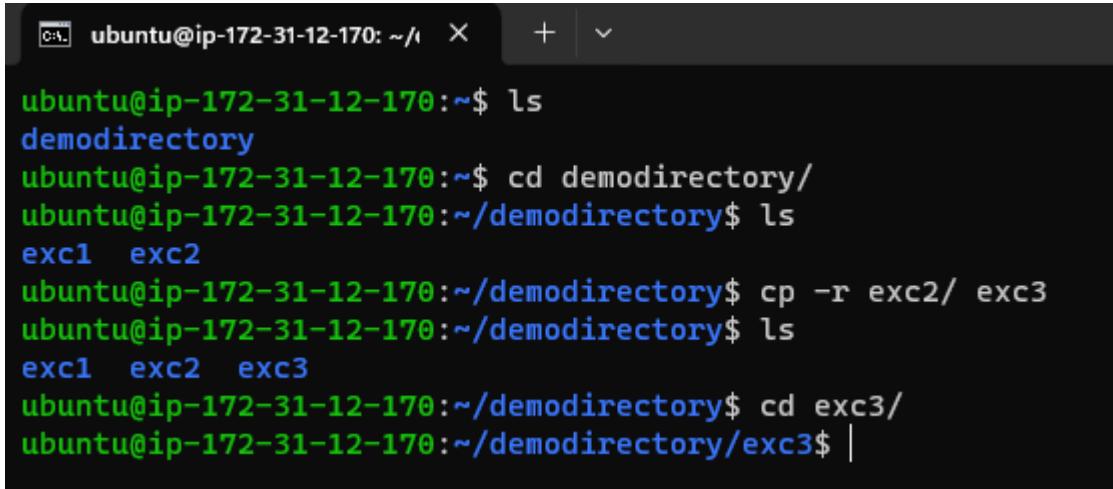


## Ad hoc Commands

1. So, in this lab you are going to learn some ad hoc commands on ansible.
2. First you need to copy the files of previous file into a new file using this command.  
**cp -r exc2/ exc3**
3. Then you need to go the new file.



```
ubuntu@ip-172-31-12-170:~/ | + | ~
ubuntu@ip-172-31-12-170:~$ ls
demodirectory
ubuntu@ip-172-31-12-170:~$ cd demodirectory/
ubuntu@ip-172-31-12-170:~/demodirectory$ ls
exc1  exc2
ubuntu@ip-172-31-12-170:~/demodirectory$ cp -r exc2/ exc3
ubuntu@ip-172-31-12-170:~/demodirectory$ ls
exc1  exc2  exc3
ubuntu@ip-172-31-12-170:~/demodirectory$ cd exc3/
ubuntu@ip-172-31-12-170:~/demodirectory/exc3$ |
```

4. Now do a cat command for inventory.
5. You will see all the content that were in the previous file are now copied into this new file.
6. Now you need to run an ad hoc command for ansible.

```

ubuntu@ip-172-31-12-170:~/demodirectory/exc3$ cat inventory
all:
  hosts:
    demoweb01:
      ansible_host: 172.31.9.178
      ansible_user: ec2-user
      ansible_ssh_private_key_file: demousrkey.pem
    demoweb02:
      ansible_host: 172.31.8.250
      ansible_user: ec2-user
      ansible_ssh_private_key_file: demousrkey.pem
    demodb01:
      ansible_host: 172.31.0.93
      ansible_user: ec2-user
      ansible_ssh_private_key_file: demousrkey.pem

  children:
    webservers:
      hosts:
        demoweb01:
        demoweb02:
    dbservers:
      hosts:
        demodb01:
    demoall:
      children:
        webservers:
        dbservers:
ubuntu@ip-172-31-12-170:~/demodirectory/exc3$ |

```

7. So, when you will run the command, you will see this error and this error says that to use this command you need to be under the root user.

```
ansible demoweb01 -m ansible.builtin.yum -a "name=httpd state=present" -i inventory
```

8. So, you are using the user as ec2-user now that user has a privilege to do sudo, but it is not going to do until and unless you tell.

```

ubuntu@ip-172-31-12-170:~/demodirectory/exc3$ ansible demoweb01 -m ansible.builtin.yum -a "name=httpd state=present" -i inventory
demoweb01 | FAILED! => {
  "ansible_facts": {
    "discovered_interpreter_python": "/usr/bin/python3"
  },
  "changed": false,
  "msg": "This command has to be run under the root user.",
  "results": []
}
ubuntu@ip-172-31-12-170:~/demodirectory/exc3$ |

```

9. So, for that you need to use the same command but in the end you need to write hyphen hyphen become (--become) .

10. Hyphen hyphen become will actually escalate the privilege. It will run the **sudo command sudo yum install** like that or any whatever module you mention there with sudo privilege.

```
ansible demoweb01 -m ansible.builtin.yum -a "name=httpd state=present" -i inventory --become
```

11. So, now if you will run this command in your machine, you will see that it has executed successfully.

```
ubuntu@ip-172-31-12-170:~/demodirectory/exc3$ ansible demoweb01 -m ansible.builtin.yum -a "name=httpd state=present" -i inventory --become
demoweb01 | CHANGED => {
    "ansible_facts": {
        "discovered_interpreter_python": "/usr/bin/python3"
    },
    "changed": true,
    "msg": "",
    "rc": 0,
    "results": [
        "Installed: mod_lua-2.4.57-6.el9.x86_64",
        "Installed: apr-util-openssl-1.6.1-23.el9.x86_64",
        "Installed: httpd-tools-2.4.57-6.el9.x86_64",
        "Installed: apr-1.7.0-12.el9.x86_64",
        "Installed: httpd-2.4.57-6.el9.x86_64",
        "Installed: mailcap-2.1.49-5.el9.noarch",
        "Installed: httpd-core-2.4.57-6.el9.x86_64",
        "Installed: apr-util-1.6.1-23.el9.x86_64",
        "Installed: mod_http2-1.15.19-5.el9.x86_64",
        "Installed: centos-logos-httpd-90.4-1.el9.noarch",
        "Installed: apr-util-bdb-1.6.1-23.el9.x86_64",
        "Installed: httpd-filesystem-2.4.57-6.el9.noarch"
    ]
}
ubuntu@ip-172-31-12-170:~/demodirectory/exc3$ |
```

12. Now run the same command but this time for a group let say for webserver group.

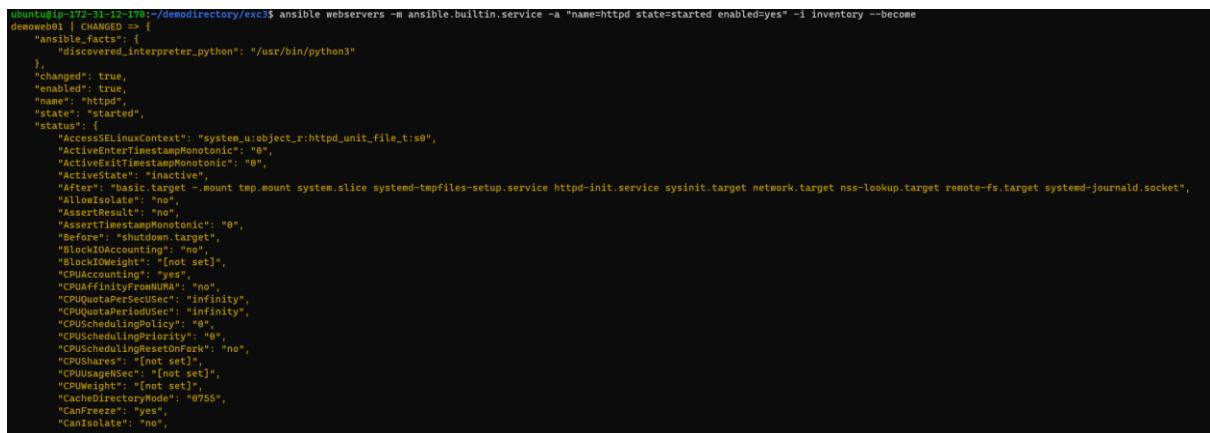
13. You will see that it runs successfully.

```
ansible webservers -m ansible.builtin.yum -a "name=httpd state=present" -i inventory --become
```

```
ubuntu@ip-172-31-12-170:~/demodirectory/exc3$ ansible webservers -m ansible.builtin.yum -a "name=httpd state=present" -i inventory --become
demoweb01 | SUCCESS => {
    "ansible_facts": {
        "discovered_interpreter_python": "/usr/bin/python3"
    },
    "changed": false,
    "msg": "Nothing to do",
    "rc": 0,
    "results": []
}
demoweb02 | CHANGED => {
    "ansible_facts": {
        "discovered_interpreter_python": "/usr/bin/python3"
    },
    "changed": true,
    "msg": "",
    "rc": 0,
    "results": [
        "Installed: mod_lua-2.4.57-6.el9.x86_64",
        "Installed: apr-util-openssl-1.6.1-23.el9.x86_64",
        "Installed: httpd-tools-2.4.57-6.el9.x86_64",
        "Installed: apr-1.7.0-12.el9.x86_64",
        "Installed: httpd-2.4.57-6.el9.x86_64",
        "Installed: mailcap-2.1.49-5.el9.noarch",
        "Installed: httpd-core-2.4.57-6.el9.x86_64",
        "Installed: apr-util-1.6.1-23.el9.x86_64",
        "Installed: mod_http2-1.15.19-5.el9.x86_64",
        "Installed: centos-logos-httpd-90.4-1.el9.noarch",
        "Installed: apr-util-bdb-1.6.1-23.el9.x86_64",
        "Installed: httpd-filesystem-2.4.57-6.el9.noarch"
    ]
}
ubuntu@ip-172-31-12-170:~/demodirectory/exc3$ |
```

14. Run a different command and you will get a huge output.

```
ansible webservers -m ansible.builtin.service -a "name=httpd state=started enabled=yes" -i inventory --become
```



```
ubuntu@ip-172-31-12-170:~/demodirectory/exc3$ ansible webservers -m ansible.builtin.service -a "name=httpd state=started enabled=yes" -i inventory --become
[debase01 | CHANGED => {
    "ansible_facts": {
        "discovered_interpreter_python": "/usr/bin/python3"
    },
    "changed": true,
    "enabled": true,
    "name": "httpd",
    "state": "started",
    "status": {
        "AccessLinuxContext": "system_u:object_r:httpd_unit_file_t:s0",
        "ActiveEnterTimestampMonotonic": "0",
        "ActiveExitTimestampMonotonic": "0",
        "ActiveState": "inactive",
        "After": "*basic.target -mount tmp.mount system.slice systemd-tmpfiles-setup.service httpd-init.service sysinit.target network.target nss-lookup.target remote-fs.target systemd-journald.socket",
        "AllowIsolate": "no",
        "AssertResult": "no",
        "AssertTimestampMonotonic": "0",
        "Before": "shutdown.target",
        "BlockIOAccounting": "no",
        "BlockIOWeight": "[not set]",
        "cpusAccounting": "no",
        "CPUAffinityFromNUMA": "no",
        "CPUQuotaPerSecusec": "infinity",
        "CPUQuotaPerPeriodSec": "infinity",
        "CPSchedulingPolicy": "0",
        "CPSchedulingPriority": "0",
        "CPSchedulingResetOnFork": "no",
        "CPUShares": "[not set]",
        "cpusShares": "[not set]",
        "CPUWeight": "[not set]",
        "CacheDirectoryMode": "0755",
        "CanFreeze": "yes",
        "CanIsolate": "no"
    }
}
```

15. This time you are going to copy a file. For that first you need to create a directory.

**vim index.html**

16. In this html file write something then just save and quit.



```
ubuntu@ip-172-31-12-170:~/demodirectory/exc3$ vim index.html
ubuntu@ip-172-31-12-170:~/demodirectory/exc3$ ls
demousrkey.pem  index.html  inventory
ubuntu@ip-172-31-12-170:~/demodirectory/exc3$ |
```

17. Now you need to run this command.

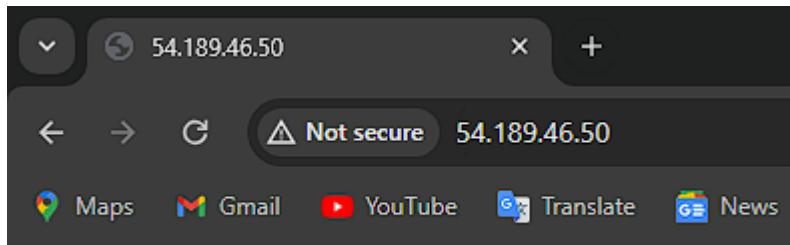
```
ansible webservers -m ansible.builtin.copy -a "src=index.html dest=/var/www/html/index.html" -i inventory --become
```

```

ubuntu@ip-172-31-12-170:~/ ~ | + | ~
ubuntu@ip-172-31-12-170:~/demodirectory/exc3$ vim index.html
ubuntu@ip-172-31-12-170:~/demodirectory/exc3$ ls
demourkey.pem index.html inventory
ubuntu@ip-172-31-12-170:~/demodirectory/exc3$ ansible webservers -m ansible.builtin.copy -a "src=index.html dest=/var/www/html/index.html" -i inventory --become
demoweb01 | CHANGED => {
    "ansible_facts": {
        "discovered_interpreter_python": "/usr/bin/python3"
    },
    "changed": true,
    "checksum": "d0cc5b8555124073911a3f87990f0e3748e0ef96",
    "dest": "/var/www/html/index.html",
    "gid": 0,
    "group": "root",
    "md5sum": "0f76cef9510459f72bb64788ddc33898",
    "mode": "0644",
    "owner": "root",
    "secontext": "system_u:object_r:httpd_sys_content_t:s0",
    "size": 47,
    "src": "/home/ec2-user/.ansible/tmp/ansible-tmp-1706886168.1083522-1459-17300551954145/source",
    "state": "file",
    "uid": 0
}
demoweb02 | CHANGED => {
    "ansible_facts": {
        "discovered_interpreter_python": "/usr/bin/python3"
    },
    "changed": true,
    "checksum": "d0cc5b8555124073911a3f87990f0e3748e0ef96",
    "dest": "/var/www/html/index.html",
    "gid": 0,
    "group": "root",
    "md5sum": "0f76cef9510459f72bb64788ddc33898",
    "mode": "0644",
    "owner": "root",
    "secontext": "system_u:object_r:httpd_sys_content_t:s0",
    "size": 47,
    "src": "/home/ec2-user/.ansible/tmp/ansible-tmp-1706886168.133736-1460-60433730567515/source",
    "state": "file",
    "uid": 0
}
ubuntu@ip-172-31-12-170:~/demodirectory/exc3$ |

```

18. After successfully running the command now you need to go back to AWS Console and open EC2.
19. Navigate to your instances with CentOS and add port 80 with HTTP on inbound rules of your security group.
20. Then copy public IP address of any of the instance and paste it in a new tab you will see the text that you've written in the index.html file which you created.



This is a demo for ad hoc commands of Ansible.