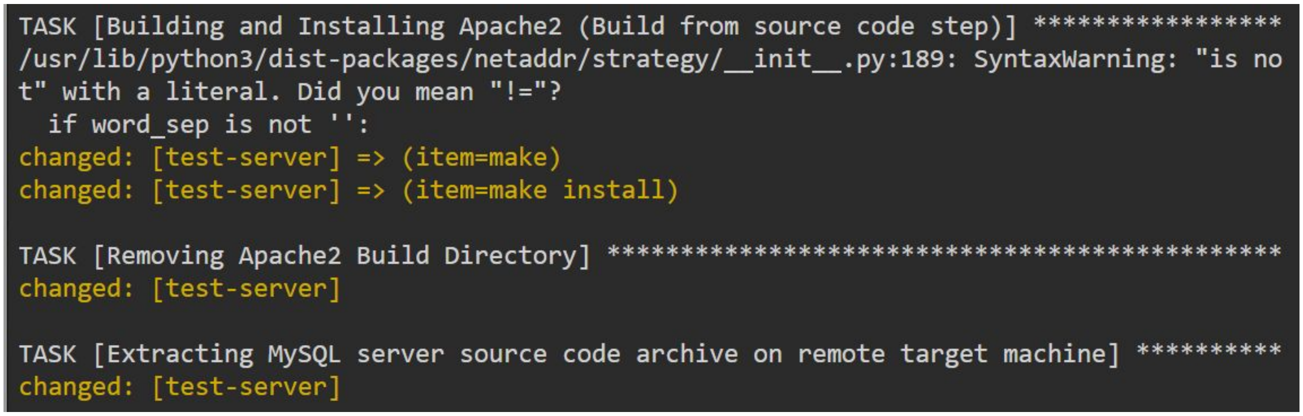
Command: ansible-playbook apache-mysql.yml -l test-server -u ubuntu



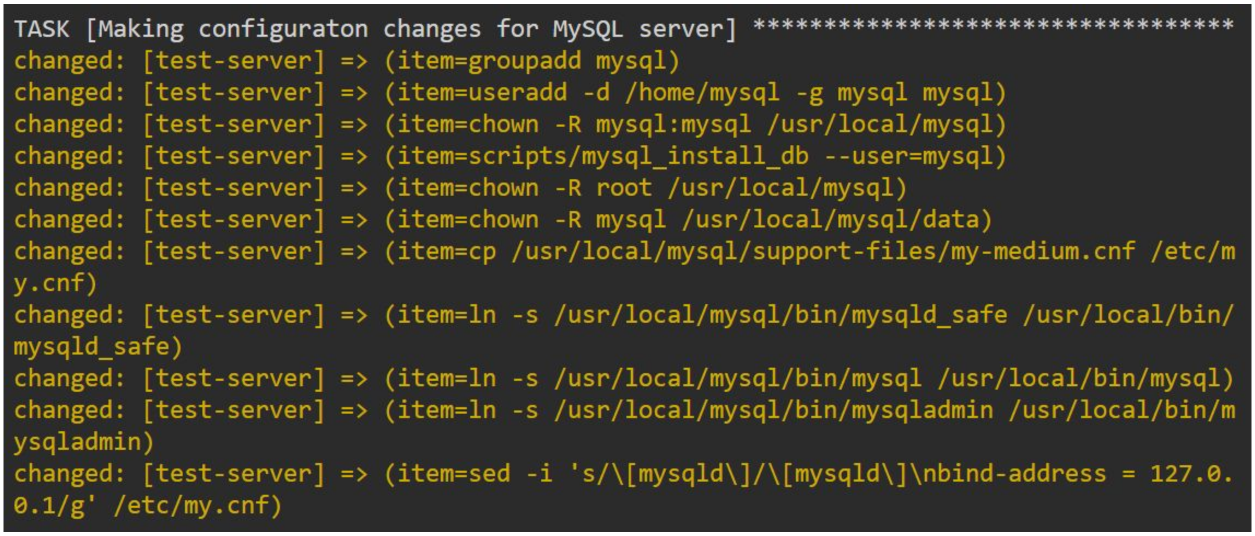
```
root@kali-cli:~# ansible-playbook apache-mysql.yml -l test-server -u ubuntu

PLAY [test-server] *******************************************************************
```



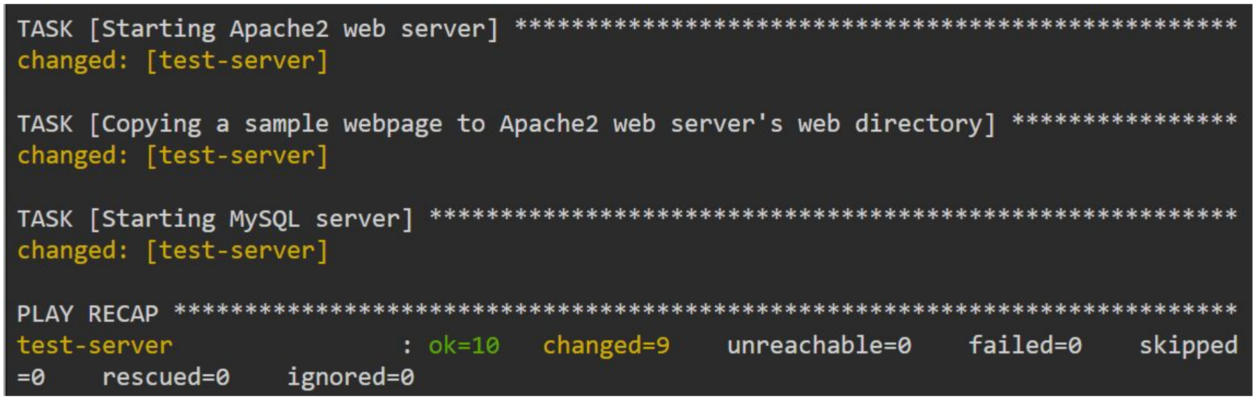
```
TASK [Gathering Facts] ****************************************************************
[DEPRECATION WARNING]: Distribution Ubuntu 18.04 on host test-server should use
/usr/bin/python3, but is using /usr/bin/python for backward compatibility with prior
Ansible releases. A future Ansible release will default to using the discovered
platform python for this host. See
https://docs.ansible.com/ansible/2.9/reference_appendices/interpreter_discovery.html
for more information. This feature will be removed in version 2.12. Deprecation
warnings can be disabled by setting deprecation_warnings=False in ansible.cfg.
ok: [test-server]

TASK [Extracting Apache2 web server source code archive on remote target machine] ****
changed: [test-server]

TASK [Run configure on Apache2  (Build from source code step)] ***********************
changed: [test-server]
```



```
TASK [Building and Installing Apache2 (Build from source code step)] *****************
/usr/lib/python3/dist-packages/netaddr/strategy/__init__.py:189: SyntaxWarning: "is no
t" with a literal. Did you mean "!="?
  if word_sep is not '':
changed: [test-server] => (item=make)
changed: [test-server] => (item=make install)

TASK [Removing Apache2 Build Directory] ***********************************************
changed: [test-server]

TASK [Extracting MySQL server source code archive on remote target machine] **********
changed: [test-server]
```



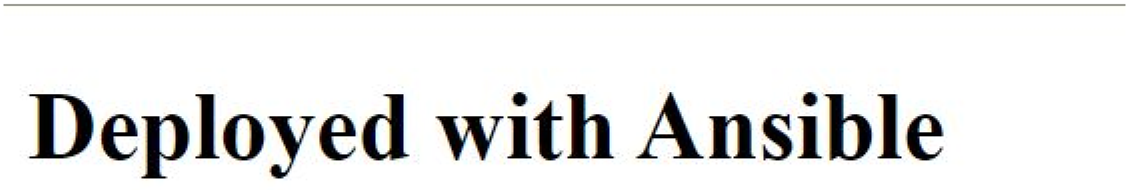
```
TASK [Making configuraton changes for MySQL server] **********************************
changed: [test-server] => (item=groupadd mysql)
changed: [test-server] => (item=useradd -d /home/mysql -g mysql mysql)
changed: [test-server] => (item=chown -R mysql:mysql /usr/local/mysql)
changed: [test-server] => (item=scripts/mysql_install_db --user=mysql)
changed: [test-server] => (item=chown -R root /usr/local/mysql)
changed: [test-server] => (item=chown -R mysql /usr/local/mysql/data)
changed: [test-server] => (item=cp /usr/local/mysql/support-files/my-medium.cnf /etc/m
y.cnf)
changed: [test-server] => (item=ln -s /usr/local/mysql/bin/mysqld_safe /usr/local/bin/
mysqld_safe)
changed: [test-server] => (item=ln -s /usr/local/mysql/bin/mysql /usr/local/bin/mysql)
changed: [test-server] => (item=ln -s /usr/local/mysql/bin/mysqladmin /usr/local/bin/m
ysqladmin)
changed: [test-server] => (item=sed -i 's/\[mysqld\]/\[mysqld\]\nbind-address = 127.0.
0.1/g' /etc/my.cnf)
```



```
TASK [Starting Apache2 web server] ***************************************************
changed: [test-server]

TASK [Copying a sample webpage to Apache2 web server's web directory] ****************
changed: [test-server]

TASK [Starting MySQL server] *********************************************************
changed: [test-server]

PLAY RECAP ***************************************************************************
test-server                : ok=10   changed=9    unreachable=0    failed=0    skipped
=0    rescued=0    ignored=0
```

The operation was executed successfully.

Step 11: Check or reload the interface showing port 80 of Ubuntu server.



Deployed with Ansible

The sample web page provided with the setup files is hosted on it.

Learnings

Install and configure components on remote machines using Ansible.