New VM: controller
IP Address: 10.0.5.90
Hostname: controller-seraphim

CONFIGURE STATIC IP ADDRESS

cd /etc/netplan/

sudo nano 00-installer-config.yaml

```
This is the network config written by 'subiquity'
twork:
ethernets:
  ens160:
    dhcp4: false
    addresses: [10.0.5.90/24]
    routes:
      - to: default
        via: 10.0.5.2
    nameservers:
        search: [seraphim.local]
        addresses: [10.0.5.5]_
version: 2
```

sudo netplan try

sudo netplan apply

CONFIGURE HOSTNAME

sudo nano /etc/hostname

```
 GNU nano 6.2                                  /etc/hostname *
controller-seraphim_
```

hostnamectl set-hostname controller-seraphim

sudo nano /etc/cloud/cloud.cfg

```
# This will cause the set+update hostname module to not operate (if true)
preserve_hostname: true
```

CONFIGURE NAMED SUDO USER

sudo adduser seraphim

sudo adduser deployer

sudo visudo

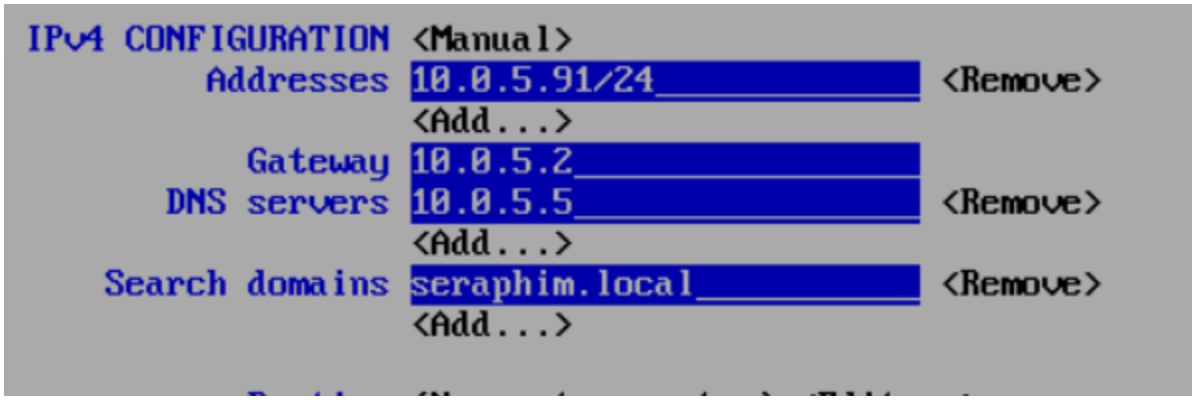
```
# User privilege specification
root    ALL=(ALL:ALL) ALL
seraphim ALL=(ALL:ALL) ALL
deployer ALL=(ALL:ALL) ALL
```

New VM: ansible01
IP Address: 10.0.5.91
Hostname: ansible01-seraphim

New VM: ansible02
IP Address: 10.0.5.92
Hostname: ansible02-seraphim

CONFIGURE STATIC IP ADDRESS

nmtui



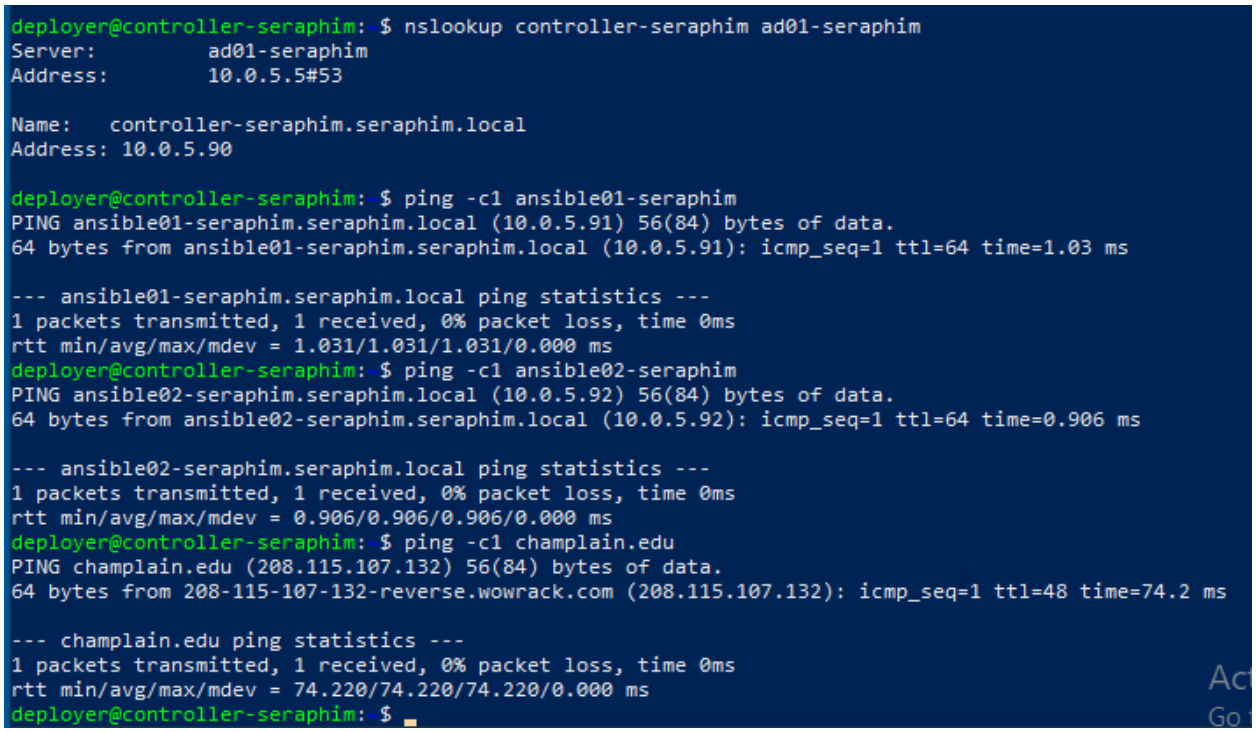
CONFIGURE NAMED SUDO USER

adduser deployer

passwd deployer

usermod -aG wheel deployer

Deliverable 1. A screenshot similar to the one below showing an SSH session from mgmt01 to controller and within that session a DNS lookup for controller against ad01, pinging ansible1, ansible2 and champlain.edu



Deliverable 2. Within your ssh login as a named sudo user, use sudo su - deployer to switch to the deployer user. Provide a screenshot similar to the one below.

```
deployer@controller-seraphim:~$ sudo su - deployer
[sudo] password for deployer:
deployer@controller-seraphim:~$ sudo -i
[sudo] password for deployer:
root@controller-seraphim:~#
```

INSTALLING ANSIBLE

sudo apt install ansible sshpass python3-paramiko

Deliverable 3. Provide a screenshot similar to the one below, indicating a successful ansible installation:

```
root@controller-seraphim:~# ansible --version
ansible 2.10.8
  config file = None
  configured module search path = ['/root/.ansible/plugins/modules', '/usr/share/ansible/plugins/modules']
  ansible python module location = /usr/lib/python3/dist-packages/ansible
  executable location = /usr/bin/ansible
  python version = 3.10.4 (main, Apr  2 2022, 09:04:19) [GCC 11.2.0]
root@controller-seraphim:~#
```


CREATE RSA KEY PAIR

ssh-keygen -t rsa -b 4096

```
deployer@controller-seraphim:~$ ssh-keygen -t rsa -b 4096
Generating public/private rsa key pair.
Enter file in which to save the key (/home/deployer/.ssh/id_rsa):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/deployer/.ssh/id_rsa
Your public key has been saved in /home/deployer/.ssh/id_rsa.pub
The key fingerprint is:
SHA256:/VwP9Rxu/trnlYCIDlV4FOBzJiMcW5EZ5rBCQ/UhMiY deployer@controller-seraphim
The key's randomart image is:
+---[RSA 4096]----+
| E.B.= BO+.      |
|  + = %+o.       |
|   . = O.o     ..|
|    . o B.. . .oo|
|     . .S... .ooo|
|      o    o .+o.|
|       .    o  oo|
|                .+|
```

ssh-copy-id -i /home/deployer/.ssh/id_rsa.pub deployer@10.0.5.91

ssh-copy-id -i /home/deployer/.ssh/id_rsa.pub deployer@10.0.5.92

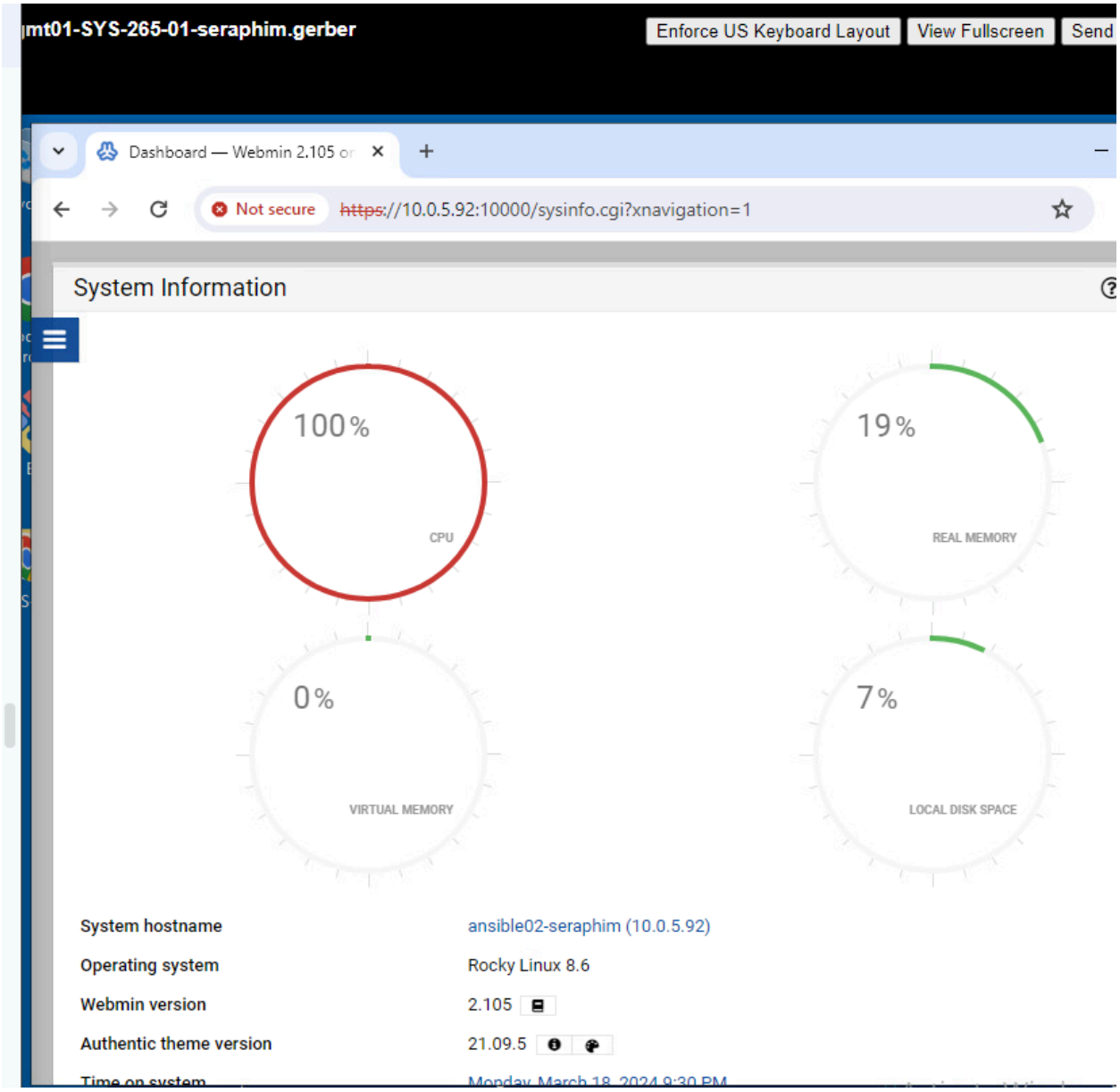
Deliverable 4. Demonstrate passwordless ssh with rsa authentication to both ansible1 and ansible2 from the controller. Provide a screenshot similar to the one below that shows passwordless authentication and then passwordless elevation to root on each system.

```
deployer@controller-seraphim:~$ ssh ansible01-seraphim
Last login: Sun Mar 17 17:01:12 2024 from controller-seraphim.seraphim.local
[deployer@ansible01-seraphim ~]$ sudo -i
[root@ansible01-seraphim ~]# exit
logout
[deployer@ansible01-seraphim ~]$ exit
logout
Connection to ansible01-seraphim closed.
deployer@controller-seraphim:~$ ssh ansible02-seraphim
Activate the web console with: systemctl enable --now cockpit.socket

Last login: Sun Mar 17 17:48:28 2024 from 10.0.5.90
[deployer@ansible02-seraphim ~]$ sudo -i
[root@ansible02-seraphim ~]# exit
logout
[deployer@ansible02-seraphim ~]$ exit
logout
Connection to ansible02-seraphim closed.
deployer@controller-seraphim:~$
```

Deliverable 5. Provide a screenshot of one of your executed commands (not id)

```
deployer@controller-seraphim:~/ansible$ ansible all -a uptime -i inventory.txt
ansible02-seraphim | CHANGED | rc=0 >>
 17:53:38 up 50 min,  1 user,  load average: 0.16, 0.05, 0.01
ansible01-seraphim | CHANGED | rc=0 >>
 17:53:39 up 50 min,  1 user,  load average: 0.33, 0.08, 0.07
deployer@controller-seraphim:~/ansible$
```

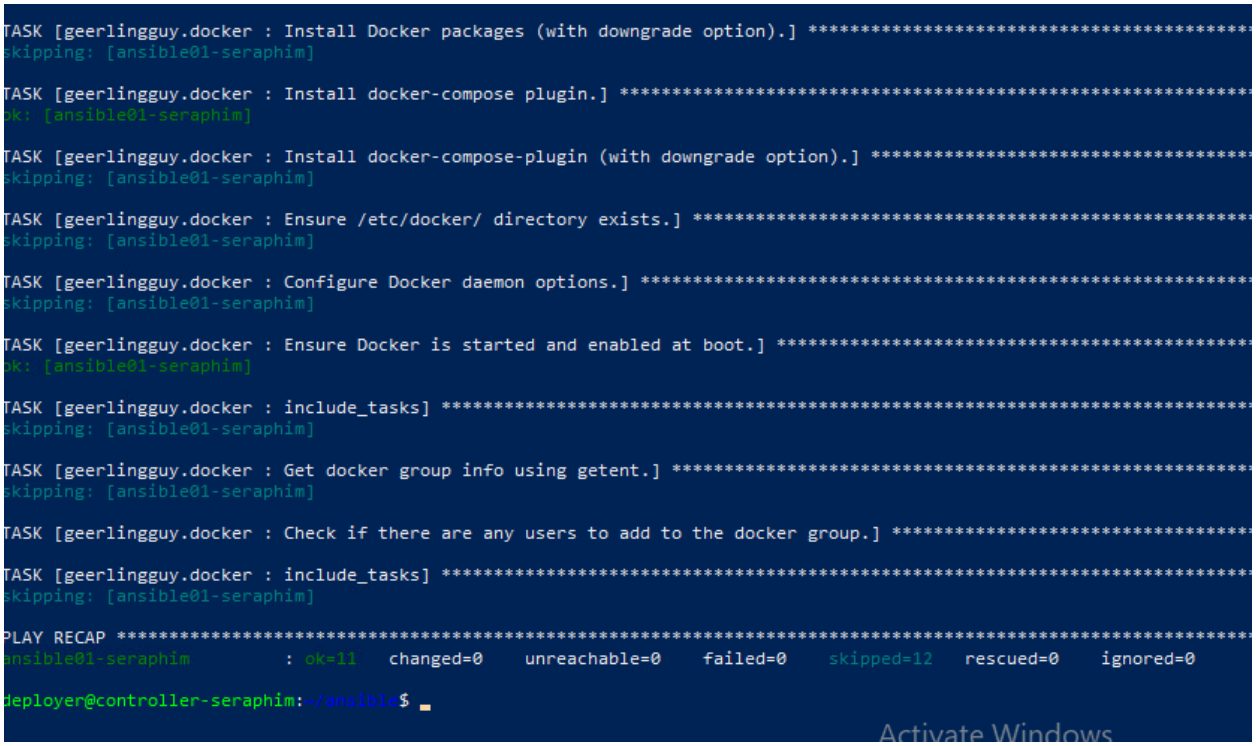
Deliverable 6. Provide a screenshot that shows some aspect of Webmin's logged-in interface like the one shown below:



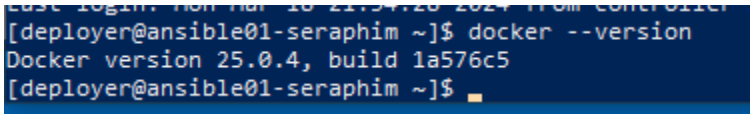
Deliverable 7: Deploy a different role to ansible1. Provide a screenshot of your successful playbook execution

```
deployer@controller-seraphim:~/ansible$ cat roles/docker.yml
---
- name: docker
  hosts: docker
  become: true
  roles:
    - geerlingguy.docker
deployer@controller-seraphim:~/ansible$
```

```
deployer@controller-seraphim:~/ansible$ ansible-playbook -i inventory.txt roles/docker.yml

PLAY [docker] ******************************************************************************************************

TASK [Gathering Facts] *********************************************************************************************
ok: [ansible01-seraphim]

TASK [geerlingguy.docker : Load OS-specific vars.] *****************************************************************
ok: [ansible01-seraphim]

TASK [geerlingguy.docker : include_tasks] **************************************************************************
included: /home/deployer/.ansible/roles/geerlingguy.docker/tasks/setup-RedHat.yml for ansible01-seraphim

TASK [geerlingguy.docker : Ensure old versions of Docker are not installed.] ***************************************
ok: [ansible01-seraphim]

TASK [geerlingguy.docker : Add Docker GPG key.] ********************************************************************
ok: [ansible01-seraphim]

TASK [geerlingguy.docker : Add Docker repository.] *****************************************************************
ok: [ansible01-seraphim]

TASK [geerlingguy.docker : Configure Docker Nightly repo.] *********************************************************
ok: [ansible01-seraphim]

TASK [geerlingguy.docker : Configure Docker Test repo.] ************************************************************
ok: [ansible01-seraphim]

TASK [geerlingguy.docker : Ensure runc is not installed.] **********************************************************
skipping: [ansible01-seraphim]

TASK [geerlingguy.docker : Ensure container-selinux is installed.] *************************************************
skipping: [ansible01-seraphim]

TASK [geerlingguy.docker : Ensure containerd.io is installed.] *****************************************************
skipping: [ansible01-seraphim]

TASK [geerlingguy.docker : include_tasks] **************************************************************************
skipping: [ansible01-seraphim]

TASK [geerlingguy.docker : Install Docker packages.] ***************************************************************
ok: [ansible01-seraphim]
```



Deliverable 8: Provide a screenshot of your new service functionality from a remote client perspective.



CONFIGURE OPENSSH SERVER FOR WINDOWS

Download the latest release of OpenSSH from https://github.com/PowerShell/Win32-OpenSSH/releases on mgmt01.

Unblock-File .\Downloads\OpenSSH-Win64.zip

Expand-Archive .\Downloads\OpenSSH-Win64.zip -DestinationPath .

Copy-Item -Recurse .\OpenSSH-Win64\ 'C:\'

&icacls C:\OpenSSH-Win64\libcrypto.dll /grant Everyone:RX

C:\OpenSSH-Win64\install-sshd.ps1

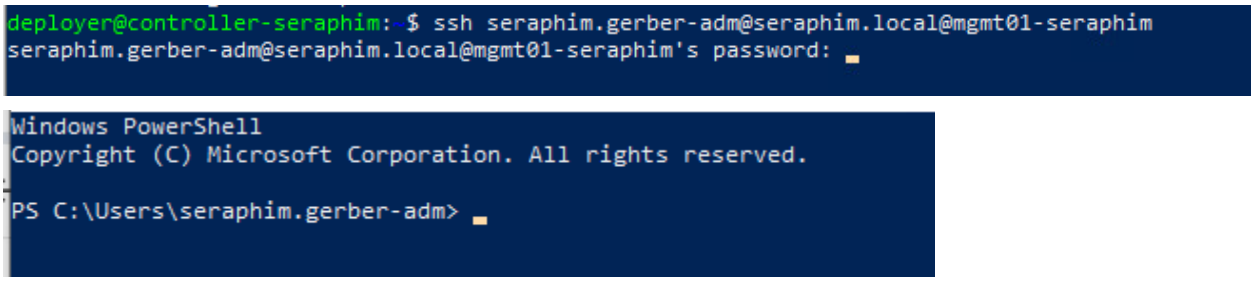
&sc.exe config sshd start= auto

&sc.exe config ssh-agent start= auto

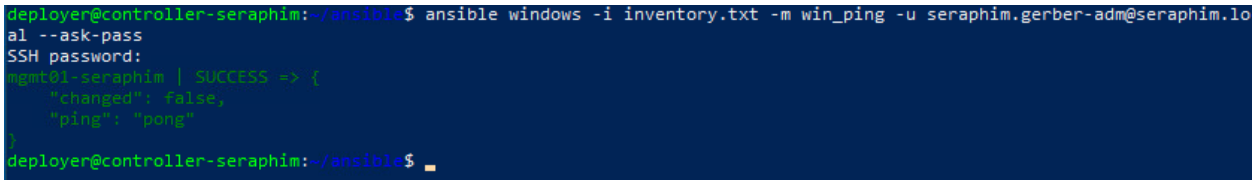
&sc.exe start sshd

&sc.exe start ssh-agent

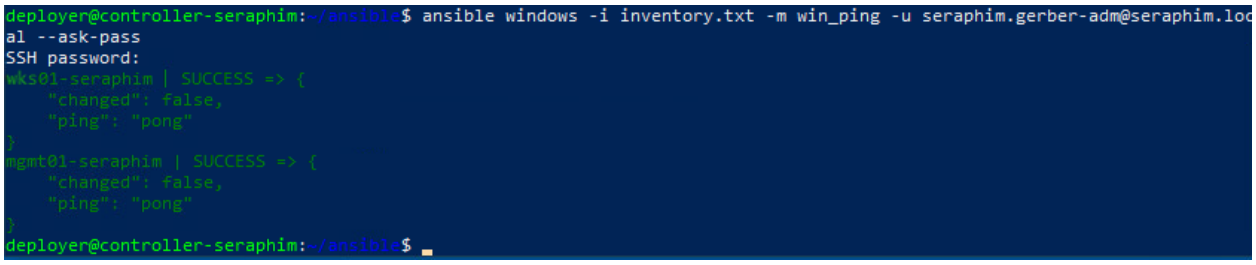
Deliverable 9. Provide a screenshot that shows a successful ssh login to a powershell prompt from controller to mgmt01 similar to the one below.



Deliverable 10. Provide a screenshot similar to the one below that shows a successful win_ping from controller to mgmt01.



Deliverable 11. Rerun the playbook with successful pings on wks1 and mgmt1 similar to the one below



Deliverable 12. Provide a screenshot showing the successful playbook run and software Installation

```
deployer@controller-seraphim:~$ cd ansible/
deployer@controller-seraphim:~/ansible$ ansible-playbook -i inventory.txt roles/windows_software.yml -u seraphim.gerber-
adm@seraphim.local --ask-pass
SSH password:

PLAY [install windows applications] ********************************************************************************

TASK [Gathering Facts] *********************************************************************************************
ok: [mgmt01-seraphim]

TASK [Install Firefox and 7zip] ************************************************************************************
changed: [mgmt01-seraphim]

PLAY RECAP *********************************************************************************************************
mgmt01-seraphim            : ok=2    changed=1    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0

deployer@controller-seraphim:~/ansible$
```

Deliverable 13. Provide a screenshot from an ssh session to mgmt01 that displays installed packages similar to the one below, notepad++ should be there.

```
PS C:\Users\seraphim.gerber-adm> C:\ProgramData\chocolatey\bin\choco.exe list
Chocolatey v2.2.2
7zip 23.1.0
7zip.install 23.1.0
chocolatey 2.2.2
chocolatey-compatibility.extension 1.0.0
chocolatey-core.extension 1.4.0
Firefox 124.0.0
notepadplusplus 8.6.4
notepadplusplus.install 8.6.4
8 packages installed.
PS C:\Users\seraphim.gerber-adm>
```

Deliverable 14. Link to your wiki. You should clearly document the commands used to install ansible on your controller, prepare linux and windows hosts for automation, as well as upload and link your various ansible specific configuration files and playbooks used in the course of this lab.

https://github.com/seraphimgerber/SYS-265