

Name: Repani, Justin Jello J.	Date Performed: September 25, 2023
Course/Section: CPE31S6	Date Submitted: September 25, 2023
Instructor: Dr. Jonathan V. Taylar	Semester and SY: 1st Sem: SY 2023-2024
PRELIM EXAMINATION	

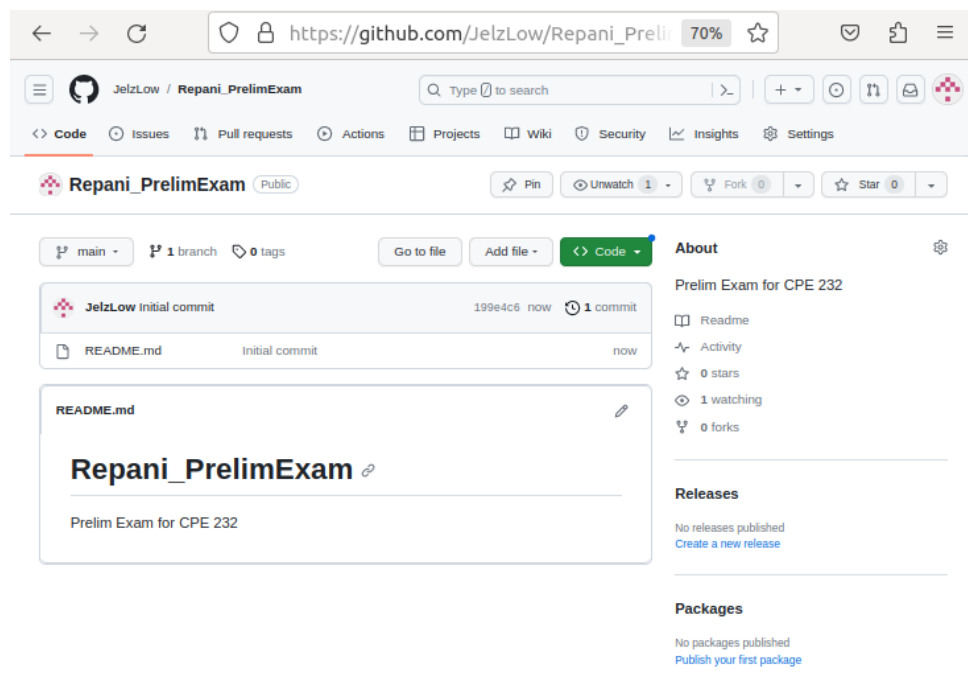
Tools Needed

1. Control Node (CN) - 1
2. Manage Node (MN) - 1 Ubuntu
3. Manage Node (MN) - 1 CentOS

Procedure

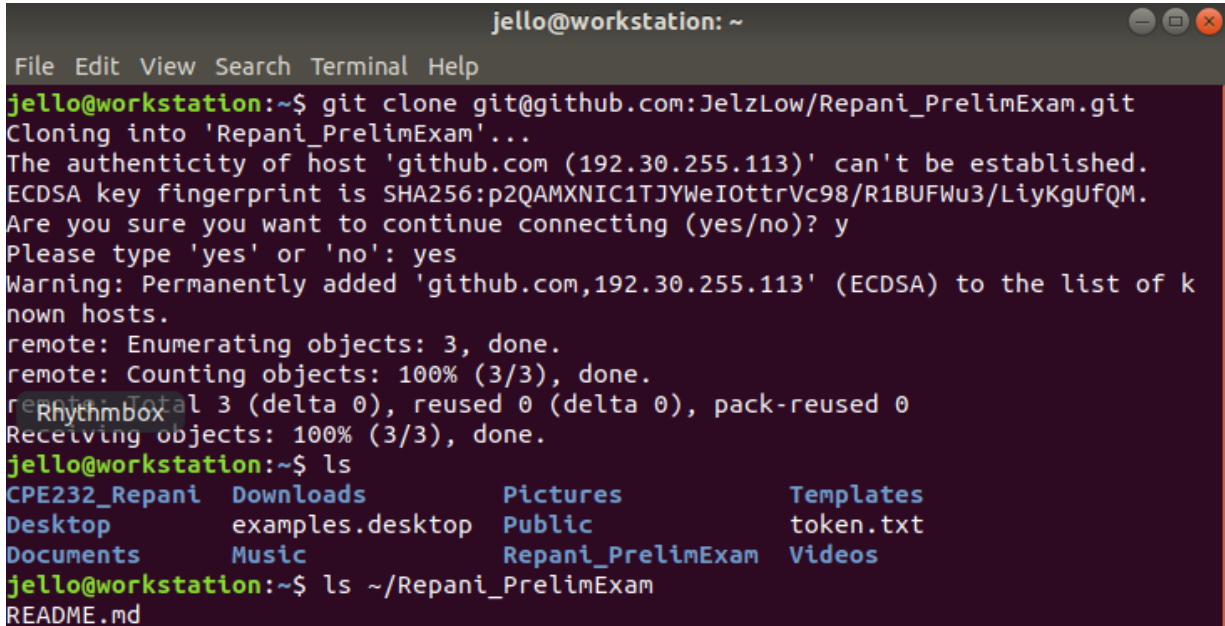
1. Note: You are required to create a document report of the steps you will do for this exam. All screenshots should be labeled and explained properly.

2. Create a repository in your GitHub account and label it as Surname_PrelimExam



The new repository is created with the default settings at public and with a readme file added upon creation

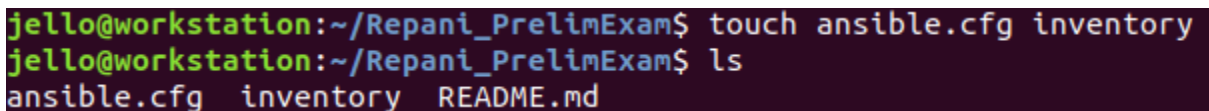
3. Clone your new repository in your CN.

A terminal window titled 'jello@workstation: ~' with a menu bar (File, Edit, View, Search, Terminal, Help). The terminal shows the execution of 'git clone git@github.com:JelzLow/Repani_PrelimExam.git'. It displays the cloning progress, a warning about the host's authenticity, and a confirmation to continue. After cloning, the user runs 'ls' showing a directory listing including 'CPE232_Repani', 'Downloads', 'Pictures', 'Templates', 'Desktop', 'examples.desktop', 'Public', 'token.txt', 'Documents', 'Music', 'Repani_PrelimExam', and 'Videos'. Finally, the user runs 'ls ~/Repani_PrelimExam' showing 'README.md'.

```
jello@workstation: ~$ git clone git@github.com:JelzLow/Repani_PrelimExam.git
Cloning into 'Repani_PrelimExam'...
The authenticity of host 'github.com (192.30.255.113)' can't be established.
ECDSA key fingerprint is SHA256:p2QAMXNIC1TJYWeIOttrVc98/R1BUFWu3/LiyKgUfQM.
Are you sure you want to continue connecting (yes/no)? y
Please type 'yes' or 'no': yes
Warning: Permanently added 'github.com,192.30.255.113' (ECDSA) to the list of k
nown hosts.
remote: Enumerating objects: 3, done.
remote: Counting objects: 100% (3/3), done.
remote: Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
Receiving objects: 100% (3/3), done.
jello@workstation:~$ ls
CPE232_Repani  Downloads      Pictures        Templates
Desktop        examples.desktop Public          token.txt
Documents      Music          Repani_PrelimExam Videos
jello@workstation:~$ ls ~/Repani_PrelimExam
README.md
```

The command git clone is used and the link copied came from the git website with the SSH function

4. In your CN, create an inventory file and ansible.cfg files.

A terminal window showing the execution of 'touch ansible.cfg inventory' in the directory '~/Repani_PrelimExam'. This is followed by a 'ls' command which lists 'ansible.cfg', 'inventory', and 'README.md'.

```
jello@workstation:~/Repani_PrelimExam$ touch ansible.cfg inventory
jello@workstation:~/Repani_PrelimExam$ ls
ansible.cfg  inventory  README.md
```

The touch command is used in order to create the ansible.cfg and inventory files

```
jello@workstation: ~/Repani_PrelimExam
File Edit View Search Terminal Help
GNU nano 2.9.3 ansible.cfg Modified

[defaults]

inventory = inventory
host_key_checking = False

deprecation_warnings = False

remote_user = jello
private_key_file = ~/.ssh/

^G Get Help      ^O Write Out    ^W Where Is     ^K Cut Text     ^J Justify
^X Exit          ^R Read File    ^\ Replace      ^U Uncut Text   ^T To Spell
```

```
jello@workstation: ~/Repani_PrelimExam
File Edit View Search Terminal Help
GNU nano 2.9.3 inventory Modified
192.168.56.102 ansible_python_interpreter=/usr/bin/python3
192.168.56.102 apache_package=apache2 php_package=libapache2-mod-php
192.168.56.104 ansible_python_interpreter=/usr/bin/python3
192.168.56.104 apache_package=httpd php_package=php
^G Get Help ^O Write Out ^W Where Is ^K Cut Text ^J Justify
^X Exit ^R Read File ^\ Replace ^U Uncut Text ^T To Spell
```

The same ansible.cfg and inventory file content is copied and pasted from the previous activities.

```
jello@workstation:~/Repani_PrelimExam$ ansible all -m ping
192.168.56.104 | SUCCESS => {
  "changed": false,
  "ping": "pong"
}
192.168.56.102 | SUCCESS => {
  "changed": false,
  "ping": "pong"
}
```

The connection is tested with the use of the ansible all -m ping command

5. Create an Ansible playbook that does the following with an input of a config.yaml file for both Manage Nodes

- Installs the latest python3 and pip3

```
jello@workstation: ~/Repani_PrelimExam
File Edit View Search Terminal Help
GNU nano 2.9.3 test.yml
---
- hosts: all
  become: true
  tasks:

    - name: Install python3 and pip3
      package:
        name:
          - python3
          - python3-pip
        state: latest
        update_cache: yes
```

```
jello@workstation:~/Repani_PrelimExam$ ansible-playbook --ask-become-pass test.
.yml
BECOME password:

PLAY [all] *****
*

TASK [Gathering Facts] *****
*
ok: [192.168.56.102]
ok: [192.168.56.104]

TASK [Install python3 and pip3] *****
*
ok: [192.168.56.104]
changed: [192.168.56.102]

PLAY RECAP *****
*
192.168.56.102      : ok=2    changed=1    unreachable=0    failed=0
skipped=0    rescued=0    ignored=0
192.168.56.104      : ok=2    changed=0    unreachable=0    failed=0
skipped=0    rescued=0    ignored=0
```

I made use of the package: command since the OS on server1 is Ubuntu, and server2 has CentOS. This enables the playbook to install the packages without encountering any conflict.

- use pip3 as default pip

```
jello@workstation: ~/Repani_PrelimExam
File Edit View Search Terminal Help
GNU nano 2.9.3 test.yml

--
- hosts: all
  become: true
  tasks:

    - name: use pip3 as default pip
      lineinfile:
        path: /etc/alternatives/pip
        line: /user/bin/pip3
        create: yes

    - name: use python3 as default python
      lineinfile:
        path: /etc/alternatives/python
        line: /user/bin/python3
        create: yes

[ Read 20 lines ]
^G Get Help      ^O Write Out    ^W Where Is    ^K Cut Text    ^J Justify
^X Exit          ^R Read File    ^\ Replace     ^U Uncut Text  ^T To Spell

jello@workstation:~/Repani_PrelimExam$ ansible-playbook --ask-become-pass test.
yml
BECOME password:

PLAY [all] *****
*

TASK [Gathering Facts] *****
*
ok: [192.168.56.102]
ok: [192.168.56.104]

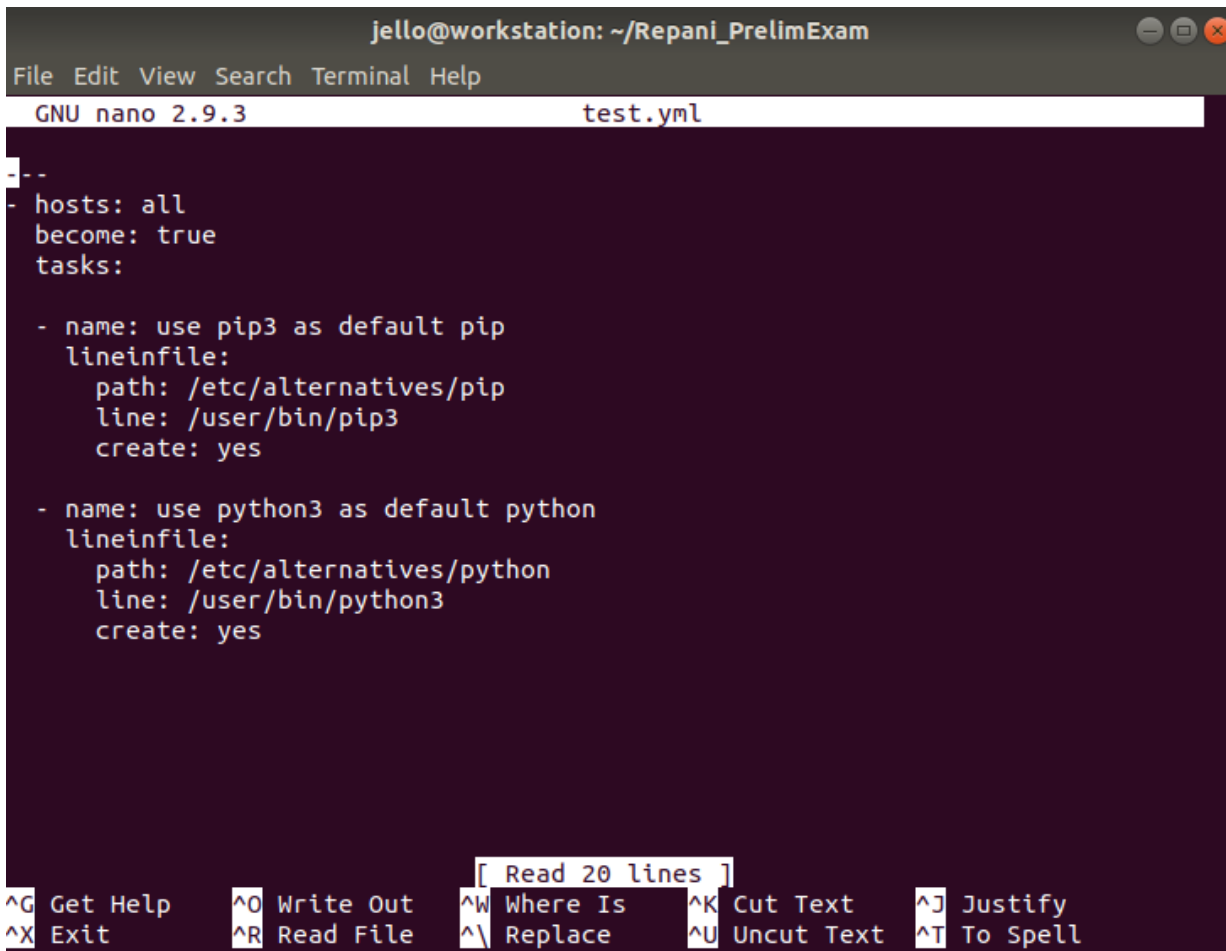
TASK [use pip3 as default pip] *****
*
changed: [192.168.56.102]
changed: [192.168.56.104]

TASK [use python3 as default python] *****
*
changed: [192.168.56.102]
changed: [192.168.56.104]
```

```
PLAY RECAP *****
*
192.168.56.102      : ok=3    changed=2    unreachable=0    failed=0
skipped=0    rescued=0    ignored=0
192.168.56.104      : ok=3    changed=2    unreachable=0    failed=0
skipped=0    rescued=0    ignored=0
```

In this part, the command `lineinfile` is used in order to point the line to the path which will set `pip3` as the default `pip`

- use `python3` as default `python`



```
jello@workstation: ~/Repani_PrelimExam
File Edit View Search Terminal Help
GNU nano 2.9.3 test.yml
--
- hosts: all
  become: true
  tasks:

  - name: use pip3 as default pip
    lineinfile:
      path: /etc/alternatives/pip
      line: /user/bin/pip3
      create: yes

  - name: use python3 as default python
    lineinfile:
      path: /etc/alternatives/python
      line: /user/bin/python3
      create: yes

[ Read 20 lines ]
^G Get Help      ^O Write Out    ^W Where Is     ^K Cut Text     ^J Justify
^X Exit          ^R Read File    ^\ Replace      ^U Uncut Text   ^T To Spell
```

```
jello@workstation:~/Repani_PrelimExam$ ansible-playbook --ask-become-pass test.
yml
BECOME password:

PLAY [all] *****
*

TASK [Gathering Facts] *****
*
ok: [192.168.56.102]
ok: [192.168.56.104]

TASK [use pip3 as default pip] *****
*
changed: [192.168.56.102]
changed: [192.168.56.104]

TASK [use python3 as default python] *****
*
changed: [192.168.56.102]
changed: [192.168.56.104]

PLAY RECAP *****
*
192.168.56.102      : ok=3    changed=2    unreachable=0    failed=0
skipped=0    rescued=0    ignored=0
192.168.56.104      : ok=3    changed=2    unreachable=0    failed=0
skipped=0    rescued=0    ignored=0
```

In this part, the command lineinfile is used in order to point the line to the path which will set python3 as the default python

- Install Java open-jdk


```
jello@workstation: ~/Repani_PrelimExam
File Edit View Search Terminal Help
GNU nano 2.9.3 test.yml

---
- hosts: all
  become: true
  tasks:

    - name: Install Java open-jdk
      apt:
        name: openjdk-17-jdk
        state: latest
        update_cache: yes
        when: ansible_distribution == 'Ubuntu'

    - name: Install Java open-jdk
      dnf:
        name: java-11-openjdk
        state: latest
        update_cache: yes
        when: ansible_distribution == 'CentOS'

[ Read 21 lines ]
^G Get Help      ^O Write Out    ^W Where Is     ^K Cut Text     ^J Justify
^X Exit          ^R Read File    ^\ Replace      ^U Uncut Text   ^T To Spell
```

```
jello@workstation:~/Repani_PrelimExam$ ansible-playbook --ask-become-pass test.
yaml
BECOME password:

PLAY [all] *****
*

TASK [Gathering Facts] *****
*
ok: [192.168.56.102]
ok: [192.168.56.104]

TASK [Install Java open-jdk] *****
*
skipping: [192.168.56.104]
ok: [192.168.56.102]

TASK [Install Java open-jdk] *****
*
skipping: [192.168.56.102]
changed: [192.168.56.104]

PLAY RECAP *****
*

PLAY RECAP *****
*
192.168.56.102      : ok=2    changed=0    unreachable=0    failed=0
skipped=1    rescued=0    ignored=0
192.168.56.104      : ok=2    changed=1    unreachable=0    failed=0
skipped=1    rescued=0    ignored=0
```

The command apt and dnf are used separately because the Ubuntu and CentOS have different versions and syntax for the Java open-jdk packages

- Create Motd containing the text defined by a variable defined in config.yaml file and if there is no variable input the default motd is "Ansible Managed node by (your user name)"

```
jello@workstation: ~/Repani_PrelimExam
File Edit View Search Terminal Help
GNU nano 2.9.3 test.yml

---
- hosts: all
  become: true

  vars:
    motd:
      - Ansible Managed Node by jello

  tasks:
    - name: motd
      ansible.builtin.debug:
        msg:
          - "{{ motd }}"

[ Read 18 lines ]
^G Get Help      ^O Write Out    ^W Where Is     ^K Cut Text     ^J Justify
^X Exit          ^R Read File    ^\ Replace      ^U Uncut Text  ^T To Spell
```

```
jello@workstation:~/Repani_PrelimExam$ ansible-playbook --ask-become-pass test.
yaml
BECOME password:

PLAY [all] *****
*

TASK [Gathering Facts] *****
*
ok: [192.168.56.102]
ok: [192.168.56.104]

TASK [motd] *****
*
ok: [192.168.56.102] => {
  "msg": [
    [
      "Ansible Managed Node by jello"
    ]
  ]
}
ok: [192.168.56.104] => {
  "msg": [
    [
      "Ansible Managed Node by jello"
    ]
  ]
}

PLAY RECAP *****
*
192.168.56.102      : ok=2    changed=0    unreachable=0    failed=0
skipped=0    rescued=0    ignored=0
192.168.56.104      : ok=2    changed=0    unreachable=0    failed=0
skipped=0    rescued=0    ignored=0
```

In this part, the `ansible.builtin.debug msg` is used and a variable of `motd` is created so that it takes the content of this variable and displays it as a message.

- Create a user with a variable defined in `config.yaml`

```
jello@workstation: ~/Repani_PrelimExam
File Edit View Search Terminal Help
GNU nano 2.9.3 test.yml

---
- hosts: all
  become: true

  vars:
    new_user: NewUserTest

  tasks:

    - name: user
      user:
        name: "{{ new_user }}"
        state: present
```

```
jello@workstation:~/Repani_PrelimExam$ ansible-playbook --ask-become-pass test.
.yml
BECOME password:

PLAY [all] *****
*

TASK [Gathering Facts] *****
*
ok: [192.168.56.102]
ok: [192.168.56.104]

TASK [user] *****
*
changed: [192.168.56.102]
changed: [192.168.56.104]

PLAY RECAP *****
*
192.168.56.102      : ok=2    changed=1    unreachable=0    failed=0
skipped=0    rescued=0    ignored=0
192.168.56.104      : ok=2    changed=1    unreachable=0    failed=0
skipped=0    rescued=0    ignored=0
```

In the new user, the user command was used and a variable of new_user is created so that it takes the content of this variable and creates it as a new user.

config.yml content

```
jello@workstation: ~/Repani_PrelimExam
File Edit View Search Terminal Help
GNU nano 2.9.3 config.yml Modified

---
- hosts: all
  become: true

  vars:
    motd:
      - Ansible Managed Node by jello
    new_user: NewUserTest

  tasks:

    - name: Install python3 and pip3
      package:
        name:
          - python3
          - python3-pip
        state: latest
        update_cache: yes

    - name: use pip3 as default pip
      lineinfile:
        path: /etc/alternatives/pip
        line: /user/bin/pip3

    - name: use python3 as default python
      lineinfile:
        path: /etc/alternatives/python
        line: /user/bin/python3
        create: yes

    - name: Install Java open-jdk
      apt:
        name: openjdk-17-jdk
        state: latest
        update_cache: yes
        when: ansible_distribution == 'Ubuntu'

    - name: Install Java open-jdk
      dnf:
        name: java-11-openjdk
        state: latest
        update_cache: yes
        when: ansible_distribution == 'CentOS'
```

```
- name: Create MOTD
  ansible.builtin.debug:
    msg:
      - "{{ motd }}"

- name: Create user with variable
  user:
    name: "{{ new_user }}"
    state: present
```

Running

```
jello@workstation:~/Repani_PrelimExam$ ansible-playbook --ask-become-pass config.yml
BECOME password:

PLAY [all] *****
*

TASK [Gathering Facts] *****
*
ok: [192.168.56.102]
ok: [192.168.56.104]

TASK [Install python3 and pip3] *****
*
ok: [192.168.56.104]
ok: [192.168.56.102]

TASK [use pip3 as default pip] *****
*
ok: [192.168.56.102]
ok: [192.168.56.104]

TASK [use python3 as default python] *****
*
ok: [192.168.56.102]
ok: [192.168.56.104]
```

```

TASK [Install Java open-jdk] *****
*
skipping: [192.168.56.104]
ok: [192.168.56.102]

TASK [Install Java open-jdk] *****
*
skipping: [192.168.56.102]
ok: [192.168.56.104]

TASK [Create MOTD] *****
*
ok: [192.168.56.102] => {
  "msg": [
    [
      "Ansible Managed Node by jello"
    ]
  ]
}
ok: [192.168.56.104] => {
  "msg": [
    [
      "Ansible Managed Node by jello"
    ]
  ]
}

TASK [Create user with variable] *****
*
ok: [192.168.56.102]
ok: [192.168.56.104]

PLAY RECAP *****
*
192.168.56.102      : ok=7    changed=0    unreachable=0    failed=0
skipped=1    rescued=0    ignored=0
192.168.56.104      : ok=7    changed=0    unreachable=0    failed=0
skipped=1    rescued=0    ignored=0

```

Proof


```
Server1_Repani [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Activities Terminal Mon 19:04
jello@server1: ~

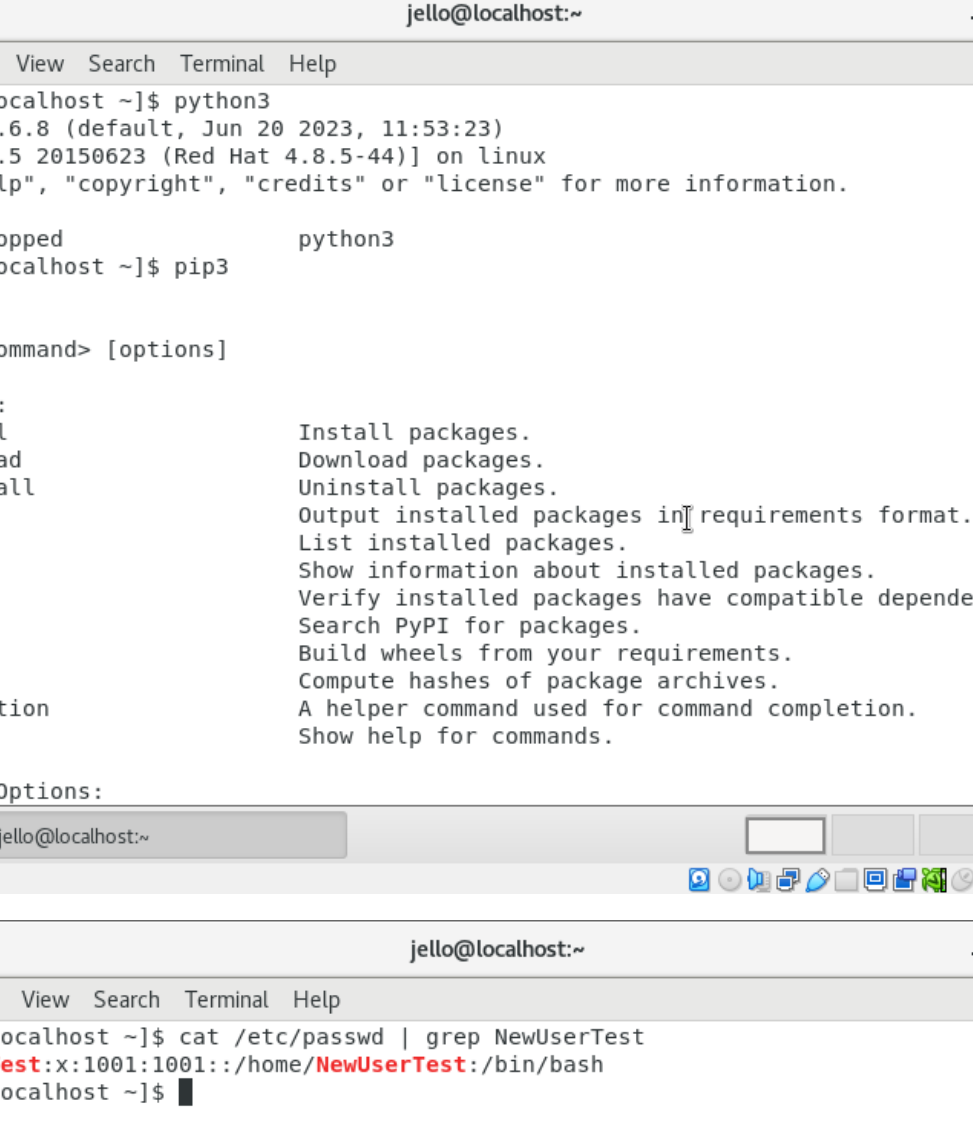
File Edit View Search Terminal Help
jello@server1:~$ python3
Python 3.6.9 (default, Mar 10 2023, 16:46:00)
[GCC 8.4.0] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>>
[3]+ Stopped python3
jello@server1:~$ pip3

Usage:
  pip <command> [options]

Commands:
  install           Install packages.
  download          Download packages.
  uninstall         Uninstall packages.
  freeze            Output installed packages in requirements format.
  list              List installed packages.
  show              Show information about installed packages.
  check             Verify installed packages have compatible dependencies.
  search            Search PyPI for packages.
  wheel             Build wheels from your requirements.
  hash              Compute hashes of package archives.
  completion        A helper command used for command completion.
  help              Show help for commands.

General Options:
  -h, --help        Show help.
  --isolated         Run pip in an isolated mode, ignoring
```

```
jello@server1:~$ cat /etc/passwd | grep NewUserTest
NewUserTest:x:1001:1001::/home/NewUserTest:/bin/sh
```



The screenshot shows a terminal window titled "CentOS_Repani [Running] - Oracle VM VirtualBox". The terminal output is as follows:

```
jello@localhost:~
File Edit View Search Terminal Help
[jello@localhost ~]$ python3
Python 3.6.8 (default, Jun 20 2023, 11:53:23)
[GCC 4.8.5 20150623 (Red Hat 4.8.5-44)] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>>
[4]+  Stopped                  python3
[jello@localhost ~]$ pip3

Usage:
  pip <command> [options]

Commands:
  install           Install packages.
  download          Download packages.
  uninstall         Uninstall packages.
  freeze            Output installed packages in requirements format.
  list              List installed packages.
  show              Show information about installed packages.
  check             Verify installed packages have compatible dependencies.
  search            Search PyPI for packages.
  wheel             Build wheels from your requirements.
  hash             Compute hashes of package archives.
  completion        A helper command used for command completion.
  help              Show help for commands.

General Options:
  -h, --help            Show this help message and exit.
  -q, --quiet            Quiet operation.
  -v, --verbose          Verbose operation.
  -y, --yes              Assume Yes for all prompts.
  -n, --no               Assume No for all prompts.
  -U, --upgrade           Upgrade pip.
  --user                 Install to user site.
  --target <dir>         Install to target directory.
  --root <dir>           Install to root directory.
  --prefix <dir>         Install to prefix directory.
  --exec-prefix <dir>    Install to exec-prefix directory.
  --install-dir <dir>    Install to install directory.
  --log <file>           Write logs to log file.
  --proxy <url>          Proxy to use for all network operations.
  --trusted-host <url>  Mark this host as trusted, even if it does not have a
                        valid certificate.
  --cert <file>          Local path to trusted SSL/certificate file.
  --key <file>            Local path to private key file.
  --client-cert <file>   Local path to client certificate file.
  --use-oauth-tokens      Use OAuth tokens to authenticate to PyPI.
  --no-index            Do not connect to the index.
  --index-url <url>     URL to the Python Package Index (PyPI) or other
                        remote package repository.
  --extra-index-url <url>
                        Extra URL to search for packages.
  --find-links <url>    Local path or URL to a directory of package
                        distributions.
  --no-cache-dir         Do not use a cache directory.
  --cache-dir <dir>     Cache directory.
  --no-color             Disable colorized output.
  --color <auto>        Enable colorized output.
  --progress-bar <on>   Enable progress bars.
  --default-timeout <sec>
                        Default timeout for network operations.
  --timeout <sec>       Set the socket timeout for network operations.
  --retry <count>        Maximum number of retries for network operations.
  --verbose              Verbose output.
  --quiet               Quiet output.
  --no-input            Do not prompt for input.
  --dry-run             Do not actually install anything.
  --help                Show this help message and exit.
  --help-commands        Show this help message and exit.
  --help-globals         Show this help message and exit.
```

```
jello@workstation: ~/Repani_PrelimExam
File Edit View Search Terminal Help
jello@workstation:~/Repani_PrelimExam$ git add *
jello@workstation:~/Repani_PrelimExam$ git commit -m "Prelim Exam"
[main 03dbe64] Prelim Exam
 4 files changed, 87 insertions(+)
 create mode 100644 ansible.cfg
 create mode 100644 config.yml
 create mode 100644 inventory
 create mode 100644 test.yml
jello@workstation:~/Repani_PrelimExam$ git push origin
Counting objects: 6, done.
Delta compression using up to 2 threads.
Compressing objects: 100% (6/6), done.
Writing objects: 100% (6/6), 1.09 KiB | 1.09 MiB/s, done.
Total 6 (delta 0), reused 0 (delta 0)
To github.com:JelzLow/Repani_PrelimExam.git
 199e4c6..03dbe64  main -> main
jello@workstation:~/Repani_PrelimExam$
```

← → ↻ https://github.com/JelzLow/Repani_PrelimExam 70% ☆

JelzLow / Repani_PrelimExam

Code Issues Pull requests Actions Projects Wiki Security Insights Settings

Repani_PrelimExam Public Pin Unwatch 1 Fork 0 Star 0

main 1 branch 0 tags Code

JelzLow Prelim Exam 03dbe64 1 minute ago 2 commits		
README.md	Initial commit	2 hours ago
ansible.cfg	Prelim Exam	1 minute ago
config.yml	Prelim Exam	1 minute ago
inventory	Prelim Exam	1 minute ago
test.yml	Prelim Exam	1 minute ago

README.md

Repani_PrelimExam

Prelim Exam for CPE 232

About

Prelim Exam for CPE 232

- [Readme](#)
- [Activity](#)
- 0 stars
- 1 watching
- 0 forks

Releases

No releases published
[Create a new release](#)

Packages

No packages published
[Publish your first package](#)

7. Your document report should be submitted here.

- In this Prelims Exam, I was able to use my knowledge gained from performing the previous activities and additionally with the help of non-stop online searching for references on different websites such as forums and tutorials like stackoverflow, github, and some youtube videos. By doing all these, I was able to achieve all the needed tasks successfully running without any errors. The process proved to be very challenging even with the references because not all the topics in the forums are exactly what I needed in this scenario but it did give me insight on what methods and commands I can use, even with unfamiliar ones, I simply had to search what the syntax of the command is, how to use it, and what it does.

8. For your prelim exam to be counted, please paste your repository link here.

https://github.com/JelzLow/Repani_PrelimExam

HONOR PLEDGE

"I affirm that I will not give or receive any unauthorized help on this exam, and that all work will be my own."