First Step:

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

```
osgdev@TG-DevOps-OS004:~/WorkLog$ script Day12_wl.log
Script started, file is Day2_wl.log
```

1. Let us enable log generation in the ansible.cfg file which will record all activities

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

inventory = /home/osgdev/ansilab/ansiserver
log_path = /home/osgdev/ansilab/ansible.log

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 localhost -m ping
localhost | SUCCESS => {
    "changed": false,
    "ping": "pong"
}

osgdev@TG-DevOps-OS004:~/ansilab$ cat ansible.log
2018-04-09 15:34:48,318 p=10702 u=osgdev |  localhost | SUCCESS => {
    "changed": false,
    "ping": "pong"
}
```

---

2. Let us try with few more ansible modules

Command Module (-m command) with a command as argument (-a 'echo "Hello World"')

```
osgdev@TG-DevOps-OS004:~/ansilab$ ansible localhost -m command -a 'echo
"Hello World"'
localhost | SUCCESS | rc=0 >>
Hello World
```

Alternately you may use just the argument (-a hostname) without mentioning "-m command"

```
osgdev@TG-DevOps-OS004:~/ansilab$ ansible localhost -a 'echo "Hello
World"'
```

```
localhost | SUCCESS | rc=0 >>
Hello World

osgdev@TG-DevOps-OS004:~/ansilab$ ansible localhost -a 'hostname'
localhost | SUCCESS | rc=0 >>
TG-DevOps-OS004

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

---

3. Shell command from shell module

```
osgdev@TG-DevOps-OS004:~/ansilab$ ansible localhost -m shell -a 'cat
/etc/resolv.conf'
localhost | SUCCESS | rc=0 >>
# Dynamic resolv.conf(5) file for glibc resolver(3) generated by
resolvconf(8)
#     DO NOT EDIT THIS FILE BY HAND -- YOUR CHANGES WILL BE OVERWRITTEN
nameserver 10.198.50.100
nameserver 10.198.50.200
search wipro.com
```

---

4. Create a File using touch command

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

osgdev@TG-DevOps-OS004:~/ansilab$ ansible localhost -m shell -a 'touch
/home/osgdev/ansilab/testfile'
 [WARNING]: Consider using file module with state=touch rather than
running
touch

localhost | SUCCESS | rc=0 >>


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

Note: Warning to avoid using commands instead to switch to ansible modules

---

5. To create a file:

```
osgdev@TG-DevOps-OS004:~/ansilab$ ansible localhost -m file -a
'path=/home/osgdev/ansilab/testfile state=touch'
localhost | SUCCESS => {
    "changed": true,
```

```
    "dest": "/home/osgdev/ansilab/testfile,",
    "gid": 999,
    "group": "docker",
    "mode": "0644",
    "owner": "osgdev",
    "size": 0,
    "state": "file",
    "uid": 1000
}
```

---

6. To check availability of file:

```
osgdev@TG-DevOps-OS004:~/ansilab$ ansible localhost -m file -a
'path=/home/osgdev/ansilab/testfile state=file'
localhost | SUCCESS => {
    "changed": false,
    "gid": 999,
    "group": "docker",
    "mode": "0644",
    "owner": "osgdev",
    "path": "/home/osgdev/ansilab/testfile,",
    "size": 0,
    "state": "file",
    "uid": 1000
}
```

---

7. Creating a directory:

```
osgdev@TG-DevOps-OS004:~/ansilab$ ansible localhost -m file -a
'path=/home/osgdev/ansilab/resource state=directory'
localhost | SUCCESS => {
    "changed": true,
    "gid": 999,
    "group": "docker",
    "mode": "0755",
    "owner": "osgdev",
    "path": "/home/osgdev/ansilab/resource",
    "size": 4096,
    "state": "directory",
    "uid": 1000
}

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

---

8. To remove the file:

```
osgdev@TG-DevOps-OS004:~/ansilab$ ansible localhost -m file -a
'path=/home/osgdev/ansilab/testfile state=absent'
localhost | SUCCESS => {
    "changed": true,
    "path": "/home/osgdev/ansilab/testfile",
    "state": "absent"
}

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

---

9. To remove directory:

```
osgdev@TG-DevOps-OS004:~/ansilab$ ansible localhost -m file -a
'path=/home/osgdev/ansilab/resource state=absent'
localhost | SUCCESS => {
    "changed": true,
    "path": "/home/osgdev/ansilab/resource",
    "state": "absent"
}

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

---

10. To copy the file:

```
osgdev@TG-DevOps-OS004:~/ansilab$ ansible localhost -m copy -a
'src=/home/osgdev/ansilab/ansible.log
dest=/home/osgdev/ansilab/ansible1.txt'
localhost | SUCCESS => {
    "changed": true,
    "checksum": "8162e38cd69c34e1893d5ba9995ad622ed306e67",
    "dest": "/home/osgdev/ansilab/ansible1.txt",
    "gid": 999,
    "group": "docker",
    "md5sum": "b4970987a31d37955f08543b5151357f",
    "mode": "0644",
    "owner": "osgdev",
    "size": 3520,
    "src": "/home/osgdev/.ansible/tmp/ansible-tmp-1523271573.29-
223739787954780/source",
    "state": "file",
    "uid": 1000
}

osgdev@TG-DevOps-OS004:~/ansilab$ ls
ansible1.txt  ansible.cfg  ansible.log  ansiserver
```

---

11. Following module is to unarchive a tar.gz file. For convenience either use the resource folder created before (if not deleted) otherwise create a folder.

Note: This command is using group of tar files available in the topgear machine provided for the training purpose:

```
osgdev@TG-DevOps-OS004:~/ansilab$ ls /home/osgdev/myDownloads/
apache-tomcat-7.0.82.tar.gz                  gedit_3.10.4-
0ubuntu4_amd64.deb
apache-tomcat-8.5.27-src.tar.gz              jenkins.tar
apache-tomcat-8.5.27.tar.gz                  jenkins.war
eclipse-inst-linux64.tar.gz                  uex-16.1.0.22_amd64.tar.gz
eclipse-jee-oxygen-2-linux-gtk-x86_64.tar.gz

osgdev@TG-DevOps-OS004:~/ansilab$ mkdir /home/osgdev/resource

osgdev@TG-DevOps-OS004:~/ansilab$ ansible localhost -m unarchive -a
'src=/home/osgdev/myDownloads/apache-tomcat-8.5.27.tar.gz
dest=/home/osgdev/resource'
localhost | SUCCESS => {
    "changed": true,
    "dest": "/home/osgdev/resource",
    "extract_results": {
        "cmd": [
            "/bin/tar",
            "--extract",
            "-C",
            "/home/osgdev/resource",
            "-z",
            "-f",
            "/home/osgdev/.ansible/tmp/ansible-tmp-1523272284.52-
130830733633323/source"
        ],
        "err": "",
        "out": "",
        "rc": 0
    },
    "gid": 999,
    "group": "docker",
    "handler": "TgzArchive",
    "mode": "0755",
    "owner": "osgdev",
    "size": 4096,
    "src": "/home/osgdev/.ansible/tmp/ansible-tmp-1523272284.52-
130830733633323/source",
    "state": "directory",
    "uid": 1000
}

osgdev@TG-DevOps-OS004:~/ansilab$ ls /home/osgdev/resource
apache-tomcat-8.5.27
```

12. Make a copy of server.conf file at the path ~/resource/apache-tomcat-8.5.27/conf to ~/resource to change the working port for tomcat server.

```
osgdev@TG-DevOps-OS004:~/ansilab$ ansible localhost -m copy -a
'src=/home/osgdev/resource/apache-tomcat-8.5.27/conf/server.xml
dest=/home/osgdev/resource/server.xml.j2'
localhost | SUCCESS => {
    "changed": true,
    "checksum": "c46bef72fe29a59d93244b889f130e28ee575df5",
    "dest": "/home/osgdev/resource/server.xml.j2",
    "gid": 999,
    "group": "docker",
    "md5sum": "1a1db053f27ddddebd261c4b1d8c534c",
    "mode": "0644",
    "owner": "osgdev",
    "size": 7511,
    "src": "/home/osgdev/.ansible/tmp/ansible-tmp-1523281328.0-
277474787736770/source",
    "state": "file",
    "uid": 1000
}
```

Manually edit the following line number 69. Change the Connector port=9095

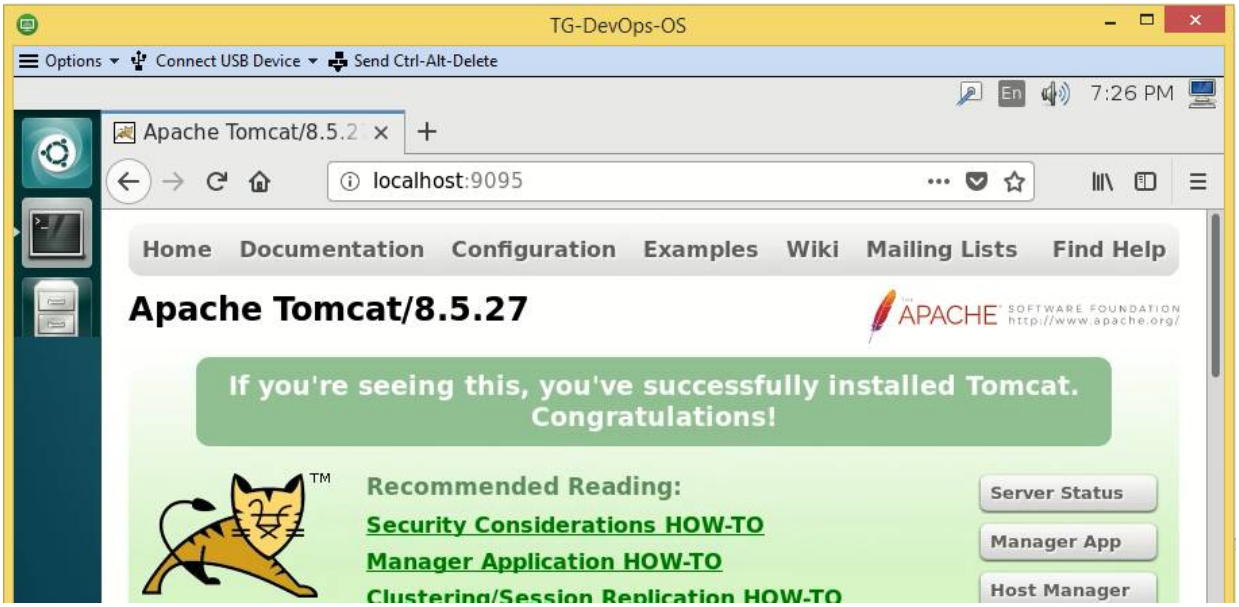
```
   <Connector port="8080" protocol="HTTP/1.1"
              connectionTimeout="20000"
              redirectPort="8443" />
```

Use the copy module again to copy the modified file back to original location.

```
osgdev@TG-DevOps-OS004:~/ansilab$ ansible localhost -m copy -a
'src=/home/osgdev/resource/server.xml.j2
dest=/home/osgdev/resource/apache-tomcat-8.5.27/conf/server.xml'
localhost | SUCCESS => {
    "changed": true,
    "checksum": "ca5761ea0ff20e47df81a758de86bb189a41c046",
    "dest": "/home/osgdev/resource/apache-tomcat-8.5.27/conf/server.xml",
    "gid": 999,
    "group": "docker",
    "md5sum": "af0c9bb3b165cb3f5a0aa07943502c2a",
    "mode": "0600",
    "owner": "osgdev",
    "size": 7511,
    "src": "/home/osgdev/.ansible/tmp/ansible-tmp-1523281876.4-
164128969797725/source",
    "state": "file",
    "uid": 1000
}
```

13. Use the shell module to start the tomcat server

```
osgdev@TG-DevOps-OS004:~/ansilab$ ansible localhost -m shell -a
'/home/osgdev/resource/apache-tomcat-8.5.27/bin/startup.sh'
localhost | SUCCESS | rc=0 >>
Tomcat started.
```



---

14. We shall create ansible playbook for the activities done in previous steps.

To execute echo command module, create a file named echo.yaml with following content.

```
osgdev@TG-DevOps-OS004:~/ansilab$ cat echo.yaml
---
  - hosts: localhost
    tasks:
    - name: Use echo command
      command: 'echo "Hello world"'
```

Execute the playbook using following command:

```
osgdev@TG-DevOps-OS004:~/ansilab$ ansible-playbook -v echo.yaml
Using /home/osgdev/ansilab/ansible.cfg as config file

PLAY [localhost]
*****************************************************************

TASK [Gathering Facts]
*********************************************************
ok: [localhost]
```

```
TASK [Use echo command]
*****************************************************
changed: [localhost] => {"changed": true, "cmd": ["echo", "Hello world"],
"delta": "0:00:00.002799", "end": "2018-04-09 19:58:24.150397", "rc": 0,
"start": "2018-04-09 19:58:24.147598", "stderr": "", "stderr_lines": [],
"stdout": "Hello world", "stdout_lines": ["Hello world"]}

PLAY RECAP
*********************************************************************
localhost                  : ok=2    changed=1    unreachable=0
failed=0
```

To execute hostname command in playbook.

```
osgdev@TG-DevOps-OS004:~/ansilab$ cat hostname.yaml
- hosts: localhost
  tasks:
  - name: Use hostname command
    command: hostname

osgdev@TG-DevOps-OS004:~/ansilab$ ansible-playbook -v hostname.yaml
Using /home/osgdev/ansilab/ansible.cfg as config file

PLAY [localhost]
*****************************************************************

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

TASK [Use echo command]
*****************************************************
changed: [localhost] => {"changed": true, "cmd": ["hostname"], "delta":
"0:00:00.003299", "end": "2018-04-09 20:03:36.161582", "rc": 0, "start":
"2018-04-09 20:03:36.158283", "stderr": "", "stderr_lines": [], "stdout":
"TG-DevOps-OS004", "stdout_lines": ["TG-DevOps-OS004"]}

PLAY RECAP
*********************************************************************
localhost                  : ok=2    changed=1    unreachable=0
failed=0
```

To use shell command to display content of /etc/resolv.conf

```
osgdev@TG-DevOps-OS004:~/ansilab$ cat resolv.yaml
- hosts: localhost
  tasks:
  - name: Use shell command
    shell: cat /etc/resolv.conf

osgdev@TG-DevOps-OS004:~/ansilab$ ansible-playbook resolv.yaml

PLAY [localhost]
*****************************************************************
```

```
TASK [Gathering Facts]
*********************************************************
ok: [localhost]

TASK [Use shell command]
*********************************************************
changed: [localhost]

PLAY RECAP
*********************************************************************
localhost                  : ok=2    changed=1    unreachable=0
failed=0

osgdev@TG-DevOps-OS004:~/ansilab$ ansible-playbook -v resolv.yaml
Using /home/osgdev/ansilab/ansible.cfg as config file

PLAY [localhost]
*************************************************************

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

TASK [Use shell command]
*********************************************************
changed: [localhost] => {"changed": true, "cmd": "cat /etc/resolv.conf",
"delta": "0:00:00.002900", "end": "2018-04-09 20:07:24.945019", "rc": 0,
"start": "2018-04-09 20:07:24.942119", "stderr": "", "stderr_lines": [],
"stdout": "# Dynamic resolv.conf(5) file for glibc resolver(3) generated
by resolvconf(8)\n#     DO NOT EDIT THIS FILE BY HAND -- YOUR CHANGES
WILL BE OVERWRITTEN\nnameserver 10.198.50.100\nnameserver
10.198.50.200\nsearch wipro.com", "stdout_lines": ["# Dynamic
resolv.conf(5) file for glibc resolver(3) generated by resolvconf(8)", "#
DO NOT EDIT THIS FILE BY HAND -- YOUR CHANGES WILL BE OVERWRITTEN",
"nameserver 10.198.50.100", "nameserver 10.198.50.200", "search
wipro.com"]}

PLAY RECAP
*********************************************************************
localhost                  : ok=2    changed=1    unreachable=0
failed=0
```

---

15. To create a folder and a file inside the folder.

Note: Ansible Playbook can use a collection of modules in a single yaml file.

```
osgdev@TG-DevOps-OS004:~/ansilab$ cat file.yaml
- hosts: localhost
  tasks:
  - name: To Create a folder
```

```
    file:
      path: /home/osgdev/ansilab/NEWSAMPLE
      state: directory
      mode: 0755

  - name: To Create a file
    file:
      path: /home/osgdev/ansilab/NEWSAMPLE/new_file
      state: touch

osgdev@TG-DevOps-OS004:~/ansilab$ ansible-playbook file.yaml

PLAY [localhost]
*****************************************************************

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

TASK [To Create a folder]
*****************************************************
changed: [localhost]

TASK [To Create a file]
*******************************************************
changed: [localhost]

PLAY RECAP
*********************************************************************
localhost                  : ok=3    changed=2    unreachable=0
failed=0
```

List to see the folder and file created inside the folder

```
osgdev@TG-DevOps-OS004:~/ansilab$ ls
ansible1.txt  ansible.log  echo.yaml  hostname.yaml  resolv.yaml
ansible.cfg   ansiserver   file.yaml  NEWSAMPLE

osgdev@TG-DevOps-OS004:~/ansilab$ ls NEWSAMPLE/
new_file
```

Note: You may try removing the file and folder just by changing state: absent

---

16. Introducing the concept of variables. The activity we did in previous step is using same path for creating both folder and file. This may be used to create a variable representing the path. file.yaml is modified as follows.

```
osgdev@TG-DevOps-OS004:~/ansilab$ cat file1.yaml
- hosts: localhost

  vars:
    folder_path: /home/osgdev/ansilab
```

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

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

osgdev@TG-DevOps-OS004:~/ansilab$ ansible-playbook file1.yaml

PLAY [localhost]
*****************************************************************

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

TASK [To Create a folder]
*****************************************************************
changed: [localhost]

TASK [To Create a file]
*****************************************************************
changed: [localhost]

PLAY RECAP
*****************************************************************
localhost                  : ok=3    changed=2    unreachable=0
failed=0

osgdev@TG-DevOps-OS004:~/ansilab$ ls
ansible1.txt  ansible.log  echo.yaml   file.yaml      NEWSAMPLE
resolv.yaml
ansible.cfg   ansiserver   file1.yaml  hostname.yaml  NEWSAMPLE2

osgdev@TG-DevOps-OS004:~/ansilab$ ls NEWSAMPLE2
new_file2
```

---

17. Introducing the concept of handlers. The activity we did in previous two steps has a built in dependency between two tasks. While the first task creates a folder "NEWSAMPLE", the second task creates a file new_file inside this folder. Unless the first task is executed successfully, the second task is bound to fail. Hence it is essential, the first task should invoke the second task as a handler. The file is modified as follows.

```
osgdev@TG-DevOps-OS004:~/ansilab$ cat file2.yaml
- hosts: localhost
```

```
  vars:
    folder_path: /home/osgdev/ansilab

  tasks:
  - name: To Create a folder
    file:
      path: "{{folder_path}}/NEWSAMPLE3"
      state: directory
      mode: 0755
    notify:
    - To Create a file

  handlers:
  - name: To Create a file
    file:
      path: "{{folder_path}}/NEWSAMPLE3/new_file3"
      state: touch

osgdev@TG-DevOps-OS004:~/ansilab$ ansible-playbook file2.yaml

PLAY [localhost]
*****************************************************************

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

TASK [To Create a folder]
*****************************************************
changed: [localhost]

RUNNING HANDLER [To Create a file]
*****************************************
changed: [localhost]

PLAY RECAP
*********************************************************************
localhost                  : ok=3    changed=2    unreachable=0
failed=0

osgdev@TG-DevOps-OS004:~/ansilab$ ls
ansible1.txt  ansiserver  file2.retry  hostname.yaml  NEWSAMPLE3
ansible.cfg   echo.yaml   file2.yaml   NEWSAMPLE      resolv.yaml
ansible.log   file1.yaml  file.yaml    NEWSAMPLE2

osgdev@TG-DevOps-OS004:~/ansilab$ ls NEWSAMPLE3
new_file3
```

---

18. Let us redo the above activity in the form of structured playbook called roles.

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

```
osgdev@TG-DevOps-OS004:~/ansilab$ ls
ansible1.txt  ansiserver  file2.retry  hostname.yaml  NEWSAMPLE3
ansible.cfg   echo.yaml   file2.yaml   NEWSAMPLE      resolv.yaml
ansible.log   file1.yaml  file.yaml    NEWSAMPLE2     roles
```

---

19. Modify the ansible.cfg file as follows to add the path for roles

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

inventory = /home/osgdev/ansilab/ansiserver
log_path = /home/osgdev/ansilab/ansible.log
roles_path = /home/osgdev/ansilab/roles
```

---

20. Create a role called "sample" and folder structure with separate folders for vars, tasks and handlers under the sample

```
osgdev@TG-DevOps-OS004:~/ansilab$ mkdir ./roles/sample
osgdev@TG-DevOps-OS004:~/ansilab$ tree roles
roles
└── sample

1 directory, 0 files

osgdev@TG-DevOps-OS004:~/ansilab$ mkdir ./roles/sample/vars
osgdev@TG-DevOps-OS004:~/ansilab$ mkdir ./roles/sample/tasks
osgdev@TG-DevOps-OS004:~/ansilab$ mkdir ./roles/sample/handlers
osgdev@TG-DevOps-OS004:~/ansilab$ tree roles
roles
└── sample
    ├── handlers
    ├── tasks
    └── vars

4 directories, 0 files
```

---

21. From the previous activity let us copy the tasks, handlers and vars in separate main.yaml files under each of these folders. vars would go into a main.yaml under vars folder as shown below.

Note: the original contents are taken from file2.yaml used in step 17

```
osgdev@TG-DevOps-OS004:~/ansilab$ vi ./roles/sample/vars/main.yaml
osgdev@TG-DevOps-OS004:~/ansilab$ cat ./roles/sample/vars/main.yaml
folder_path: /home/osgdev/ansilab
```

22. handlers would go into main.yaml under handlers folder as shown below

```
osgdev@TG-DevOps-OS004:~/ansilab$ vi ./roles/sample/handlers/main.yaml
osgdev@TG-DevOps-OS004:~/ansilab$ cat ./roles/sample/handlers/main.yaml
- name: To Create a file
  file:
    path: "{{folder_path}}/NEWSAMPLE4/new_file4"
    state: touch
```

---

23. tasks would go into main.yaml under tasks folder as shown below

```
osgdev@TG-DevOps-OS004:~/ansilab$ vi ./roles/sample/tasks/main.yaml
osgdev@TG-DevOps-OS004:~/ansilab$ cat ./roles/sample/tasks/main.yaml
- name: To Create a folder
  file:
    path: "{{folder_path}}/NEWSAMPLE4"
    state: directory
    mode: 0755
  notify:
    - To Create a file
```

---

24. there will be a common server.yaml file to be invoked as playbook, which inturn would use the tasks, handlers and vars

```
osgdev@TG-DevOps-OS004:~/ansilab$ vi ./roles/sample/server.yaml
osgdev@TG-DevOps-OS004:~/ansilab$ cat ./roles/sample/server.yaml
- hosts: localhost
  roles:
  - sample
```

---

25. Execute ansible playbook

```
osgdev@TG-DevOps-OS004:~/ansilab$ ansible-playbook
./roles/sample/server.yaml

PLAY [localhost]
*****************************************************************

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

TASK [sample : To Create a folder]
*******************************************
```

```
changed: [localhost]

RUNNING HANDLER [sample : To Create a file]
************************************
changed: [localhost]

PLAY RECAP
*********************************************************************
localhost                  : ok=3    changed=2    unreachable=0
failed=0
```

---

26. You may also create this folder structure using ansible-galaxy command

```
osgdev@TG-DevOps-OS004:~/ansilab$ ansible-galaxy init ./roles/newrole
- ./roles/newrole was created successfully
osgdev@TG-DevOps-OS004:~/ansilab$ tree ./roles/newrole/
./roles/newrole/
├── defaults
│   └── main.yml
├── files
├── handlers
│   └── main.yml
├── meta
│   └── main.yml
├── README.md
├── tasks
│   └── main.yml
├── templates
├── tests
│   ├── inventory
│   └── test.yml
└── vars
    └── main.yml

8 directories, 8 files
```

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 Day12_wl.log
```

Now your worklog file is ready.

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

```
Day2_wl.log
```

```
osgdev@TG-DevOps-OS004:~/WorkLog$ ls -a
.  ..  Day12_wl.log  .git
```

Stage the worklog file of today.

```
osgdev@TG-DevOps-OS004:~/WorkLog$ git add Day12_wl.log
```

Commit the file to local repository.

```
osgdev@TG-DevOps-OS004:~/WorkLog$ git commit -m "Worklog for Day12"
```

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.