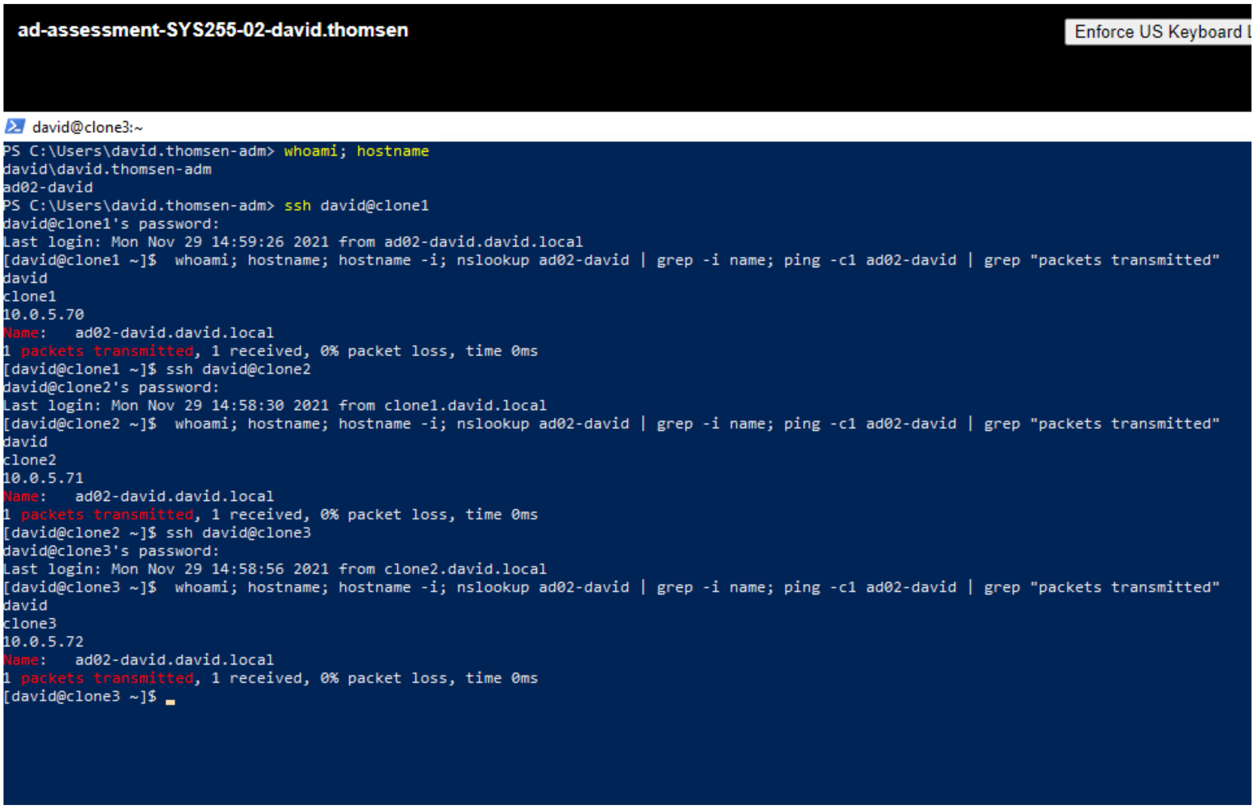
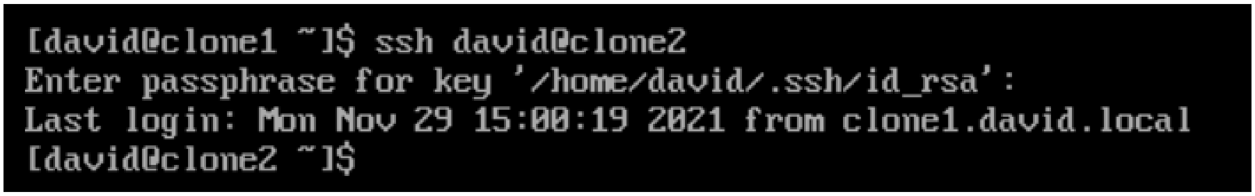
David Thomsen
SYS-255

Deliverable 1. Perform multiple routine testing for connectivity and name resolution via 1-liners similar to the below screen:



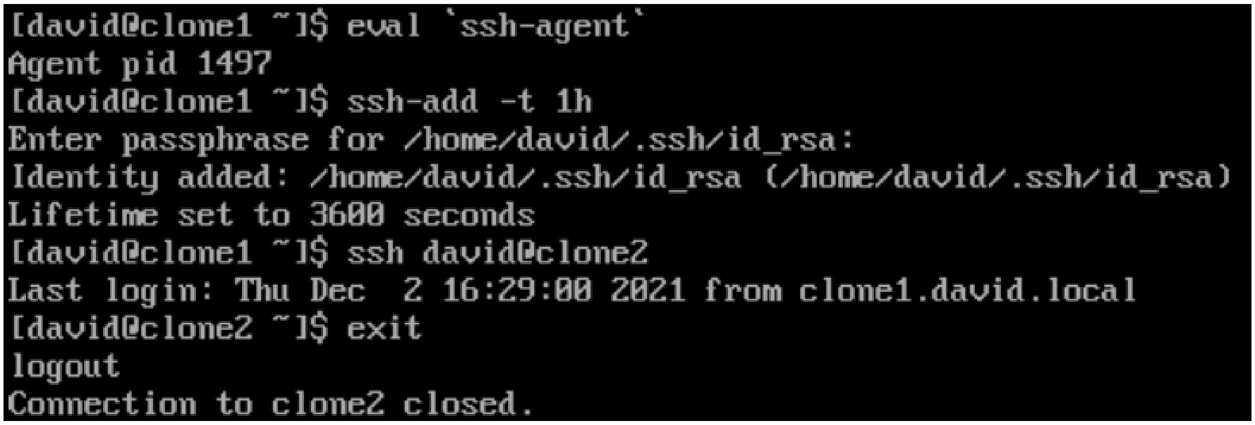
```
PS C:\Users\david.thomsen-adm> whoami; hostname
david\david.thomsen-adm
ad02-david
PS C:\Users\david.thomsen-adm> ssh david@clone1
david@clone1's password:
Last login: Mon Nov 29 14:59:26 2021 from ad02-david.david.local
[david@clone1 ~]$  whoami; hostname; hostname -i; nslookup ad02-david | grep -i name; ping -c1 ad02-david | grep "packets transmitted"
david
clone1
10.0.5.70
Name:   ad02-david.david.local
1 packets transmitted, 1 received, 0% packet loss, time 0ms
[david@clone1 ~]$ ssh david@clone2
david@clone2's password:
Last login: Mon Nov 29 14:58:30 2021 from clone1.david.local
[david@clone2 ~]$  whoami; hostname; hostname -i; nslookup ad02-david | grep -i name; ping -c1 ad02-david | grep "packets transmitted"
david
clone2
10.0.5.71
Name:   ad02-david.david.local
1 packets transmitted, 1 received, 0% packet loss, time 0ms
[david@clone2 ~]$ ssh david@clone3
david@clone3's password:
Last login: Mon Nov 29 14:58:56 2021 from clone2.david.local
[david@clone3 ~]$  whoami; hostname; hostname -i; nslookup ad02-david | grep -i name; ping -c1 ad02-david | grep "packets transmitted"
david
clone3
10.0.5.72
Name:   ad02-david.david.local
1 packets transmitted, 1 received, 0% packet loss, time 0ms
[david@clone3 ~]$ _
```

Deliverable 2. ssh into either clone2 or clone3 using your ssh key. The passphrase you enter is only for unlocking your local private key on clone1, as opposed to logging into the remote system itself. Provide a screenshot that shows the prompt for your passphrase as well as the login into clone2 or 3 that does not ask for a password.



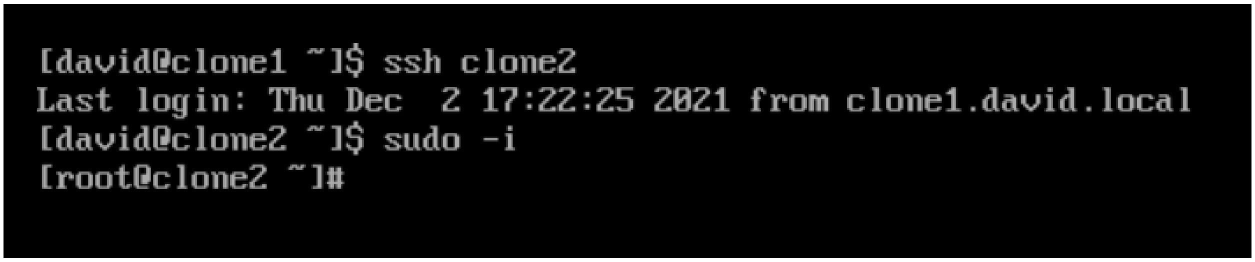
```
[david@clone1 ~]$ ssh david@clone2
Enter passphrase for key '/home/david/.ssh/id_rsa':
Last login: Mon Nov 29 15:00:19 2021 from clone1.david.local
[david@clone2 ~]$
```

Deliverable 3. Provide a screenshot showing passwordless login to clone2 or 3 after having loaded the ssh-agent and private key.



```
[david@clone1 ~]$ eval `ssh-agent`
Agent pid 1497
[david@clone1 ~]$ ssh-add -t 1h
Enter passphrase for /home/david/.ssh/id_rsa:
Identity added: /home/david/.ssh/id_rsa (/home/david/.ssh/id_rsa)
Lifetime set to 3600 seconds
[david@clone1 ~]$ ssh david@clone2
Last login: Thu Dec  2 16:29:00 2021 from clone1.david.local
[david@clone2 ~]$ exit
logout
Connection to clone2 closed.
```

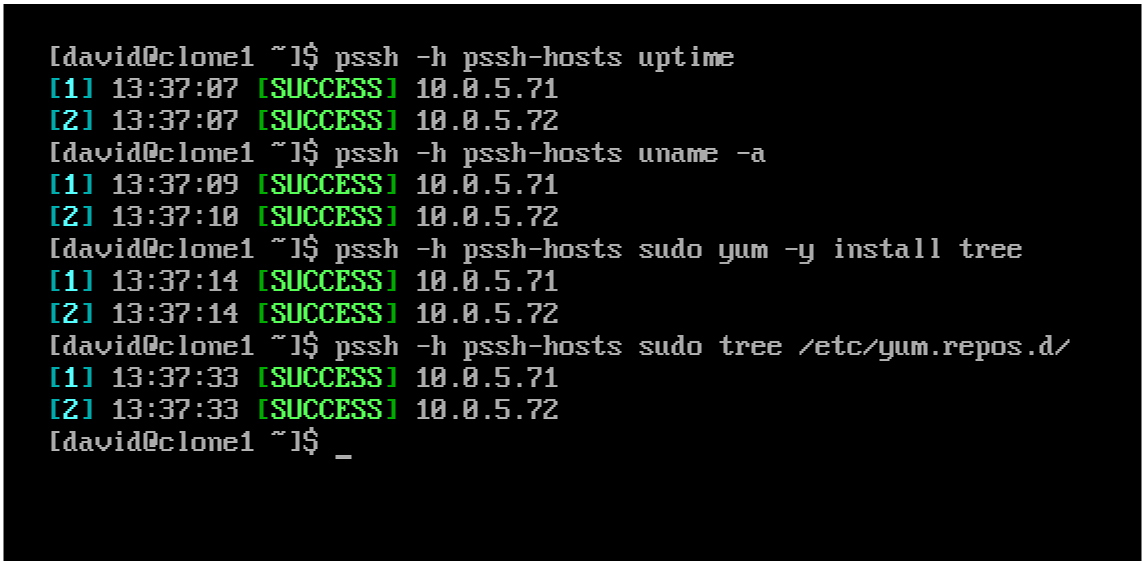
Deliverable 4. Provide a screenshot similar to the one below that shows passwordless access to clone2 or clone3 and elevation to root without retyping a password.



```
[david@clone1 ~]$ ssh clone2
Last login: Thu Dec  2 17:22:25 2021 from clone1.david.local
[david@clone2 ~]$ sudo -i
[root@clone2 ~]#
```

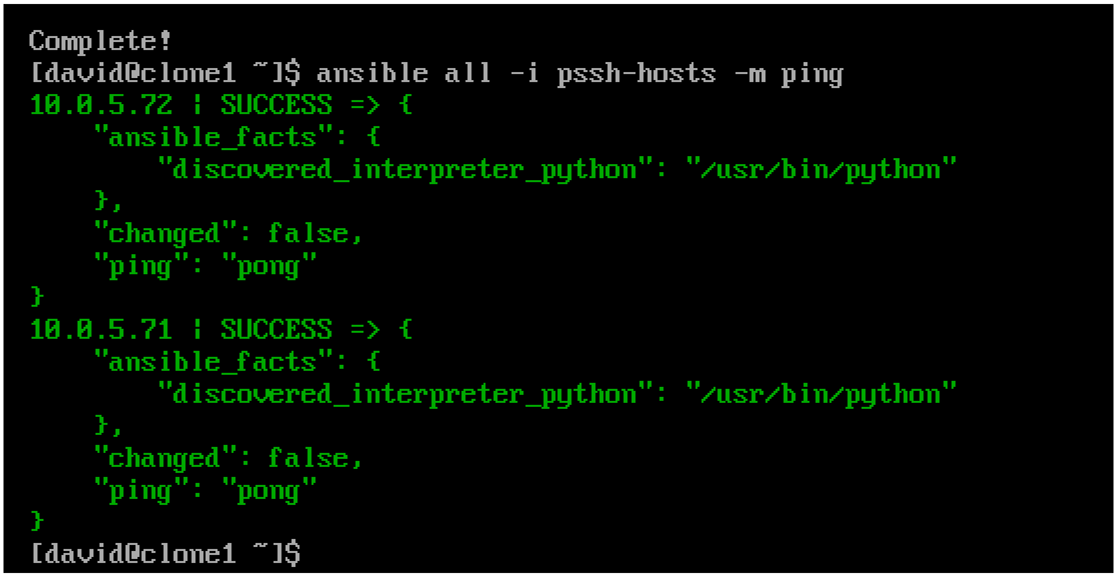
Deliverable 5. Review the man page for pssh and construct a pssh hosts file containing your clone2 and clone3. Then execute the following non-privileged and privileged commands displaying inline standard output & errors as each host completes. Provide screenshots showing the command and [SUCCESS OUTPUT] for all four commands:

- uptime
- uname -a
- sudo yum -y install tree
- tree /etc/yum.repos.d/



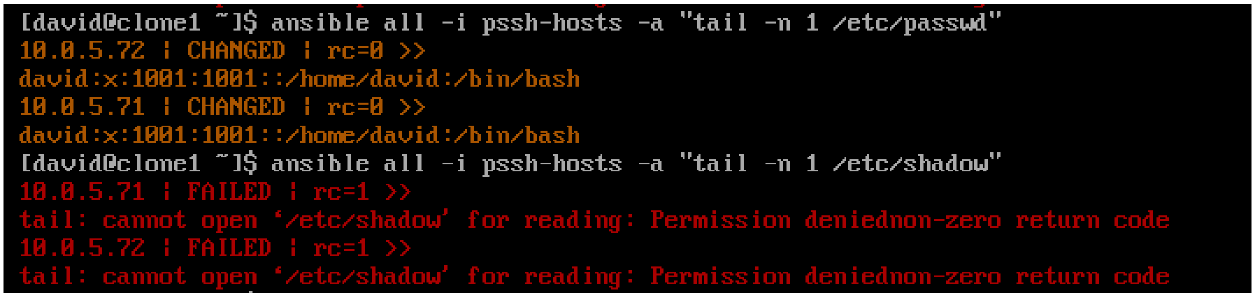
```
[david@clone1 ~]$ pssh -h pssh-hosts uptime
[1] 13:37:07 [SUCCESS] 10.0.5.71
[2] 13:37:07 [SUCCESS] 10.0.5.72
[david@clone1 ~]$ pssh -h pssh-hosts uname -a
[1] 13:37:09 [SUCCESS] 10.0.5.71
[2] 13:37:10 [SUCCESS] 10.0.5.72
[david@clone1 ~]$ pssh -h pssh-hosts sudo yum -y install tree
[1] 13:37:14 [SUCCESS] 10.0.5.71
[2] 13:37:14 [SUCCESS] 10.0.5.72
[david@clone1 ~]$ pssh -h pssh-hosts sudo tree /etc/yum.repos.d/
[1] 13:37:33 [SUCCESS] 10.0.5.71
[2] 13:37:33 [SUCCESS] 10.0.5.72
[david@clone1 ~]$ _
```

Deliverable 6. Install the ansible package using yum on just clone1. Once installed, conduct the following test that walks through all hosts in your hosts file and runs a module called ping against them. Take a screenshot similar to the one below that shows a ping and pong response from clone2 and clone3.

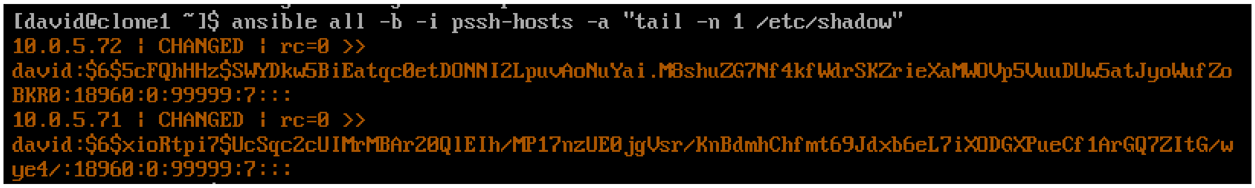


```
Complete!
[david@clone1 ~]$ ansible all -i pssh-hosts -m ping
10.0.5.72 | SUCCESS => {
    "ansible_facts": {
        "discovered_interpreter_python": "/usr/bin/python"
    },
    "changed": false,
    "ping": "pong"
}
10.0.5.71 | SUCCESS => {
    "ansible_facts": {
        "discovered_interpreter_python": "/usr/bin/python"
    },
    "changed": false,
    "ping": "pong"
}
[david@clone1 ~]$
```

Deliverable 7. Provide a screenshot similar to the one below.

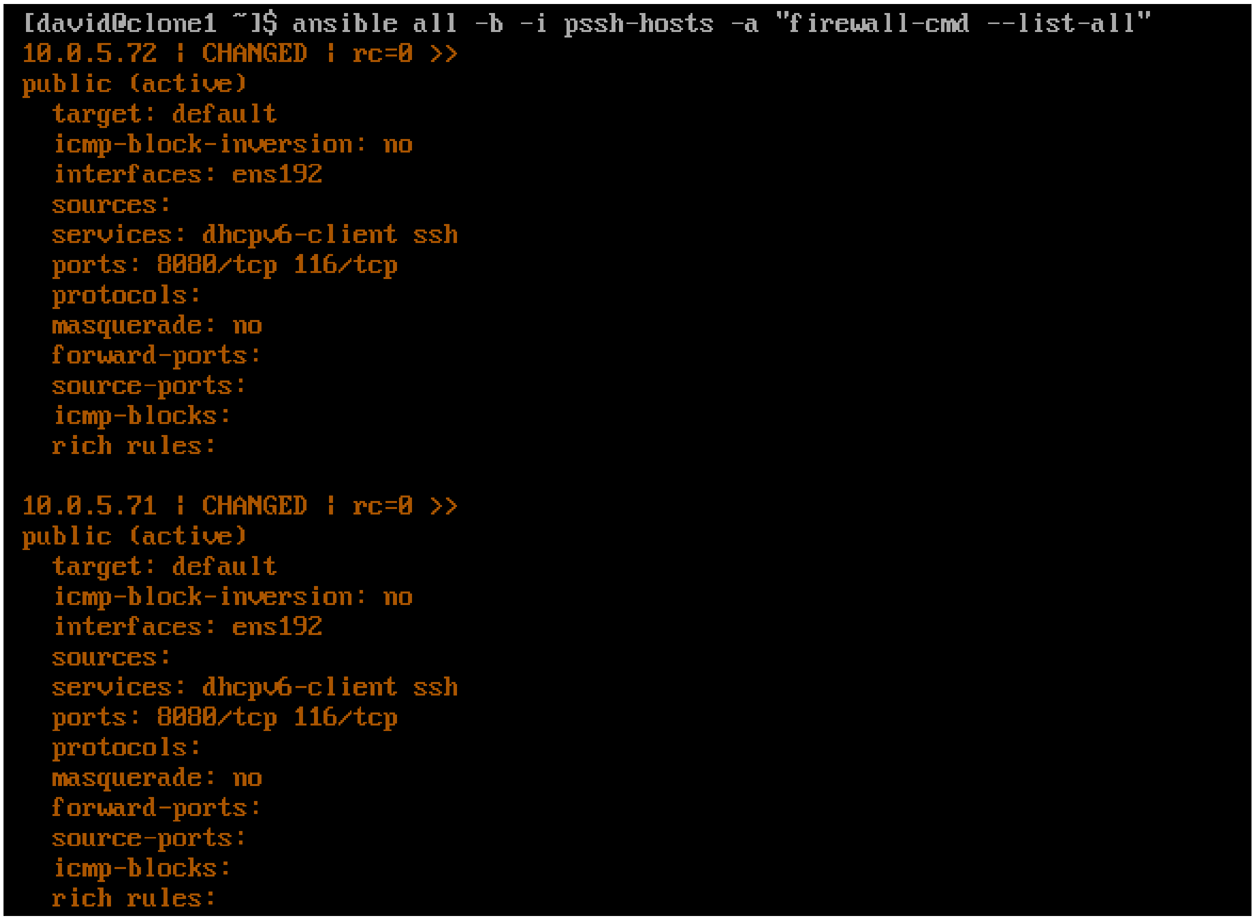


```
[david@clone1 ~]$ ansible all -i pssh-hosts -a "tail -n 1 /etc/passwd"
10.0.5.72 | CHANGED | rc=0 >>
david:x:1001:1001::/home/david:/bin/bash
10.0.5.71 | CHANGED | rc=0 >>
david:x:1001:1001::/home/david:/bin/bash
[david@clone1 ~]$ ansible all -i pssh-hosts -a "tail -n 1 /etc/shadow"
10.0.5.71 | FAILED | rc=1 >>
tail: cannot open '/etc/shadow' for reading: Permission deniednon-zero return code
10.0.5.72 | FAILED | rc=1 >>
tail: cannot open '/etc/shadow' for reading: Permission deniednon-zero return code
```



```
[david@clone1 ~]$ ansible all -b -i pssh-hosts -a "tail -n 1 /etc/shadow"
10.0.5.72 | CHANGED | rc=0 >>
david:$6$5cFQhHHz$SWYDkw5BiEatqc0etDONNI2LpuvAoNuYai.M8shuZG7Nf4kfWdrSKZrieXaMWOVp5VuuDUw5atJyoWufZo
BKR0:18960:0:99999:7:::
10.0.5.71 | CHANGED | rc=0 >>
david:$6$xioRtpi7$UcSqc2cUIMrMBAr20Q1EIh/MP17nzUE0jgVsr/KnBdmhChfmt69Jdxb6eL7iXODGXPueCf1ArGQ7ZItG/w
ye4/:18960:0:99999:7:::
```

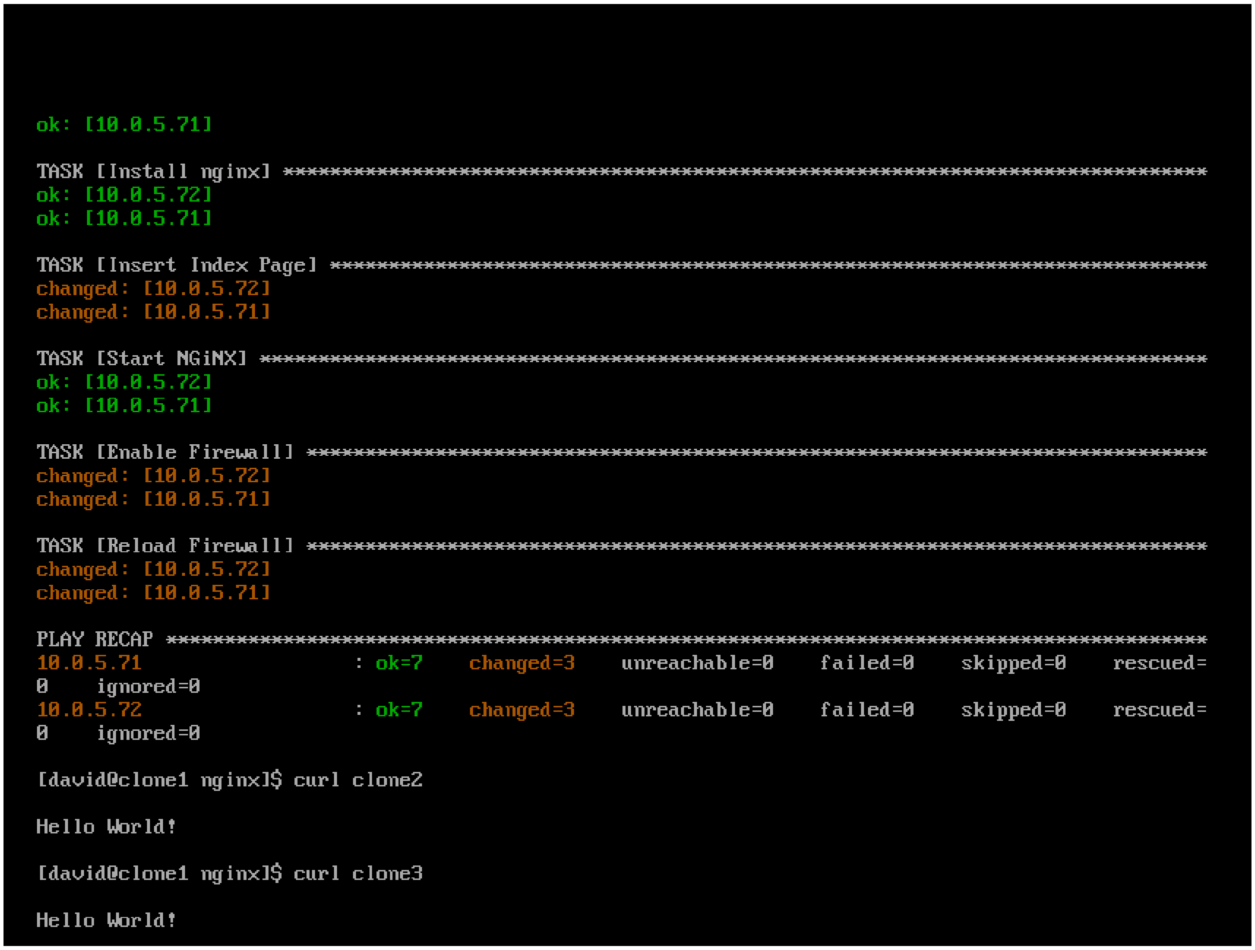
Deliverable 8. Figure out how to add an arbitrary port to the firewall using Ansible. Show the commands used, their success, and then issue another command to list those ports as shown in the following screenshot (8080/tcp in the example).



```
[david@clone1 ~]$ ansible all -b -i pssh-hosts -a "firewall-cmd --list-all"
10.0.5.72 | CHANGED | rc=0 >>
public (active)
  target: default
  icmp-block-inversion: no
  interfaces: ens192
  sources:
  services: dhcpv6-client ssh
  ports: 8080/tcp 116/tcp
  protocols:
  masquerade: no
  forward-ports:
  source-ports:
  icmp-blocks:
  rich rules:

10.0.5.71 | CHANGED | rc=0 >>
public (active)
  target: default
  icmp-block-inversion: no
  interfaces: ens192
  sources:
  services: dhcpv6-client ssh
  ports: 8080/tcp 116/tcp
  protocols:
  masquerade: no
  forward-ports:
  source-ports:
  icmp-blocks:
  rich rules:
```

Deliverable 9. Provide a screenshot that shows a successful playbook run followed by curls to the index page for clone2 and clone3.



```
ok: [10.0.5.71]

TASK [Install nginx] *******************************************************************************
ok: [10.0.5.72]
ok: [10.0.5.71]

TASK [Insert Index Page] ***************************************************************************
changed: [10.0.5.72]
changed: [10.0.5.71]

TASK [Start NGiNX] *********************************************************************************
ok: [10.0.5.72]
ok: [10.0.5.71]

TASK [Enable Firewall] *****************************************************************************
changed: [10.0.5.72]
changed: [10.0.5.71]

TASK [Reload Firewall] *****************************************************************************
changed: [10.0.5.72]
changed: [10.0.5.71]

PLAY RECAP *****************************************************************************************
10.0.5.71                  : ok=7    changed=3    unreachable=0    failed=0    skipped=0    rescued=
0    ignored=0
10.0.5.72                  : ok=7    changed=3    unreachable=0    failed=0    skipped=0    rescued=
0    ignored=0

[david@clone1 nginx]$ curl clone2

Hello World!

[david@clone1 nginx]$ curl clone3

Hello World!
```