

Jonathan Ho

jlh5360@rit.edu

Team Bravo

CSEC 473.02

2235 Spring

## **HW2 Installation/Usage Instructions**

NOTE: You should not be facing any problems, but you are then, run this command and enter this password:

```
source jlh5360-openrc.sh
```

```
h3ll()W0rld@gain_
```

1. Ensure you are the user ubuntu:

- If not, then run this command in the terminal:

```
su ubuntu
```

2. Ansible is already installed

- Run the "ansible ---version" command to verify

3. Ensure you are in this path: /home/ubuntu

- If not, then run this command in the terminal:

```
cd ~
```

4. After running the "ls" command, you should see these files:

```
jlh5360_hw2_install_sshd_and_git_linux.yml
```

```
jlh5360_hw2_start_winrm_install_openssh_create_user_windows.yml
```

```
linux_inventory.ini
```

```
win_inventory.ini
```

5. Test your connection/connectivity with Linux (192.168.2.73) based on the inventory file

(linux\_inventory.ini) by running this command and should get this result:

```
ubuntu@test1:~$ ansible linux -i linux_inventory.ini -m ping
```

```
192.168.2.73 | SUCCESS => {
```

```
"ansible_facts": {  
    "discovered_interpreter_python": "/usr/bin/python3"  
},  
"changed": false,  
"ping": "pong"  
}
```

6. Run the Linux playbook by running this command and should/will get this result:

```
ubuntu@test1:~$ ansible-playbook -i linux_inventory.ini  
jlh5360_hw2_install_sshd_and_git_linux.yml
```

PLAY [Install SSHD and Git service]

\*\*\*\*\*

TASK [Gathering Facts]

\*\*\*\*\*

ok: [192.168.2.73]

TASK [Install SSHD]

\*\*\*\*\*

ok: [192.168.2.73]

TASK [Start SSHD]

\*\*\*\*\*

ok: [192.168.2.73]

TASK [Install Git]

\*\*\*\*\*

ok: [192.168.2.73]

PLAY RECAP

\*\*\*\*\*

```
192.168.2.73 : ok=4  changed=0  unreachable=0  failed=0
skipped=0  rescued=0  ignored=0
```

```
ubuntu@test1:~$
```

7. Test your connection/connectivity with Linux (192.168.1.10) based on the inventory file (win\_inventory.ini) by running this command and should get this result:

```
ubuntu@test1:~$ ansible windows -i win_inventory.ini -m win_ping
192.168.1.10 | SUCCESS => {
  "changed": false,
  "ping": "pong"
}
```

8. Run the Windows playbook by running this command and should/will get this result:

```
ubuntu@test1:~$ ansible-playbook -i win_inventory.ini
jlh5360_hw2_start_winrm_install_openssh_create_user_windows.yml
```

```
PLAY [Start WinRM & install OpenSSH & create user]
*****
```

```
TASK [Gathering Facts]
*****
```

```
ok: [192.168.1.10]
```

```
TASK [Start WinRM service]
*****
```

```
ok: [192.168.1.10]
```

```
TASK [Install OpenSSH Server]
*****
```

```
changed: [192.168.1.10]
```

### TASK [Start OpenSSH Server]

\*\*\*\*\*

ok: [192.168.1.10]

### TASK [Create a new user]

\*\*\*\*\*

ok: [192.168.1.10]

### PLAY RECAP

\*\*\*\*\*

192.168.1.10 : ok=5 changed=1 unreachable=0 failed=0  
skipped=0 rescued=0 ignored=0

ubuntu@test1:~\$

9. This concludes the end of how to prepare and then use/run the Ansible playbooks

10. If you want, you can test or check the changes by going to the instances themselves

- Linux (target)

1. Ensure you are the user ubuntu:

- If not, then run this command in the terminal:

su ubuntu

2. Ensure you are in this path: /home/ubuntu

- If not, then run this command in the terminal:

cd ~

3. After running the "systemctl --type=service --state=running | grep ssh" command, you should get this result:

ssh.service	loaded active running OpenBSD
Secure Shell server	

4. Finally, after running "git --version", you should get this result:

git version 2.34.1

- Windows (ansible\_windows)

1. Login the user ansible with these credentials:

Username: ansible

Password: ansible

2. First, you need to check the installment of OpenSSH by going into Windows Powershell and type the command "ssh", and you will then be provided with an overview of how to use the "ssh" command

3. To check if the user testuser has been added, you will need to logout of the user ansible, then you should/will see the user testuser in the list of accounts, and try to login to the user testuser using these credentials:

Username: testuser

Password: testuser

11. This concludes the end of verifying the successfullness of each Ansible playbooks

### Questions

**Answer the following questions about your Ansible scripts. Your answers should be thorough and fully answer the question. Answers should be grammatically correct and be full sentences.**

**1. (8 points) What is the goal of these Ansible scripts? What purpose does it bring to the Gray team effort (or possibly Red or Blue)?**

The goal of these Ansible scripts is to automate the installation and configuration of essential services on both Linux and Windows systems; in my case, I specifically installed the SSHD service and Git as a feature on Linux, and installed OpenSSH service and created a new user (testuser) on Windows. By leveraging Ansible, teams like Gray, Red, or Blue can efficiently manage and deploy configurations across heterogeneous environments (a network connecting computers and other devices where the operating systems and protocols have significant differences), ensuring consistency and reliability in operations. This automation reduces manual intervention, minimizes human error, and accelerates the deployment process. As a result, it enhances the overall productivity and agility in the team's efforts.

**2. (8 points) What was the most challenging aspect of working with Ansible?**

The most challenging aspect of working with Ansible was trying to perform tasks on Windows; I specifically had difficulties trying to install OpenSSH Server and had researched and tried many various attempts. I ultimately succeeded by researching and realizing I can do it by simply using the “ansible.windows.win\_shell” module which allows me to run command in Windows Powershell. All the other tasks of mine for both Linux and Windows were much simpler and easier to complete.

**3. (8 points) If you were to expand this Ansible script, what would you add?**

If I were to expand this Ansible script, I would incorporate more features and services, ideally advanced ones, to cater to diverse use cases. For instance:

- Installing Samba and Apache
- Creating a web and file server
- Implementing role-based access control (RBAC) configurations
- Password changes for root and non-admin accounts
- Edit the contents of a file
- Setup SSH keys