First Step:

Start the worklog file using script command. This would capture your handson activities for today.

```
osgdev@TG-DevOps-OS004:~/WorkLog$ script Day11_wl.log
Script started, file is Day11_wl.log
```

---

1. Check whether Ansible is available on your machine

```
osgdev@TG-DevOps-OS004:~$ ansible --version
ansible 2.4.2.0
  config file = /etc/ansible/ansible.cfg
  configured module search path =
[u'/home/osgdev/.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, Dec  4 2017, 14:50:18) [GCC 5.4.0
20160609]
```

---

2. Visit Global Ansible Configuration File, which applies to use of Ansible anywhere in the machine

```
Some Important Lines in this ansible.cfg file:

osgdev@TG-DevOps-OS004:~$ vi /etc/ansible/ansible.cfg


Following line defines the location of Inventory File:
#inventory      = /etc/ansible/hosts

Following line defines the location of Roles:
#roles_path    = /etc/ansible/roles

Following line defines the location of Log File:
#log_path = /var/log/ansible.log
```

---

3. Visit Glbal inventory file available at location /etc/ansible/hosts as defined in the inventory file

```
osgdev@TG-DevOps-OS004:~$ vi /etc/ansible/hosts
```

Following is the way servers to be configured can be listed:
Note: All entries are commented here, hence can only be used as reference.

```
# Ex 1: Ungrouped hosts, specify before any group headers.

## green.example.com
## blue.example.com
## 192.168.100.1
## 192.168.100.10

# Ex 2: A collection of hosts belonging to the 'webservers' group

## [webservers]
## alpha.example.org
## beta.example.org
## 192.168.1.100
## 192.168.1.110

# Ex 3: A collection of database servers in the 'dbservers' group

## [dbservers]
##
## db01.intranet.mydomain.net
## db02.intranet.mydomain.net
## 10.25.1.56
## 10.25.1.57

# If you have multiple hosts following a pattern you can specify
# them like this:

## www[001:006].example.com

# Here's another example of host ranges, this time there are no
# leading 0s:

## db-[99:101]-node.example.com
```

---

4. For accessing and configuring a set of servers, it is always better to create a local ansible configuration file and inventory file for better management of playbooks.

Location of Home folder

```
osgdev@TG-DevOps-OS004:~$ pwd
/home/osgdev
```

Let us now create a folder:

```
osgdev@TG-DevOps-OS004:~$ mkdir ansilab
osgdev@TG-DevOps-OS004:~$ cd ansilab/
osgdev@TG-DevOps-OS004:~/ansilab$
```

---

5. Let us "ping" the local machine.

The local machine on which we are now working is referred as "localhost" in the /etc/hosts file

```
osgdev@TG-DevOps-OS004:~/ansilab$ cat /etc/hosts
127.0.0.1   localhost


osgdev@TG-DevOps-OS004:~/ansilab$ ansible all -i 'localhost,' -m ping -c
local
localhost | SUCCESS => {
    "changed": false,
    "ping": "pong"
}
```

Note: This step is actually pushing a "ping" ansible module into local machine and on successful working the module we are getting "pong" response. The module has not made any change to the environment of local machine. Hence "changed": false,

---

6. Let us create a local inventory file with a group called "local" and server called "localhost"

```
osgdev@TG-DevOps-OS004:~/ansilab$ cat ansiserver
[local]
localhost
```

Note: You may use any name for local inventory file

---

7. Let us ping the local machine again using its entry available in the local inventory file

```
osgdev@TG-DevOps-OS004:~/ansilab$ ansible all -i
'/home/osgdev/ansilab/ansiserver' -m ping -c local
localhost | SUCCESS => {
    "changed": false,
    "ping": "pong"
}
```

---

8. In the command used in previous step the word "all" refers to all the servers listed in local invenroty file. Hence let us be specific to server "localhost" or to group "local"

```
osgdev@TG-DevOps-OS004:~/ansilab$ ansible localhost -i
'/home/osgdev/ansilab/ansiserver' -m ping -c local
localhost | SUCCESS => {
```

```
    "changed": false,
    "ping": "pong"
}

osgdev@TG-DevOps-OS004:~/ansilab$ ansible local -i
'/home/osgdev/ansilab/ansiserver' -m ping -c local
localhost | SUCCESS => {
    "changed": false,
    "ping": "pong"
}
```

---

9. To avoid specifying the entire path of inventory file, let us specify this path creating a local ansible.cfg file.

Note: Both the inventory file ansiserver and ansible.cfg are applicable only when Ansible is used in this folder. In all other folders the global configuration file /etc/ansible/ansible.cfg and the inventory file specified inside this configuration file /etc/ansible/hosts is applied.

```
osgdev@TG-DevOps-OS004:~/ansilab$ pwd
/home/osgdev/ansilab

osgdev@TG-DevOps-OS004:~/ansilab$ ls
ansible.cfg  ansiserver

oosgdev@TG-DevOps-OS004:~/ansilab$ cat ansible.cfg
[defaults]

inventory = /home/osgdev/ansilab/ansiserver
```

---

10. Now you can avoid specifying the path of inventory file in ansible command

```
osgdev@TG-DevOps-OS004:~/ansilab$ ansible localhost -m ping -c local
localhost | SUCCESS => {
    "changed": false,
    "ping": "pong"
}
```

---

11. In the command we are using "-c local" refers to connection as local machine hence ansible is not looking for ssh connection. If you don't use "-c local", ansible looks for ssh connectivity and fails on pushing the "ping" module to local machine.

```
osgdev@TG-DevOps-OS004:~/ansilab$ ansible localhost -m ping
The authenticity of host 'localhost (::1)' can't be established.
ECDSA key fingerprint is
SHA256:cajiFI0f4f+cYAGtWNyWgZsOmghVjHzgu7HeS/BGRZI.
Are you sure you want to continue connecting (yes/no)? yes
localhost | UNREACHABLE! => {
```

```
    "changed": false,
    "msg": "Failed to connect to the host via ssh: Warning: Permanently
added 'localhost' (ECDSA) to the list of known hosts.\r\nPermission
denied (publickey,password).\r\n",
    "unreachable": true
}
```

---

12. You can provide the required username and password to resolve this.

Note: For all topgear machine that we are using in this training
username(login): osgdev password: osg@1234

```
osgdev@TG-DevOps-OS004:~/ansilab$ ansible localhost -m ping -k -u osgdev
SSH password:
localhost | SUCCESS => {
    "changed": false,
    "ping": "pong"
}

Note: -k prompt for the password and -u specify the username
```

---

13. You may also resolve this by adding this information connection type as "local" in the inventory file:

```
osgdev@TG-DevOps-OS004:~/ansilab$ cat ansiserver
[local]
localhost ansible_connection=local
```

---

14. Now try the same command without using the flag "-c local"

```
osgdev@TG-DevOps-OS004:~/ansilab$ ansible localhost -m ping
localhost | SUCCESS => {
    "changed": false,
    "ping": "pong"
}
```

---

15. Let us try to add the same local machine with its other name provided in /etc/hosts file:

```
osgdev@TG-DevOps-OS004:~/ansilab$ cat /etc/hosts
127.0.0.1   localhost
127.0.1.1   TG-DevOps-OS004.wipro.com TG-DevOps-OS004
```

Modify the local inventory file to add this second server (Same local machine with different name)

```
osgdev@TG-DevOps-OS004:~/ansilab$ cat ansiserver
[local]
localhost ansible_connection=local
TG-DevOps-OS004 hostname=127.0.1.1 ansible_ssh_user=osgdev
```

---

16. If you try this with the ssh connectivity like we did in step 12, it may fail like below, as this ssh connection (to itself) with new name is still not known to local machine.

```
osgdev@TG-DevOps-OS004:~/ansilab$ ansible TG-DevOps-OS004 -m ping -k -u
osgdev
SSH password:
TG-DevOps-OS004 | FAILED! => {
    "msg": "Using a SSH password instead of a key is not possible because
Host Key checking is enabled and sshpass does not support this.  Please
add this host's fingerprint to your known_hosts file to manage this
host."
}
```

---

17. Let us try to ping the local machine first using this new name of local machine itself.

```
osgdev@TG-DevOps-OS004:~/ansilab$ ping TG-DevOps-OS004
PING TG-DevOps-OS004.wipro.com (127.0.1.1) 56(84) bytes of data.
64 bytes from TG-DevOps-OS004.wipro.com (127.0.1.1): icmp_seq=1 ttl=64
time=0.076 ms
64 bytes from TG-DevOps-OS004.wipro.com (127.0.1.1): icmp_seq=2 ttl=64
time=0.044 ms
64 bytes from TG-DevOps-OS004.wipro.com (127.0.1.1): icmp_seq=3 ttl=64
time=0.036 ms
^C
--- TG-DevOps-OS004.wipro.com ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 1998ms
rtt min/avg/max/mdev = 0.036/0.052/0.076/0.017 ms
```

---

18. Try ssh connectivity to the local machine with the new name.

```
osgdev@TG-DevOps-OS004:~/ansilab$ ssh osgdev@TG-DevOps-OS004
The authenticity of host 'tg-devops-os004 (127.0.1.1)' can't be
established.
ECDSA key fingerprint is
SHA256:cajiFI0f4f+cYAGtWNyWgZsOmghVjHzgu7HeS/BGRZI.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added 'tg-devops-os004' (ECDSA) to the list of known
hosts.
osgdev@tg-devops-os004's password:
Welcome to Ubuntu 16.04.3 LTS (GNU/Linux 4.4.0-112-generic x86_64)
```

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

218 packages can be updated.
105 updates are security updates.


Last login: Mon Apr  9 10:49:01 2018 from ::1
osgdev@TG-DevOps-OS004:~$ exit
logout
Connection to tg-devops-os004 closed.
```

Note: This has added new ssh connectivity to local machine with new name in the following "known_hosts" file in the .ssh directory in the home folder.

```
osgdev@TG-DevOps-OS004:~/ansilab$ ls ~/.ssh/
known_hosts
```

---

19. Now if you try with pushing "ping" module to local machine with new server name for the local machine it should work:

Content of your inventory file:

```
osgdev@TG-DevOps-OS004:~/ansilab$ cat ansiserver
[local]
localhost ansible_connection=local
TG-DevOps-OS004 hostname=127.0.1.1 ansible_ssh_user=osgdev

osgdev@TG-DevOps-OS004:~/ansilab$ ansible TG-DevOps-OS004 -m ping -k -u
osgdev
SSH password:
TG-DevOps-OS004 | SUCCESS => {
    "changed": false,
    "ping": "pong"
}
```

Note: You may either use or avoid specifying the username, as ssh connectivity fingerprint in "known_hosts" is already having this information. You may also specify them in inventory file.

```
osgdev@TG-DevOps-OS004:~/ansilab$ ansible TG-DevOps-OS004 -m ping -k
SSH password:
TG-DevOps-OS004 | SUCCESS => {
    "changed": false,
    "ping": "pong"
}
```

---

20. We shall now work on to avoid giving the password. Instead we shall use keypair to get password less access with this ssh connectivity. Normally it require password to get ssh connectivity.

```
osgdev@TG-DevOps-OS004:~/ansilab$ ssh osgdev@TG-DevOps-OS004
osgdev@tg-devops-os004's password:
Welcome to Ubuntu 16.04.3 LTS (GNU/Linux 4.4.0-112-generic x86_64)

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

218 packages can be updated.
105 updates are security updates.


Last login: Mon Apr  9 11:16:36 2018 from 127.0.0.1
osgdev@TG-DevOps-OS004:~$ exit
logout
Connection to tg-devops-os004 closed.
```

Let us generate the keypair with a private and public key. This will go into .ssh folder in the home folder /home/osgdev.

```
osgdev@TG-DevOps-OS004:~/ansilab$ ls ~/.ssh/
known_hosts
```

Note: If there are any previosly existed keypair in the same directory please remove them:

```
osgdev@TG-DevOps-OS004:~/ansilab$ ls ~/.ssh/
id_rsa  id_rsa.pub  known_hosts

osgdev@TG-DevOps-OS004:~/ansilab$ rm ~/.ssh/id_rsa*

osgdev@TG-DevOps-OS004:~/ansilab$ ls ~/.ssh/
known_hosts

osgdev@TG-DevOps-OS004:~/ansilab$ ssh-keygen
Generating public/private rsa key pair.
Enter file in which to save the key (/home/osgdev/.ssh/id_rsa):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/osgdev/.ssh/id_rsa.
Your public key has been saved in /home/osgdev/.ssh/id_rsa.pub.
The key fingerprint is:
SHA256:zzB0qwpZ2TxeKcNjCx1jNp9e1W0kwlW4EQCQmGJxrZo osgdev@TG-DevOps-OS004
The key's randomart image is:
+---[RSA 2048]----+
|    ...+.o.ooo++o|
|    o.o o   ..o+.|
|   . . .B .   .o+|
|      .X * + ... |
|     o+ S * .    |
```

```
|    Eo + # .     |
|    o   + +      |
|     . .         |
|      .          |
+----[SHA256]-----+
```

Note: Now you should be able to see the keypair in ~/.ssh folder:

```
osgdev@TG-DevOps-OS004:~/ansilab$ ls ~/.ssh
id_rsa  id_rsa.pub  known_hosts
```

---

21. Now let us insert the public key into local machine get an authorized connection, where ssh connectivity can be done without password.

```
osgdev@TG-DevOps-OS004:~/ansilab$ ssh-copy-id -i ~/.ssh/id_rsa.pub TG-
DevOps-OS004
/usr/bin/ssh-copy-id: INFO: Source of key(s) to be installed:
"/home/osgdev/.ssh/id_rsa.pub"
/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
prompted now it is to install the new keys
osgdev@tg-devops-os004's password:

Number of key(s) added: 1

Now try logging into the machine, with:   "ssh 'TG-DevOps-OS004'"
and check to make sure that only the key(s) you wanted were added.
```

---

22. You can check the password less access to local machine with new server name.

```
osgdev@TG-DevOps-OS004:~/ansilab$ ssh osgdev@TG-DevOps-OS004
Welcome to Ubuntu 16.04.3 LTS (GNU/Linux 4.4.0-112-generic x86_64)

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

218 packages can be updated.
105 updates are security updates.


Last login: Mon Apr  9 11:21:14 2018 from 127.0.0.1
osgdev@TG-DevOps-OS004:~$ exit
logout
Connection to tg-devops-os004 closed.
```

Note: You may also check a new file "authorized keys" which will be holding this information under ~/.ssh folder.

```
osgdev@TG-DevOps-OS004:~/ansilab$ ls ~/.ssh
authorized_keys  id_rsa  id_rsa.pub  known_hosts
```

---

23. This password less ssh access using keypair will allow you to push ping module without specifying the password.

```
osgdev@TG-DevOps-OS004:~/ansilab$ ansible TG-DevOps-OS004 -m ping
TG-DevOps-OS004 | SUCCESS => {
    "changed": false,
    "ping": "pong"
}

osgdev@TG-DevOps-OS004:~/ansilab$ ansible local -m ping
localhost | SUCCESS => {
    "changed": false,
    "ping": "pong"
}
TG-DevOps-OS004 | SUCCESS => {
    "changed": false,
    "ping": "pong"
}

osgdev@TG-DevOps-OS004:~/ansilab$ ansible all -m ping
localhost | SUCCESS => {
    "changed": false,
    "ping": "pong"
}
TG-DevOps-OS004 | SUCCESS => {
    "changed": false,
    "ping": "pong"
}
```

---

24. You may also try this with another Fully Qualified Domain Name "TG-DevOps-OS004.wipro.com" given to this same machine.

```
osgdev@TG-DevOps-OS004:~/ansilab$ cat /etc/hosts
127.0.0.1   localhost
127.0.1.1   TG-DevOps-OS004.wipro.com TG-DevOps-OS004

osgdev@TG-DevOps-OS004:~/ansilab$ ping TG-DevOps-OS004.wipro.com
PING TG-DevOps-OS004.wipro.com (127.0.1.1) 56(84) bytes of data.
64 bytes from TG-DevOps-OS004.wipro.com (127.0.1.1): icmp_seq=1 ttl=64
time=0.037 ms
64 bytes from TG-DevOps-OS004.wipro.com (127.0.1.1): icmp_seq=2 ttl=64
time=0.052 ms
64 bytes from TG-DevOps-OS004.wipro.com (127.0.1.1): icmp_seq=3 ttl=64
time=0.042 ms
^C
```

```
--- TG-DevOps-OS004.wipro.com ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 1999ms
rtt min/avg/max/mdev = 0.037/0.043/0.052/0.009 ms

osgdev@TG-DevOps-OS004:~/ansilab$ cat ansiserver
[local]
localhost ansible_connection=local
TG-DevOps-OS004 hostname=127.0.1.1 ansible_ssh_user=osgdev
TG-DevOps-OS004.wipro.com hostname=127.0.1.1 ansible_ssh_user=osgdev

osgdev@TG-DevOps-OS004:~/ansilab$ ansible TG-DevOps-OS004.wipro.com -m
ping -k -u osgdev
SSH password:
TG-DevOps-OS004.wipro.com | FAILED! => {
    "msg": "Using a SSH password instead of a key is not possible because
Host Key checking is enabled and sshpass does not support this.  Please
add this host's fingerprint to your known_hosts file to manage this
host."
}

osgdev@TG-DevOps-OS004:~/ansilab$ ssh osgdev@TG-DevOps-OS004.wipro.com
The authenticity of host 'tg-devops-os004.wipro.com (127.0.1.1)' can't be
established.
ECDSA key fingerprint is
SHA256:cajiFI0f4f+cYAGtWNyWgZsOmghVjHzgu7HeS/BGRZI.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added 'tg-devops-os004.wipro.com' (ECDSA) to the
list of known hosts.
Welcome to Ubuntu 16.04.3 LTS (GNU/Linux 4.4.0-112-generic x86_64)

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

218 packages can be updated.
105 updates are security updates.


Last login: Mon Apr  9 11:41:28 2018 from 127.0.0.1
osgdev@TG-DevOps-OS004:~$ exit
logout
Connection to tg-devops-os004.wipro.com closed.
```

Note: It didn't prompted for the password, but only got the fingerprint added to known_hosts
Now you will be able to push the ping module.

```
osgdev@TG-DevOps-OS004:~/ansilab$ ansible TG-DevOps-OS004.wipro.com -m
ping -k -u osgdev
SSH password:
TG-DevOps-OS004.wipro.com | SUCCESS => {
    "changed": false,
    "ping": "pong"
}
```

Since the key is already added for the hostname "TG-DevOps-OS004" the same can also be used for password less access with the FQDN "TG-DevOps-OS004.wipro.com"

```
osgdev@TG-DevOps-OS004:~/ansilab$ ssh osgdev@TG-DevOps-OS004.wipro.com
Welcome to Ubuntu 16.04.3 LTS (GNU/Linux 4.4.0-112-generic x86_64)

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

218 packages can be updated.
105 updates are security updates.


Last login: Mon Apr  9 14:39:44 2018 from 127.0.0.1
osgdev@TG-DevOps-OS004:~$ exit
logout
Connection to tg-devops-os004.wipro.com closed.

osgdev@TG-DevOps-OS004:~/ansilab$ ansible TG-DevOps-OS004.wipro.com -m
ping
TG-DevOps-OS004.wipro.com | SUCCESS => {
    "changed": false,
    "ping": "pong"
}
osgdev@TG-DevOps-OS004:~/ansilab$ ansible all -m ping
localhost | SUCCESS => {
    "changed": false,
    "ping": "pong"
}
TG-DevOps-OS004.wipro.com | SUCCESS => {
    "changed": false,
    "ping": "pong"
}
TG-DevOps-OS004 | SUCCESS => {
    "changed": false,
    "ping": "pong"
}
```

---

25. To check the applicability of hosts from inventory file

```
osgdev@TG-DevOps-OS004:~/ansilab$ cat ansiserver
[local]
localhost ansible_connection=local
TG-DevOps-OS004 hostname=127.0.1.1 ansible_ssh_user=osgdev
TG-DevOps-OS004.wipro.com hostname=127.0.1.1 ansible_ssh_user=osgdev

osgdev@TG-DevOps-OS004:~/ansilab$ ansible --list-hosts all
  hosts (3):
    localhost
    TG-DevOps-OS004
```

```
    TG-DevOps-OS004.wipro.com

osgdev@TG-DevOps-OS004:~/ansilab$ ansible --list-hosts local
  hosts (3):
    localhost
    TG-DevOps-OS004
    TG-DevOps-OS004.wipro.com

osgdev@TG-DevOps-OS004:~/ansilab$ ansible --list-hosts localhost
  hosts (1):
    localhost
```

Last Step:

Execute the “exit” command to get the script done to generate log file. Push the file to remote repository “DevOpsTools” in your account.

```
osgdev@TG-DevOps-OS004:~$ exit
exit
Script done, file is Day11_wl.log
```

Now your worklog file is ready.

```
osgdev@TG-DevOps-OS004:~/WorkLog$ ls
Day1_wl.log

osgdev@TG-DevOps-OS004:~/WorkLog$ ls -a
.  ..  Day11_wl.log  .git
```

Stage the worklog file of today.

```
osgdev@TG-DevOps-OS004:~/WorkLog$ git add Day11_wl.log
```

Commit the file to local repository.

```
osgdev@TG-DevOps-OS004:~/WorkLog$ git commit -m "Worklog for Day11"
```

Push the file to remote repository in your account.

```
osgdev@TG-DevOps-OS004:~/WorkLog$ git push -u worklog master
```

Check whether the file is available in the remote repository in your account.