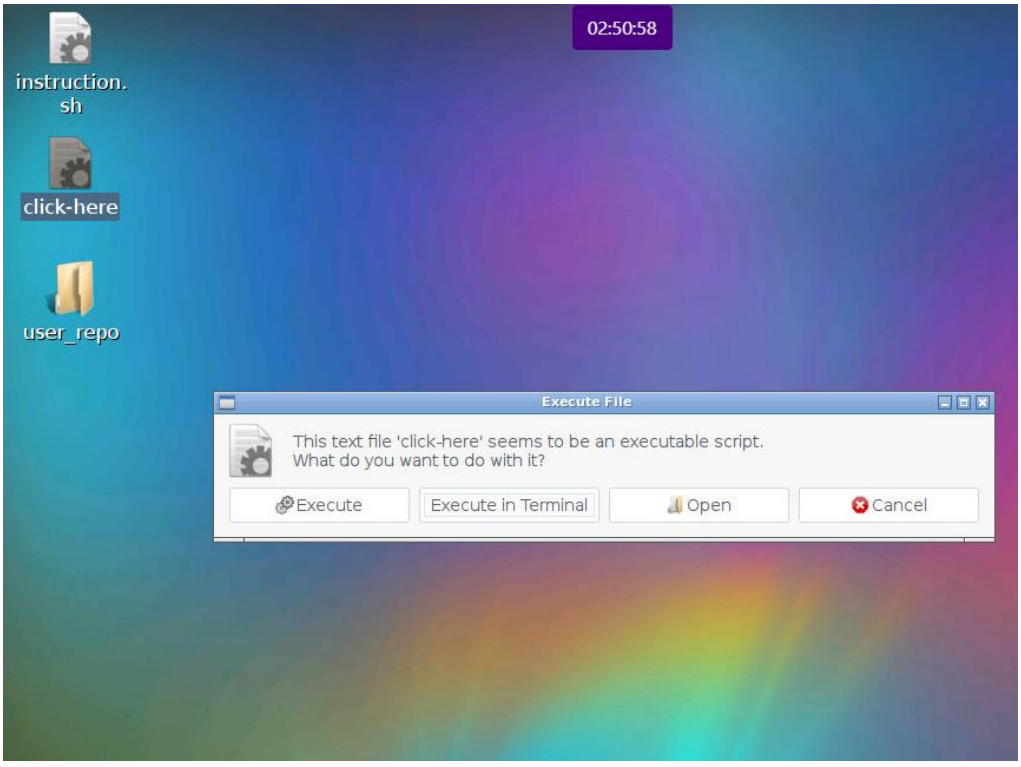
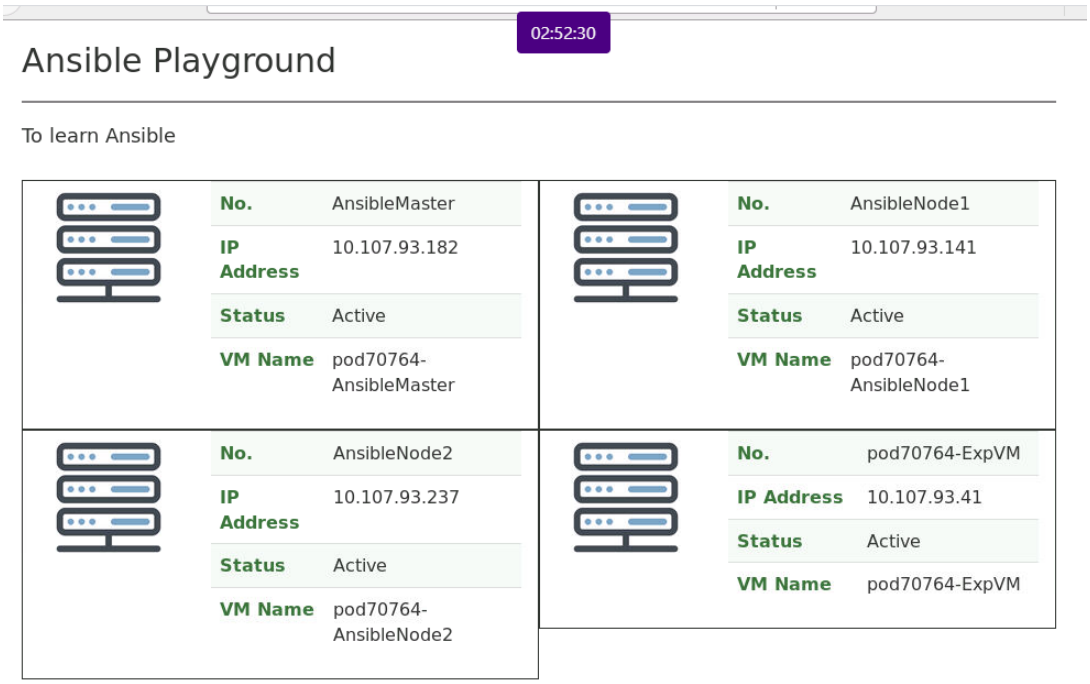
Ansible Hands on Guide

Contents

Setup Ansible Master and Nodes

After landing on the server in Ansible playground, double click on click-here file and select execute. Ansible server and node details are displayed in Mozilla Firefox.





Ssh to the Ansible master node using the below command

ssh esadmin@<vmname>

Enter the password when prompted.

Navigate to the folder where hands-on code is present.

Lab 01 – Verify Ansible Installation

Validate the version with the command.

ansible ---version

```
esadmin@pod70764-AnsibleMaster:~/lab/ansible-tutorial$ ansible --version
ansible 2.8.3
  config file = /etc/ansible/ansible.cfg
  configured module search path = [u'/home/esadmin/.ansible/plugins/modules', u'
/usr/share/ansible/plugins/modules']
  ansible python module location = /usr/lib/python2.7/dist-packages/ansible
  executable location = /usr/bin/ansible
  python version = 2.7.12 (default, Apr 15 2020, 17:07:12) [GCC 5.4.0 20160609]
esadmin@pod70764-AnsibleMaster:~/lab/ansible-tutorial$
```

Also you may test the simple Ansible command to ping the local host by issuing below command

ansible –m ping localhost

Navigate to lab-01 directory

cd lab-01

Both the commands are included in verify-version.sh script.

verify-version.sh script can be executed as shown below.

```
esadmin@pod70764-AnsibleMaster:~/lab/ansible-tutorial/lab-01$ ./verify-version.s
h
ansible 2.8.3
  config file = /etc/ansible/ansible.cfg
  configured module search path = [u'/home/esadmin/.ansible/plugins/modules', u'
/usr/share/ansible/plugins/modules']
  ansible python module location = /usr/lib/python2.7/dist-packages/ansible
  executable location = /usr/bin/ansible
  python version = 2.7.12 (default, Apr 15 2020, 17:07:12) [GCC 5.4.0 20160609]
localhost | SUCCESS => {
    "changed": false,
    "ping": "pong"
}
esadmin@pod70764-AnsibleMaster:~/lab/ansible-tutorial/lab-01$
```

Configure SSH key for remote access

1. Create a SSH key using ***ssh-keygen*** on Ansible master

```
esadmin@pod70764-AnsibleMaster:~/lab/ansible-tutorial/lab-01$ ssh-keygen
Generating public/private rsa key pair.
Enter file in which to save the key (/home/esadmin/.ssh/id_rsa):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/esadmin/.ssh/id_rsa.
Your public key has been saved in /home/esadmin/.ssh/id_rsa.pub.
The key fingerprint is:
SHA256:td4rgDs7fZ7VM2VS30IVR9f0jm01HOwiO2Vq1RTzAnM esadmin@pod70764-AnsibleMaste
r
The key's randomart image is:
+---[RSA 2048]----+
|            o EBX|
|             +.+O|
|          .   *o=|
|         . o * O=|
|       .S . B = O|
|      . .. * . * |
|       o .o + +  |
|      + . oo . o |
|      .+ oo..    |
+----[SHA256]-----+
esadmin@pod70764-AnsibleMaster:~/lab/ansible-tutorial/lab-01$
```

2. Copy the generated key into all other hosts for easier access by running the below command.

- ssh-copy-id -i /home/esadmin/.ssh/id_rsa esadmin@<node1 name>
- ssh-copy-id -i /home/esadmin/.ssh/id_rsa esadmin@<node2 name>

```
esadmin@pod70764-AnsibleMaster:~/lab/ansible-tutorial/lab-01$ ssh-copy-id -i /ho
me/esadmin/.ssh/id_rsa esadmin@pod70764-AnsibleNode1
/usr/bin/ssh-copy-id: INFO: Source of key(s) to be installed: "/home/esadmin/.ss
h/id_rsa.pub"
The authenticity of host 'pod70764-ansiblenode1 (fd42:af61:ff2:b2e7:216:3eff:fef
0:c4b9)' can't be established.
ECDSA key fingerprint is SHA256:vK56RBViBlQgsF3s31k7w5soeYKqxcjzo67Bg5es0WM.
Are you sure you want to continue connecting (yes/no)? yes
/usr/bin/ssh-copy-id: INFO: attempting to log in with the new key(s), to filter
out any that are already installed
/usr/bin/ssh-copy-id: INFO: 1 key(s) remain to be installed -- if you are prompt
ed now it is to install the new keys
esadmin@pod70764-ansiblenode1's password:

Number of key(s) added: 1

Now try logging into the machine, with:   "ssh 'esadmin@pod70764-AnsibleNode1'"
and check to make sure that only the key(s) you wanted were added.
```

```
esadmin@pod70764-AnsibleMaster:~/lab/ansible-tutorial/lab-01$ ssh-copy-id -i /ho
me/esadmin/.ssh/id_rsa esadmin@pod70764-AnsibleNode2
/usr/bin/ssh-copy-id: INFO: Source of key(s) to be installed: "/home/esadmin/.ss
h/id_rsa.pub"
The authenticity of host 'pod70764-ansiblenode2 (fd42:af61:ff2:b2e7:216:3eff:fec
e:27ab)' can't be established.
ECDSA key fingerprint is SHA256:E5wkqeWHjZYY3nTE2Zx7SMSowhXNt6m8mXmYf3xWElk.
Are you sure you want to continue connecting (yes/no)? yes
/usr/bin/ssh-copy-id: INFO: attempting to log in with the new key(s), to filter
out any that are already installed
/usr/bin/ssh-copy-id: INFO: 1 key(s) remain to be installed -- if you are prompt
ed now it is to install the new keys
esadmin@pod70764-ansiblenode2's password:

Number of key(s) added: 1

Now try logging into the machine, with:   "ssh 'esadmin@pod70764-AnsibleNode2'"
and check to make sure that only the key(s) you wanted were added.

esadmin@pod70764-AnsibleMaster:~/lab/ansible-tutorial/lab-01$
```

```
esadmin@pod70764-AnsibleMaster:~/lab/ansible-tutorial/lab-01$ ssh esadmin@pod707
64-AnsibleNode1
Welcome to Ubuntu 16.04.6 LTS (GNU/Linux 4.15.0-1063-aws x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage

  Get cloud support with Ubuntu Advantage Cloud Guest:
    http://www.ubuntu.com/business/services/cloud

59 packages can be updated.
0 updates are security updates.

New release '18.04.4 LTS' available.
Run 'do-release-upgrade' to upgrade to it.


Last login: Thu Aug  8 08:25:46 2019 from fd42:d267:f50:5fc1:216:3eff:fe63:eb85
esadmin@pod70764-AnsibleNode1:~$
```

Note: In production, we will be using Ansible tower where encrypted password of user ID of the remote machine account is stored for remote. However, for our convenience we are using ssh key for remote access.

```
esadmin@pod70764-AnsibleMaster:~/lab/ansible-tutorial/lab-01$ ssh esadmin@pod707
64-AnsibleNode2
Welcome to Ubuntu 16.04.6 LTS (GNU/Linux 4.15.0-1063-aws x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage

  Get cloud support with Ubuntu Advantage Cloud Guest:
    http://www.ubuntu.com/business/services/cloud

59 packages can be updated.
0 updates are security updates.

New release '18.04.4 LTS' available.
Run 'do-release-upgrade' to upgrade to it.


Last login: Thu Aug  8 08:25:46 2019 from fd42:d267:f50:5fc1:216:3eff:fe63:eb85
esadmin@pod70764-AnsibleNode2:~$
```

Validate the SSH access to all machines by running below command

- ssh esadmin@<node1 name> hostname
- ssh esadmin@<node2 name> hostname

```
esadmin@pod70764-AnsibleMaster:~/lab/ansible-tutorial/lab-01$ ssh esadmin@pod70764-AnsibleNode1 hostname
pod70764-AnsibleNode1
esadmin@pod70764-AnsibleMaster:~/lab/ansible-tutorial/lab-01$ ssh esadmin@pod70764-AnsibleNode2 hostname
pod70764-AnsibleNode2
esadmin@pod70764-AnsibleMaster:~/lab/ansible-tutorial/lab-01$
```

For running ad-hoc commands as well as to run playbooks, we will be creating an inventory file and update/replace the file /etc/Ansible/host in control node.

Lab 02 - Creating Inventory

1. On control node, change directory to lab-02 directory running below command

cd lab-02

```
esadmin@pod70764-AnsibleMaster:~/lab/ansible-tutorial/lab-02$ ls -all
total 16
drwxrwxr-x 2 esadmin esadmin 4096 May 19 09:43 .
drwxrwxr-x 9 esadmin esadmin 4096 May 19 09:27 ..
-rw-rw-r-- 1 esadmin esadmin  532 May 19 09:40 ad-hoc-commands-on-inventory.sh
-rw-rw-r-- 1 esadmin esadmin  139 May 19 09:43 hosts
esadmin@pod70764-AnsibleMaster:~/lab/ansible-tutorial/lab-02$
```

```
web1 ansible_host=pod70764-AnsibleNode1
app1 ansible_host=pod70764-AnsibleNode2

[webservers]
web1

[appservers]
app1

[dc:children]
webservers
appservers
~
```

2. Replace the Ansible hosts file with ab-02 by running below command

sudo cp -v hosts /etc/ansible/hosts

```
esadmin@pod70764-AnsibleMaster:~/lab/ansible-tutorial/lab-02$ sudo cp -v hosts /
etc/ansible/hosts
'hosts' -> '/etc/ansible/hosts'
esadmin@pod70764-AnsibleMaster:~/lab/ansible-tutorial/lab-02$
```

3. Run below ad-hoc commands and validate the result.

- ansible -u esadmin -m ping web1
- ansible -u esadmin -m ping app1
- ansible -u esadmin -m ping webservers
- ansible -u esadmin -m ping dc
- ansible -u esadmin -m shell -a 'ls -al' web1
- ansible -u esadmin -m shell -a 'whoami' app1
- ansible -u esadmin -m shell -a 'ifconfig' webservers
- ansible -u esadmin -m shell -a 'hostname' dc
- ansible -u esadmin -m ping all
- ansible -u esadmin -m ping web*
- ansible -u esadmin -m ping 'appservers:dbservers'
- ansible -u esadmin -m ping 'dc:!webservers'
- ansible -u esadmin -m ping 'dc:&webservers'

```
esadmin@pod70764-AnsibleMaster:~/lab/ansible-tutorial/lab-02$ ansible -u esadmin -m ping web1
[DEPRECATION WARNING]: Distribution Ubuntu 16.04 on host web1 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/2.8/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.
web1 | SUCCESS => {
    "ansible_facts": {
        "discovered_interpreter_python": "/usr/bin/python"
    },
    "changed": false,
    "ping": "pong"
}
esadmin@pod70764-AnsibleMaster:~/lab/ansible-tutorial/lab-02$ ansible -u esadmin -m ping app1
[DEPRECATION WARNING]: Distribution Ubuntu 16.04 on host app1 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/2.8/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.
app1 | SUCCESS => {
    "ansible_facts": {
        "discovered_interpreter_python": "/usr/bin/python"
    },
    "changed": false,
    "ping": "pong"
}
esadmin@pod70764-AnsibleMaster:~/lab/ansible-tutorial/lab-02$
```

```
#!/bin/sh

ansible -u esadmin -m ping web1
ansible -u esadmin -m ping app1
ansible -u esadmin -m ping webservers
ansible -u esadmin -m ping dc
ansible -u esadmin -m shell -a 'ls -al' web1
ansible -u esadmin -m shell -a 'whoami' app1
ansible -u esadmin -m shell -a 'ifconfig' webservers
ansible -u esadmin -m shell -a 'hostname' dc
ansible -u esadmin -m ping all
ansible -u esadmin -m ping web*
ansible -u esadmin -m ping 'appservers:dbservers'
ansible -u esadmin -m ping 'dc:!webservers'
ansible -u esadmin -m ping 'dc:&webservers'
~
```

ad-hoc-commands-on-inventory.sh has the above commands. This file can be executed to run ad-hoc commands.

```
esadmin@pod70764-AnsibleMaster:~/lab/ansible-tutorial/lab-02$ ./ad-hoc-commands-on-inventory.sh
[DEPRECATION WARNING]: Distribution Ubuntu 16.04 on host web1 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/
2.8/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.
web1 | SUCCESS => {
    "ansible_facts": {
        "discovered_interpreter_python": "/usr/bin/python"
    },
    "changed": false,
    "ping": "pong"
}
[DEPRECATION WARNING]: Distribution Ubuntu 16.04 on host app1 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/
2.8/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.
app1 | SUCCESS => {
    "ansible_facts": {
        "discovered_interpreter_python": "/usr/bin/python"
    },
    "changed": false,
```

Lab-03 – Using common modules

1. To install the nginx on webserver we can use the –m apt

ansible -m apt -a "name=nginx state=present update_cache=yes" web1

as you may observe in below screenshot, nginx installation failed due to access privileges. Hence by using the parameter --***become*** helps to run the ansible module on remote server with sudo privileges.

```
esadmin@pod70764-AnsibleMaster:~/lab/ansible-tutorial/[redacted]nsible -m apt -a "name=nginx state=present update
_cache=yes" web1
 [WARNING]: Updating cache and auto-installing missing dependency: python-apt

[DEPRECATION WARNING]: Distribution Ubuntu 16.04 on host web1 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/2.8/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.
web1 | FAILED! => {
    "ansible_facts": {
        "discovered_interpreter_python": "/usr/bin/python"
    },
    "changed": false,
    "cmd": "apt-get update",
    "msg": "W: chmod 0700 of directory /var/lib/apt/lists/partial failed - SetupAPTPartialDirectory (1: Operatio
n not permitted)\nE: Could not open lock file /var/lib/apt/lists/lock - open (13: Permission denied)\nE: Unable
to lock directory /var/lib/apt/lists/\nW: Problem unlinking the file /var/cache/apt/pkgcache.bin - RemoveCaches
(13: Permission denied)\nW: Problem unlinking the file /var/cache/apt/srcpkgcache.bin - RemoveCaches (13: Permis
sion denied)",
    "rc": 100,
    "stderr": "W: chmod 0700 of directory /var/lib/apt/lists/partial failed - SetupAPTPartialDirectory (1: Opera
tion not permitted)\nE: Could not open lock file /var/lib/apt/lists/lock - open (13: Permission denied)\nE: Unab
le to lock directory /var/lib/apt/lists/\nW: Problem unlinking the file /var/cache/apt/pkgcache.bin - RemoveCach
es (13: Permission denied)\nW: Problem unlinking the file /var/cache/apt/srcpkgcache.bin - RemoveCaches (13: Per
mission denied)\n",
    "stderr_lines": [
        "W: chmod 0700 of directory /var/lib/apt/lists/partial failed - SetupAPTPartialDirectory (1: Operation n
ot permitted)",
        "E: Could not open lock file /var/lib/apt/lists/lock - open (13: Permission denied)",
        "E: Unable to lock directory /var/lib/apt/lists/",
        "W: Problem unlinking the file /var/cache/apt/pkgcache.bin - RemoveCaches (13: Permission denied)",

esadmin@pod70764-AnsibleMaster:~/lab/ansible-tutorial/lab-02$ ansible -m apt -a "name=nginx state=present update
_cache=yes" web1 --become
 [WARNING]: Updating cache and auto-installing missing dependency: python-apt

 [WARNING]: Could not find aptitude. Using apt-get instead

[DEPRECATION WARNING]: Distribution Ubuntu 16.04 on host web1 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/2.8/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.
web1 | CHANGED => {
    "ansible_facts": {
        "discovered_interpreter_python": "/usr/bin/python"
    },
    "cache_update_time": 1589882486,
    "cache_updated": true,
    "changed": true,
    "stderr": "",
    "stderr_lines": [],
    "stdout": "Reading package lists...\nBuilding dependency tree...\nReading state information...\nThe followin
g additional packages will be installed:\n  fontconfig-config fonts-dejavu-core libfontconfig1 libgd3 libjbig0\n
  libjpeg-turbo8 libjpeg8 libtiff5 libvpx3 libxpm4 nginx-common nginx-core\nSuggested packages:\n  libgd-tools f
cgiwrap nginx-doc ssl-cert\nThe following NEW packages will be installed:\n  fontconfig-config fonts-dejavu-core
 libfontconfig1 libgd3 libjbig0\n  libjpeg-turbo8 libjpeg8 libtiff5 libvpx3 libxpm4 nginx nginx-common\n  nginx-
core\n0 upgraded, 13 newly installed, 0 to remove and 61 not upgraded.\nNeed to get 2860 kB of archives.\nAfter
this operation, 9315 kB of additional disk space will be used.\nGet:1 http://archive.ubuntu.com/ubuntu xenial-up
dates/main amd64 libjpeg-turbo8 amd64 1.4.2-0ubuntu3.3 [111 kB]\nGet:2 http://archive.ubuntu.com/ubuntu xenial/m
```

2. To un-install the application, we need to mark the application state as absent.

ansible -m apt -a "name=nginx state=absent" web1 --become

```
esadmin@pod70764-AnsibleMaster:~/lab/ansible-tutorial/lab-02$ ansible -m apt -a "name=nginx state=absent" web1 -
-become
 [WARNING]: Could not find aptitude. Using apt-get instead

[DEPRECATION WARNING]: Distribution Ubuntu 16.04 on host web1 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/2.8/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.
web1 | CHANGED => {
    "ansible_facts": {
        "discovered_interpreter_python": "/usr/bin/python"
    },
    "changed": true,
    "stderr": "",
    "stderr_lines": [],
    "stdout": "Reading package lists...\nBuilding dependency tree...\nReading state information...\nThe followin
g packages were automatically installed and are no longer required:\n  fontconfig-config fonts-dejavu-core libfo
ntconfig1 libfreetype6 libgd3\n  libjbig0 libjpeg-turbo8 libjpeg8 libtiff5 libvpx3 libxpm4 nginx-common\n  nginx
-core\nUse 'sudo apt autoremove' to remove them.\nThe following packages will be REMOVED:\n  nginx\n0 upgraded,
0 newly installed, 1 to remove and 61 not upgraded.\nAfter this operation, 38.9 kB disk space will be freed.\n(R
eading database ... \r(Reading database ... 5%\r(Reading database ... 10%\r(Reading database ... 15%\r(Reading d
atabase ... 20%\r(Reading database ... 25%\r(Reading database ... 30%\r(Reading database ... 35%\r(Reading datab
ase ... 40%\r(Reading database ... 45%\r(Reading database ... 50%\r(Reading database ... 55%\r(Reading database
... 60%\r(Reading database ... 65%\r(Reading database ... 70%\r(Reading database ... 75%\r(Reading database ...
80%\r(Reading database ... 85%\r(Reading database ... 90%\r(Reading database ... 95%\r(Reading database ... 100%
\r(Reading database ... 26925 files and directories currently installed.)\r\nRemoving nginx (1.10.3-0ubuntu0.16.
04.5) ...\r\n",
    "stdout_lines": [
        "Reading package lists...",
```

3. Issue the below command to execute Ansible module to start the service on remote node.

ansible -m service -a "name=nginx state=started enabled=yes" webservers --become

```
esadmin@pod70764-AnsibleMaster:~/lab/ansible-tutorial/lab-03$ ansible -m service -a "name=nginx state=started" w
eb1 --become
[DEPRECATION WARNING]: Distribution Ubuntu 16.04 on host web1 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/2.8/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.
web1 | SUCCESS => {
    "ansible_facts": {
        "discovered_interpreter_python": "/usr/bin/python"
    },
    "changed": false,
    "name": "nginx",
    "state": "started",
    "status": {
        "ActiveEnterTimestamp": "Tue 2020-05-19 10:01:30 UTC",
        "ActiveEnterTimestampMonotonic": "1317135955836",
        "ActiveExitTimestampMonotonic": "0",
        "ActiveState": "active",
        "After": "system.slice sysinit.target basic.target network.target systemd-journald.socket",
        "AllowIsolate": "no"
```

4. We can run the ansible module on groups as well. Here is the example of how ansible module be run to install nginx on webserver

```
esadmin@pod70764-AnsibleMaster:~/lab/ansible-tutorial/lab-03$ ansible -m apt -a "name=nginx state=present update
_cache=yes" webservers --become
 [WARNING]: Could not find aptitude. Using apt-get instead

[DEPRECATION WARNING]: Distribution Ubuntu 16.04 on host web1 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/2.8/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.
web1 | CHANGED => {
    "ansible_facts": {
        "discovered_interpreter_python": "/usr/bin/python"
    },
    "cache_update_time": 1589882974,
    "cache_updated": true,
    "changed": true,
    "stderr": "",
    "stderr_lines": [],
```

5. Ansible provides various modules to perform configuration activity on remote server.

Here is the example where index.html file from control node is been copied to /usr/share/nginx/html directory.

```
esadmin@pod70764-AnsibleMaster:~/lab/ansible-tutorial/lab-03$ ansible -m file -a "path=/usr/share/nginx/html sta
te=directory" webservers --become
[DEPRECATION WARNING]: Distribution Ubuntu 16.04 on host web1 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/2.8/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.
web1 | SUCCESS => {
    "ansible_facts": {
        "discovered_interpreter_python": "/usr/bin/python"
    },
    "changed": false,
    "gid": 0,
    "group": "root",
    "mode": "0755",
    "owner": "root",
    "path": "/usr/share/nginx/html",
    "size": 4096,
    "state": "directory",
    "uid": 0
}

esadmin@pod70764-AnsibleMaster:~/lab/ansible-tutorial/lab-03$ ansible -m copy -a "src=index.html dest=/usr/share
/nginx/html/index.html" webservers --become
[DEPRECATION WARNING]: Distribution Ubuntu 16.04 on host web1 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/2.8/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.
web1 | CHANGED => {
    "ansible_facts": {
        "discovered_interpreter_python": "/usr/bin/python"
    },
    "changed": true,
    "checksum": "6bf18ee785817e9ba60d3a953f13080b5e5ae303",
    "dest": "/usr/share/nginx/html/index.html",
    "gid": 0,
    "group": "root",
    "md5sum": "c583e6297a082341b3001958dd268da6",
    "mode": "0644",
    "owner": "root",
    "size": 22,
    "src": "/home/esadmin/.ansible/tmp/ansible-tmp-1589883196.94-89306763078986/source",
    "state": "file",
    "uid": 0
}
esadmin@pod70764-AnsibleMaster:~/lab/ansible-tutorial/lab-03$
```

```
esadmin@pod70764-AnsibleMaster:~/lab/ansible-tutorial/lab-03$ ls -all
total 16
drwxrwxr-x 2 esadmin esadmin 4096 May 19 10:14 .
drwxrwxr-x 9 esadmin esadmin 4096 May 19 09:27 ..
-rw-rw-r-- 1 esadmin esadmin   22 May 19 09:27 index.html
-rw-rw-r-- 1 esadmin esadmin  351 May 19 10:14 modules-as-ad-hoc-commands.sh
esadmin@pod70764-AnsibleMaster:~/lab/ansible-tutorial/lab-03$
```

Below is the content of modules-as-ad-hoc-commands.sh

```
#!/bin/sh
ansible -m apt -a "name=nginx state=present update_cache=yes" webservers --become
ansible -m service -a "name=nginx state=started enabled=yes" webservers --become
ansible -m file -a "path=/usr/share/nginx/html state=directory" webservers --become
ansible -m copy -a "src=index.html dest=/usr/share/nginx/html/index.html" webservers --become
~
~
~
```

04-Running Playbooks

Ansible playbook is the easier way to run the Ansible modules on remote machines. These are files which can be developed and tested on local environments before running on production environments.

Here is the example of how we can install nginx on webservers.

Change directory to lab-04

Playbook syntax-check

Ansible play throws errors if there any whitespace. In playbook space matters a lot in YAML, as it strictly uses space character as indent. To validate the written playbook, you may use the –syntax-check parameter. If there is no error, then the playbook name will be displayed as output.

```
esadmin@pod70764-AnsibleMaster:~/lab/ansible-tutorial/lab-03$ cd ..
esadmin@pod70764-AnsibleMaster:~/lab/ansible-tutorial$ cd lab-04/
esadmin@pod70764-AnsibleMaster:~/lab/ansible-tutorial/lab-04$ ls -all
total 16
drwxrwxr-x 2 esadmin esadmin 4096 May 19 09:27 .
drwxrwxr-x 9 esadmin esadmin 4096 May 19 09:27 ..
-rw-rw-r-- 1 esadmin esadmin  293 May 19 09:27 install-nginx-and-jdk.yml
-rw-rw-r-- 1 esadmin esadmin  179 May 19 09:27 install-nginx.yml
esadmin@pod70764-AnsibleMaster:~/lab/ansible-tutorial/lab-04$
```

```
esadmin@pod70764-AnsibleMaster:~/lab/ansible-tutorial/lab-04$ cat install-nginx.yml
---
- hosts: webservers
  become: true
  tasks:
  - name: Install nginx
    apt: name=nginx state=present
  - name: Start nginx
    service: name=nginx state=started enabled=yes

esadmin@pod70764-AnsibleMaster:~/lab/ansible-tutorial/lab-04$

esadmin@pod70764-AnsibleMaster:~/lab/ansible-tutorial/lab-04$ ansible-playbook install-nginx.yml --syntax-check

playbook: install-nginx.yml
esadmin@pod70764-AnsibleMaster:~/lab/ansible-tutorial/lab-04$
```

Running the playbook

Before running the playbook, un-install the nginx on webservers using below command

ansible -m apt -a "name=nginx state=absent update_cache=yes" webservers –become

run the playbook using below command to install nginx on webservers

ansible-playbook install-nginx.yml

```
esadmin@pod70764-AnsibleMaster:~/lab/ansible-tutorial/lab-04$ ansible-playbook install-nginx.yml

PLAY [webservers] ***************************************************************************************************

TASK [Gathering Facts] **********************************************************************************************
ok: [web1]

TASK [Install nginx] ************************************************************************************************
 [WARNING]: Could not find aptitude. Using apt-get instead

changed: [web1]

TASK [Start nginx] **************************************************************************************************
ok: [web1]

PLAY RECAP **********************************************************************************************************
web1                       : ok=3    changed=1    unreachable=0    failed=0    skipped=0    rescued=0    ignored
=0

esadmin@pod70764-AnsibleMaster:~/lab/ansible-tutorial/lab-04$
```

You may observe the task Install nginx indicate as changed. Which implies that ansible has installed the nginx on the server.

Since Ansible is idempotent, if we run the playbook once again, ansible will skip the installation and proceed with rest of the playbook step. This feature of any configuration management tool helps to ensure there will not be any issues especially when replacing the configuration or files of remote system.

```
esadmin@pod70764-AnsibleMaster:~/lab/ansible-tutorial/lab-04$ ansible-playbook install-nginx.yml

PLAY [webservers] ***************************************************************************************************

TASK [Gathering Facts] **********************************************************************************************
ok: [web1]

TASK [Install nginx] ************************************************************************************************
 [WARNING]: Could not find aptitude. Using apt-get instead

ok: [web1]

TASK [Start nginx] **************************************************************************************************
ok: [web1]

PLAY RECAP **********************************************************************************************************
web1                       : ok=3    changed=0    unreachable=0    failed=0    skipped=0    rescued=0    ignored
=0

esadmin@pod70764-AnsibleMaster:~/lab/ansible-tutorial/lab-04$
```

Install nginx on webservers and jdk on app server

```
esadmin@pod70764-AnsibleMaster:~/lab/ansible-tutorial/lab-04$ cat install-nginx-and-jdk.yml
---
- hosts: webservers
  become: true
  tasks:
  - name: Install nginx
    apt: name=nginx state=present
  - name: Start nginx
    service: name=nginx state=started enabled=yes
- hosts: appservers
  become: true
  tasks:
  - name: Install default-jdk
    apt: name=default-jdk state=present

esadmin@pod70764-AnsibleMaster:~/lab/ansible-tutorial/lab-04$
```

As you can see in below screenshot, nginx installation was not performed as it was already present on webserver. However, JDK was installed on app server. The play recap will summaries the status of changes which was performed by the playbook.

```
esadmin@pod70764-AnsibleMaster:~/lab/ansible-tutorial/lab-04$ ansible-playbook install-nginx-and-jdk.yml

PLAY [webservers] ***************************************************************************************************

TASK [Gathering Facts] **********************************************************************************************
ok: [web1]

TASK [Install nginx] ************************************************************************************************
 [WARNING]: Could not find aptitude. Using apt-get instead

ok: [web1]

TASK [Start nginx] **************************************************************************************************
ok: [web1]

PLAY [appservers] ***************************************************************************************************

TASK [Gathering Facts] **********************************************************************************************
ok: [app1]

TASK [Install default-jdk] ******************************************************************************************
 [WARNING]: Updating cache and auto-installing missing dependency: python-apt

changed: [app1]

PLAY RECAP **********************************************************************************************************
app1                       : ok=2    changed=1    unreachable=0    failed=0    skipped=0    rescued=0    ignored
=0
web1                       : ok=3    changed=0    unreachable=0    failed=0    skipped=0    rescued=0    ignored
=0

esadmin@pod70764-AnsibleMaster:~/lab/ansible-tutorial/lab-04$
```

Lab-05 – Running Playbook with roles

Change directory to lab-05.

We a playbook with roles which performed the below activity.

- ntp: installs the ntp on webservers.
- nginx – install nginx on webservers and updates the default configuration
- deploy_static_content – copies the zip files containing static html and unzips on webservers under /opt/html directory

```
esadmin@pod70764-AnsibleMaster:~/lab/ansible-tutorial/lab-05$ cat site.yml
---
- hosts: webservers
  become: true

  vars_files:
    - "environments/dev.yml"

  roles:
    - ntp
    - nginx
    - deploy_static_content

esadmin@pod70764-AnsibleMaster:~/lab/ansible-tutorial/lab-05$
```

.

File structure of the role based playbook

```
esadmin@pod70764-AnsibleMaster:~/lab/ansible-tutorial/lab-05$ tree ../lab-05/
../lab-05/
├── environments
│   ├── dev.yml
│   ├── prod.yml
│   └── test.yml
├── roles
│   ├── deploy_static_content
│   │   ├── files
│   │   │   └── static_content.zip
│   │   └── tasks
│   │       └── main.yml
│   ├── nginx
│   │   ├── files
│   │   │   └── default.conf
│   │   ├── handlers
│   │   │   └── main.yml
│   │   └── tasks
│   │       └── main.yml
│   └── ntp
│       └── tasks
│           └── main.yml
└── site.yml

11 directories, 10 files
esadmin@pod70764-AnsibleMaster:~/lab/ansible-tutorial/lab-05$
```

Tasks on Roles - explained :

Each role will have tasks which details what is shall it performed on mentioned host. Here are the three roles which were used in our playbook.

```
esadmin@pod70764-AnsibleMaster:~/lab/ansible-tutorial/lab-05$ cat roles/ntp/tasks/main.yml
---
- name: Install ntp service
  apt: name=ntp state=present

- name: Start ntp service
  service: name=ntp state=started enabled=yes

esadmin@pod70764-AnsibleMaster:~/lab/ansible-tutorial/lab-05$ cat roles/nginx/tasks/main.yml
---
- name: Install nginx server
  apt: name=nginx state=present

- name: Create static content directory
  file: path=/opt/html state=directory owner=www-data group=www-data

- name: Start nginx
  service: name=nginx state=started enabled=yes

- name: Update nginx default config
  copy: src=default.conf dest=/etc/nginx/sites-enabled/default
  notify:
    - Test nginx configuration
    - Reload nginx configuration

- name: Flush handlers
  meta: flush_handlers


esadmin@pod70764-AnsibleMaster:~/lab/ansible-tutorial/lab-05$ cat roles/deploy_static_content/tasks/main.yml
---
- name: Install unzip
  apt: name=unzip state=present

- name: Ensure static content directory exists
  file: path=/opt/html state=directory

- name: Get static content
  shell: ls -1 /opt/html
  register: content

- name: Undeploy previous version of static content
  file: path=/opt/html/{{item}} state=absent
  with_items: "{{content.stdout_lines}}"

- name: Deploy static content
  unarchive: src={{static_content_file_name}} dest=/opt/html/

esadmin@pod70764-AnsibleMaster:~/lab/ansible-tutorial/lab-05$
```

Running the playbook with roles:

Running playbook with roles is as similar as any regular playbook. However, we need to ensure the file structure is been followed so that Ansible can look up the roles and performed tasks as defined in each role. Here is the screenshots of playbook run for your reference.

```
esadmin@pod70764-AnsibleMaster:~/lab/ansible-tutorial/lab-05$ ansible-playbook site.yml

PLAY [webservers] ***************************************************************************************

TASK [Gathering Facts] **********************************************************************************
ok: [web1]

TASK [ntp : Install ntp service] ************************************************************************
 [WARNING]: Could not find aptitude. Using apt-get instead

changed: [web1]

TASK [ntp : Start ntp service] **************************************************************************
ok: [web1]

TASK [nginx : Install nginx server] *********************************************************************
ok: [web1]

TASK [nginx : Create static content directory] ***********************************************************
changed: [web1]

TASK [nginx : Start nginx] ******************************************************************************
ok: [web1]

TASK [nginx : Update nginx default config] ***************************************************************
changed: [web1]

RUNNING HANDLER [nginx : Test nginx configuration] ******************************************************
changed: [web1]

RUNNING HANDLER [nginx : Reload nginx configuration] ****************************************************
changed: [web1]


TASK [deploy_static_content : Install unzip] ************************************************************
changed: [web1]

TASK [deploy_static_content : Ensure static content directory exists] ***********************************
ok: [web1]

TASK [deploy_static_content : Get static content] *******************************************************
changed: [web1]

TASK [deploy_static_content : Undeploy previous version of static content] ******************************

TASK [deploy_static_content : Deploy static content] ****************************************************
changed: [web1]

PLAY RECAP **********************************************************************************************
web1                       : ok=13   changed=8    unreachable=0    failed=0    skipped=1    rescued=0    ignored
=0

esadmin@pod70764-AnsibleMaster:~/lab/ansible-tutorial/lab-05$
```