Guided Practice: Create Custom Python Modules

Outcome

In this Guided Practice, you will create a few custom Python modules. While Ansible comes with several modules, you might have the need to create your own. Since Ansible is written in Python, one of the natural and most efficient choices is to use Python for writing them (though Ansible allows you to write modules in any language). Modules are segments of code that can be triggered from a playbook.

Resources Needed

- VCASTLE Pod configured for the class. For this Guided Practice, we use the CentOS 8 and Ubuntu 20.04 LTS machines.
- Your user needs to be able to elevate their privileges with sudo.

Level of Difficulty

Moderate

Deliverables

Deliverables are marked in red font or with a red picture border around the screenshot. Additionally, there are questions at the end. **Your username or studentid should be visible in all screenshots that you submit.**

General Considerations

Ansible should already be installed on your machine from a prior Guided Practice.

Create Your First Python Module

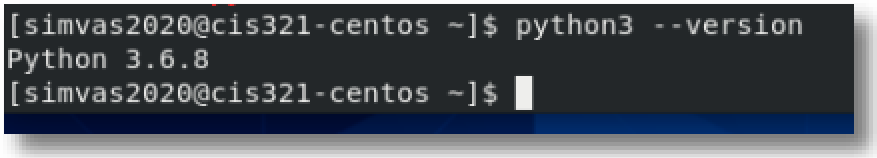
Modules are building blocks of playbooks. In order to create a Python module, you will need to use Python. Let's check that it is installed, and then we will use a text editor to write our very first custom Python module.

1. Check that Python is installed (it should be but let's check anyway).

   a. Log onto the CentOS computer using your own user.
   b. Open a terminal.
   c. Type:

```
python3 --version
```

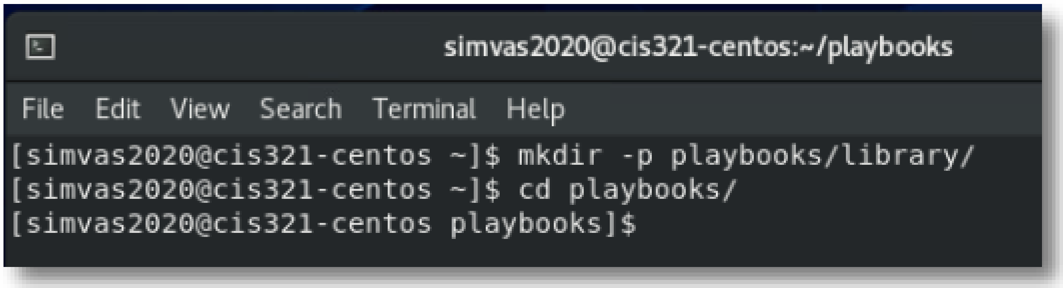
   Your screen should resemble this:



```
[simvas2020@cis321-centos ~]$ python3 --version
Python 3.6.8
[simvas2020@cis321-centos ~]$ 
```

2. We will write a Python module which checks if two users exist or not. Let's create a directory structure to organize our files first. Type:

```
mkdir -p playbooks/library
cd playbooks
```


simvas2020@cis321-centos:~/playbooks
File Edit View Search Terminal Help

```
[simvas2020@cis321-centos ~]$ mkdir -p playbooks/library/
[simvas2020@cis321-centos ~]$ cd playbooks/
[simvas2020@cis321-centos playbooks]$
```

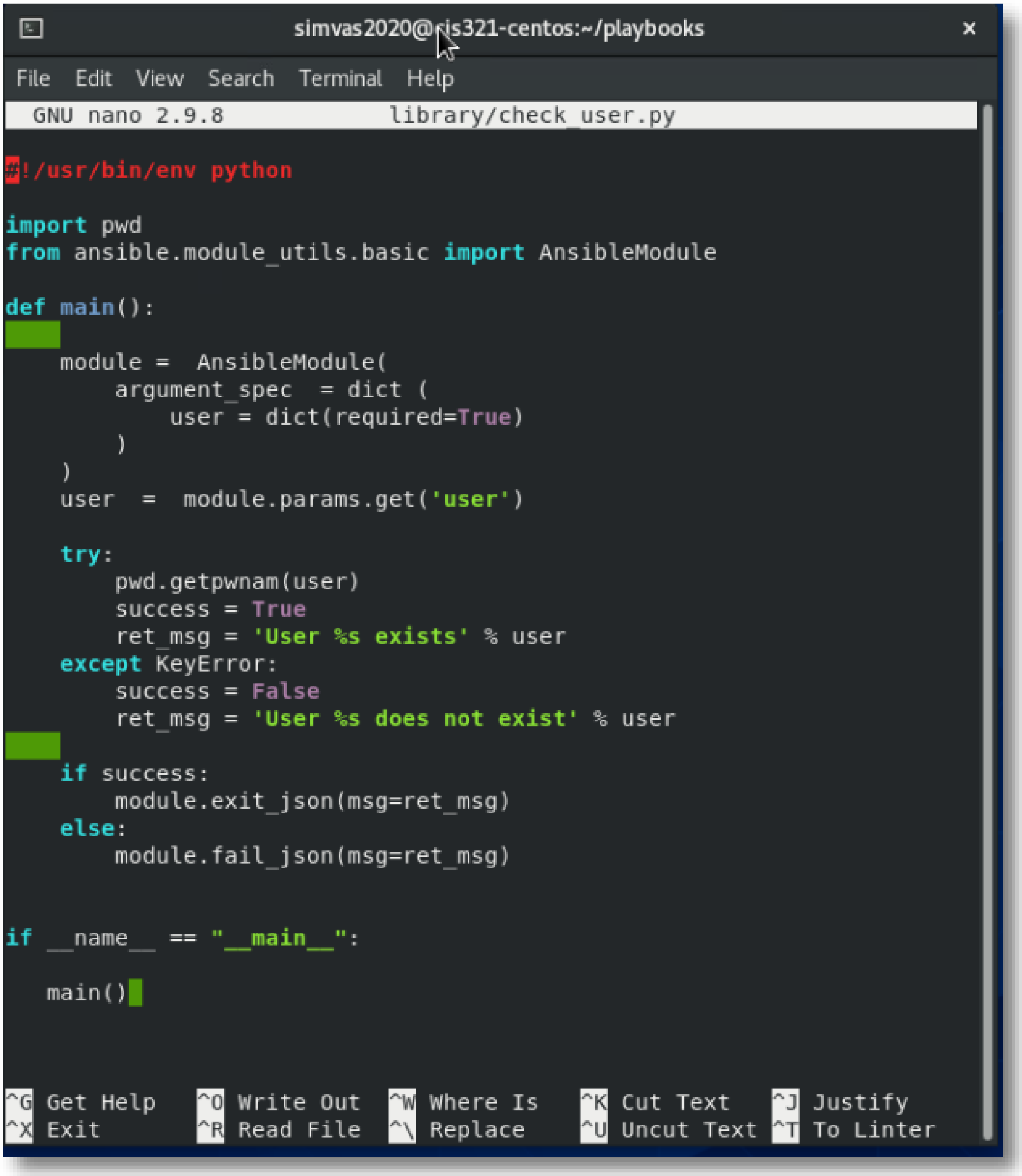
   a. Create and edit the file check_user.py using nano:

```
nano library/check_user.py
```

   b. Type the following in the new file. Then, save the file with Ctrl+S, and exit with Ctrl+X.

```
#!/usr/bin/env python


import pwd
```

```
from ansible.module_utils.basic import AnsibleModule

def main():

    module =  AnsibleModule(
        argument_spec  = dict (
            user = dict(required=True)
        )
    )
    user  =  module.params.get('user')

    try:
        pwd.getpwnam(user)
        success = True
        ret_msg = 'User %s exists' % user
    except KeyError:
        success = False
        ret_msg = 'User %s does not exist' % user

    if success:
        module.exit_json(msg=ret_msg)
    else:
        module.fail_json(msg=ret_msg)



if __name__ == "__main__":

   main()
```



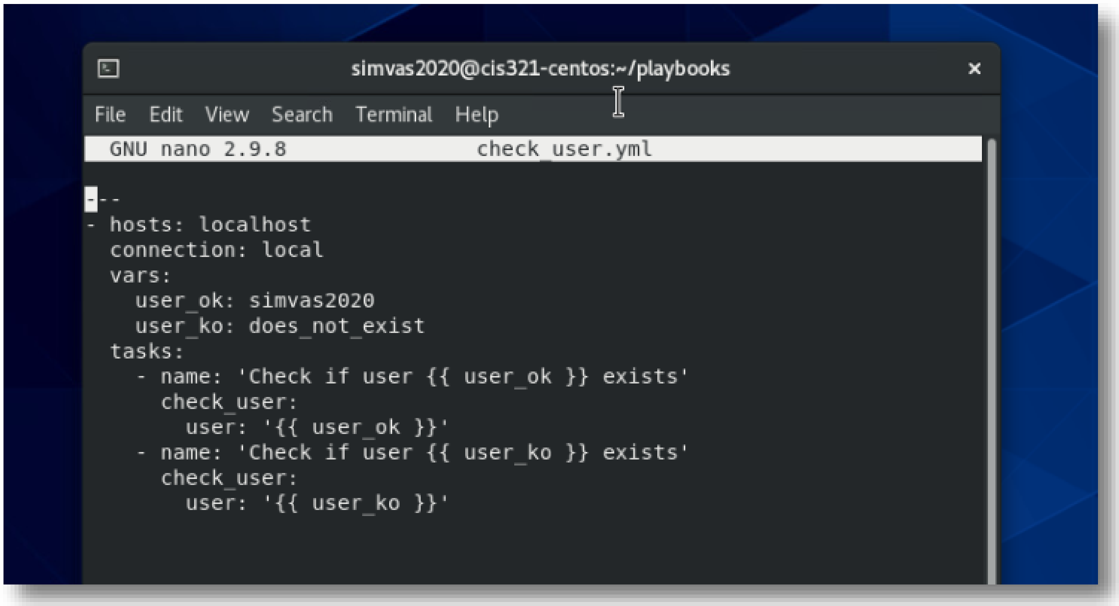
```
#!/usr/bin/env python

import pwd
from ansible.module_utils.basic import AnsibleModule

def main():

    module =  AnsibleModule(
        argument_spec  = dict (
            user = dict(required=True)
        )
    )
    user  =  module.params.get('user')

    try:
        pwd.getpwnam(user)
        success = True
        ret_msg = 'User %s exists' % user
    except KeyError:
        success = False
        ret_msg = 'User %s does not exist' % user

    if success:
        module.exit_json(msg=ret_msg)
    else:
        module.fail_json(msg=ret_msg)


if __name__ == "__main__":

   main()
```

c. You will now create the playbook to help you test this very simple custom Python module. In your CentOS terminal, type:

```
nano check_user.yml
```

d. Create the content below (replace simvas2020 in the screenshot below with your own studentID/username), then save the file by typing Ctrl+S, and exit nano by typing Ctrl+X.

```
---
- hosts: localhost
  connection: local
  vars:
    user_ok: simvas2020
    user_ko: does_not_exist
  tasks:
    - name: 'Check if user {{ user_ok }} exists'
      check_user:
        user: '{{ user_ok }}'
    - name: 'Check if user {{ user_ko }} exists'
      check_user:
        user: '{{ user_ko }}'
```

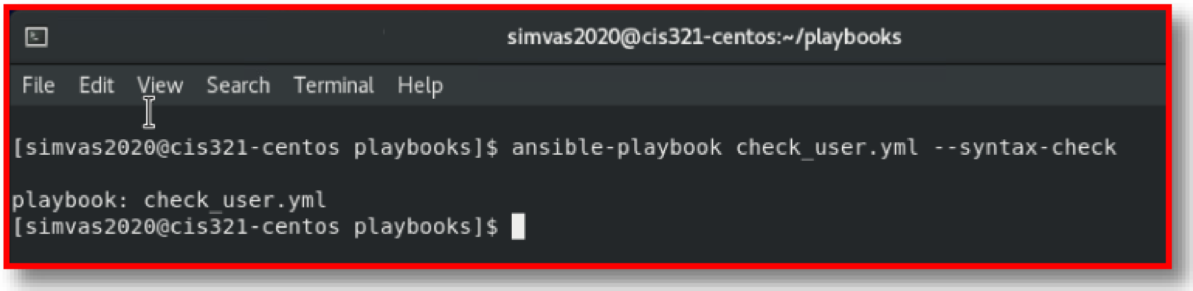


```
---
- hosts: localhost
  connection: local
  vars:
    user_ok: simvas2020
    user_ko: does_not_exist
  tasks:
    - name: 'Check if user {{ user_ok }} exists'
      check_user:
        user: '{{ user_ok }}'
    - name: 'Check if user {{ user_ko }} exists'
      check_user:
        user: '{{ user_ko }}'
```

3. Part of the focus of this week is to introduce you to some options to validate your code, and teach you how to debug it. Ansible offers you an option to check the syntax of a playbook for errors (syntax errors) before you run the playbook. That option is **--syntax-check**. In the CentOs terminal, type:

```
ansible-playbook check_user.yml --syntax-check
```

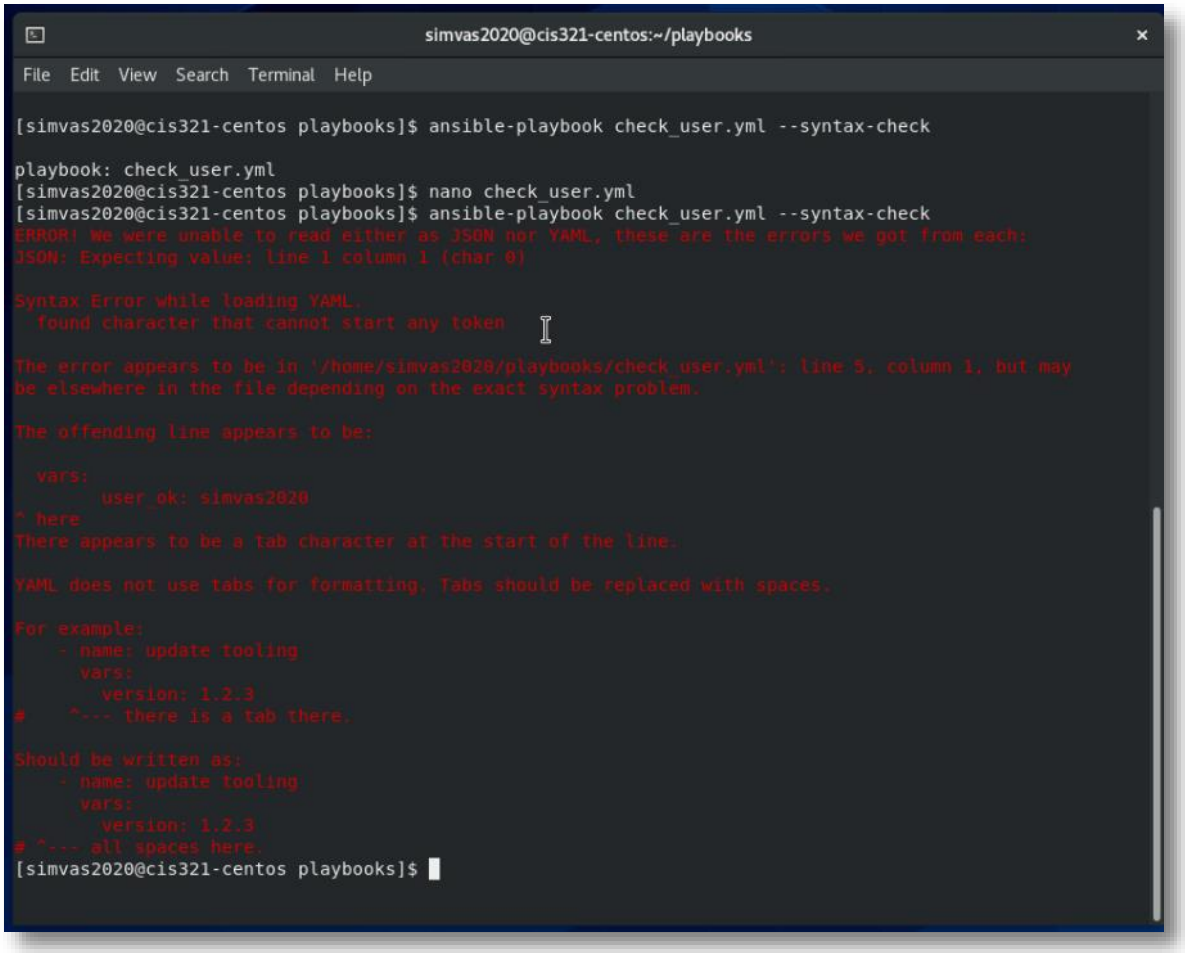
If your syntax is correct, your output should resemble the one below, but there might be errors to correct. For instance, YAML does not use tabs to format the code. So, if you used tabs, your output might look like the second screenshot, and you will have to use the provided error messages to try to correct your code. You will have to repeat the process until your code is error free.


simvas2020@cis321-centos:~/playbooks
File Edit View Search Terminal Help

```
[simvas2020@cis321-centos playbooks]$ ansible-playbook check_user.yml --syntax-check

playbook: check_user.yml
[simvas2020@cis321-centos playbooks]$ 
```

Take a screenshot for your lab report.

Example error:

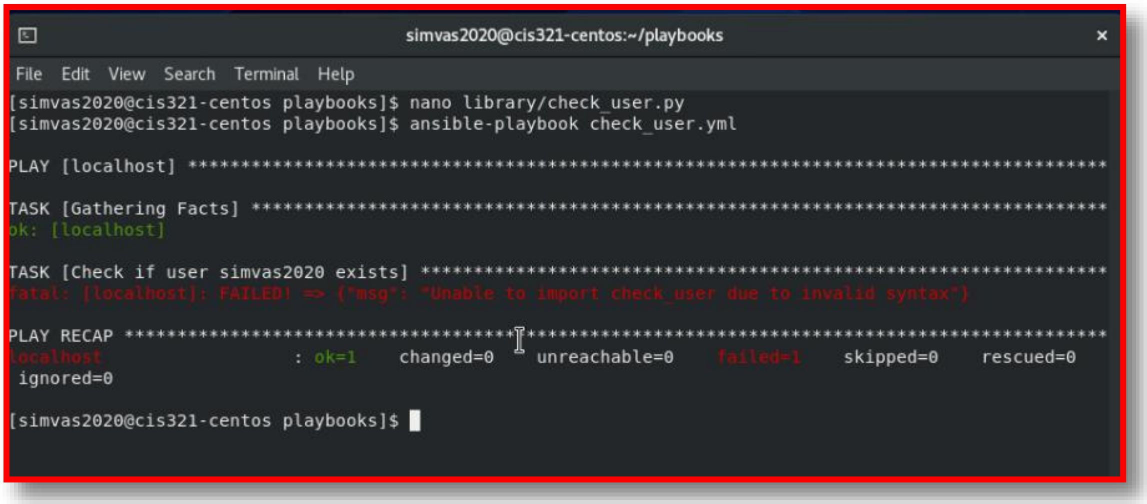

simvas2020@cis321-centos:~/playbooks
File Edit View Search Terminal Help

```
[simvas2020@cis321-centos playbooks]$ ansible-playbook check_user.yml --syntax-check

playbook: check_user.yml
[simvas2020@cis321-centos playbooks]$ nano check_user.yml
[simvas2020@cis321-centos playbooks]$ ansible-playbook check_user.yml --syntax-check
ERROR! We were unable to read either as JSON nor YAML, these are the errors we got from each:
JSON: Expecting value: line 1 column 1 (char 0)

Syntax Error while loading YAML.
  found character that cannot start any token

The error appears to be in '/home/simvas2020/playbooks/check_user.yml': line 5, column 1, but may
be elsewhere in the file depending on the exact syntax problem.

The offending line appears to be:

  vars:
        user_ok: simvas2020
^ here
There appears to be a tab character at the start of the line.

YAML does not use tabs for formatting. Tabs should be replaced with spaces.

For example:
    - name: update tooling
      vars:
        version: 1.2.3
#    ^--- there is a tab there.

Should be written as:
    - name: update tooling
      vars:
        version: 1.2.3
# ^--- all spaces here.
[simvas2020@cis321-centos playbooks]$ 
```

4. After correcting the syntax of your YAML code, you can try to run the playbook by typing:

```
ansible-playbook check_user.yml
```

5. However, you might encounter some issues with your Python module syntax. If that is the case, you will be seeing something like the following image:


simvas2020@cis321-centos:~/playbooks
File Edit View Search Terminal Help

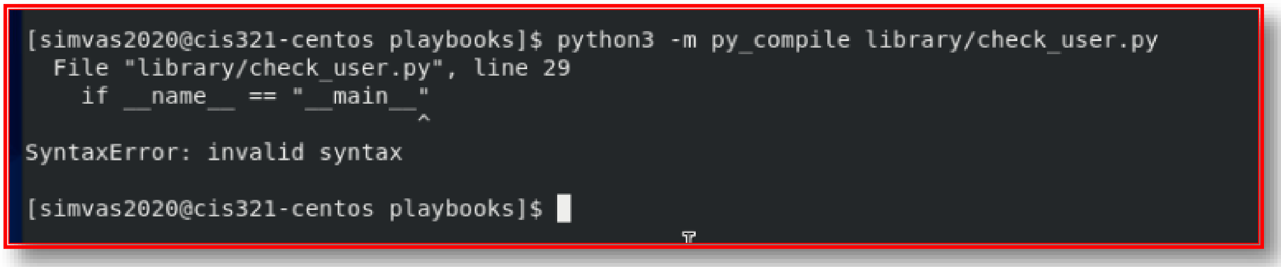
```
[simvas2020@cis321-centos playbooks]$ nano library/check_user.py
[simvas2020@cis321-centos playbooks]$ ansible-playbook check_user.yml

PLAY [localhost] *************************************************************************

TASK [Gathering Facts] *******************************************************************
ok: [localhost]

TASK [Check if user simvas2020 exists] ***************************************************
fatal: [localhost]: FAILED! => {"msg": "Unable to import check_user due to invalid syntax"}

PLAY RECAP *******************************************************************************
localhost                  : ok=1    changed=0    unreachable=0    failed=1    skipped=0    rescued=0
 ignored=0

[simvas2020@cis321-centos playbooks]$ 
```

Take a screenshot for your lab report.

6. The output above is not terribly insightful, and you might spend some time trying to look for where the error is. Instead of doing that, let us compile the Python code, and see if we can get closer to where the problem is. Type

   ```
   python3 -m py_compile library/check_user.py
   ```

   In the screenshot below, it is much easier to observe where the error is: **a :** was forgotten at the end of the line.

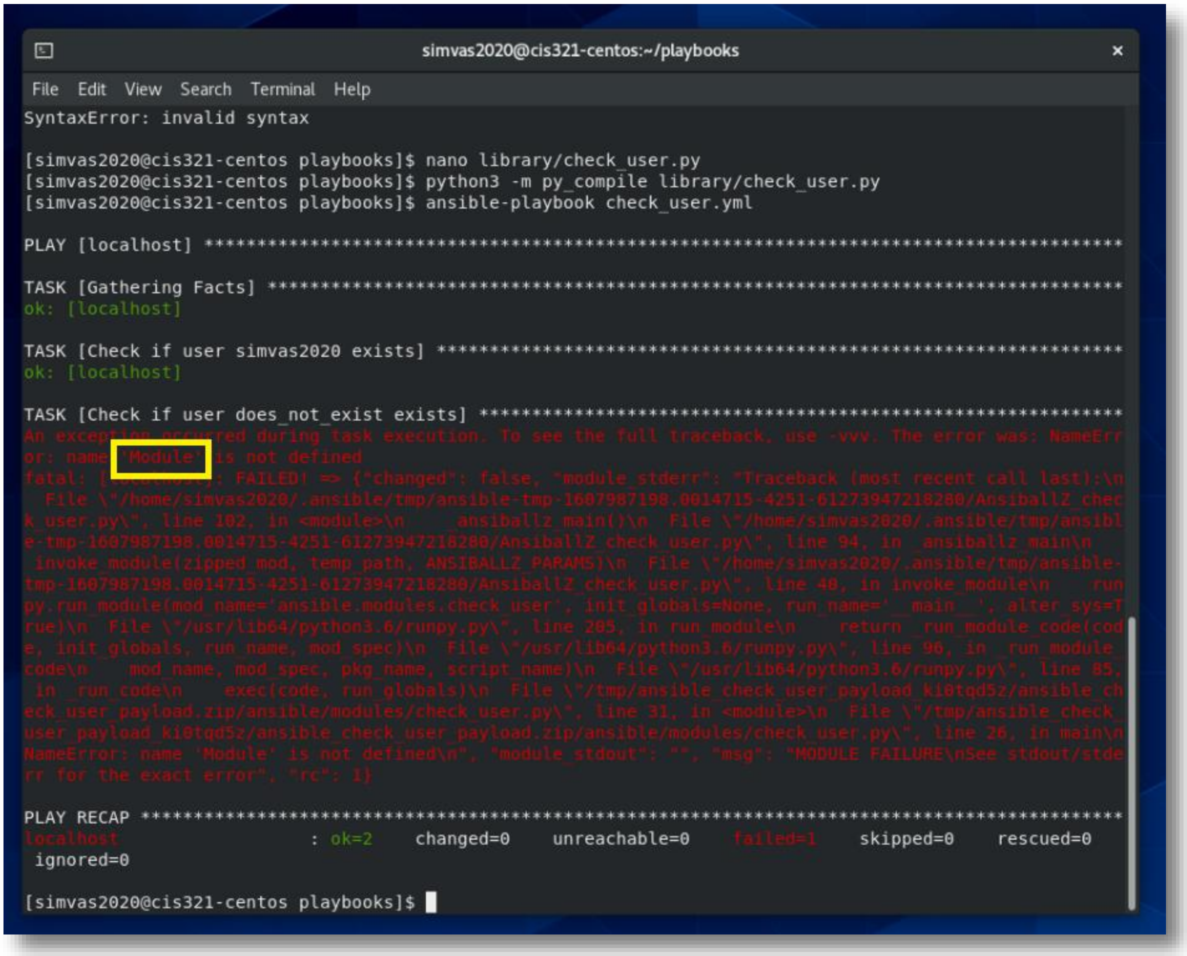


```
[simvas2020@cis321-centos playbooks]$ python3 -m py_compile library/check_user.py
  File "library/check_user.py", line 29
    if __name__ == "__main__"
                            ^
SyntaxError: invalid syntax

[simvas2020@cis321-centos playbooks]$ 
```

   Take a screenshot for your lab report.

   Fix the Python code syntax errors, and attempt to compile again until the code is error free.

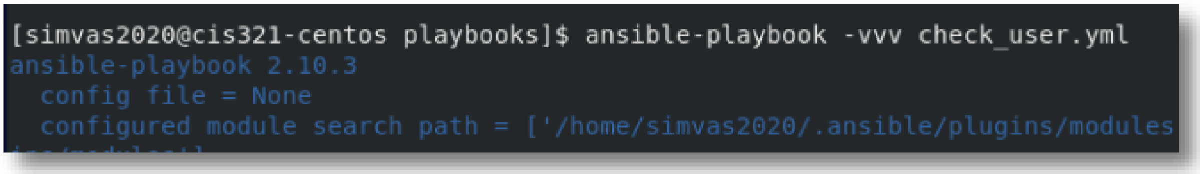
7. Even after compiling the Python code and using the **--syntax-check** option of the ansible-playbook command, you might STILL have problems running your code. Please see an example in the screenshot below:


simvas2020@cis321-centos:~/playbooks
File Edit View Search Terminal Help

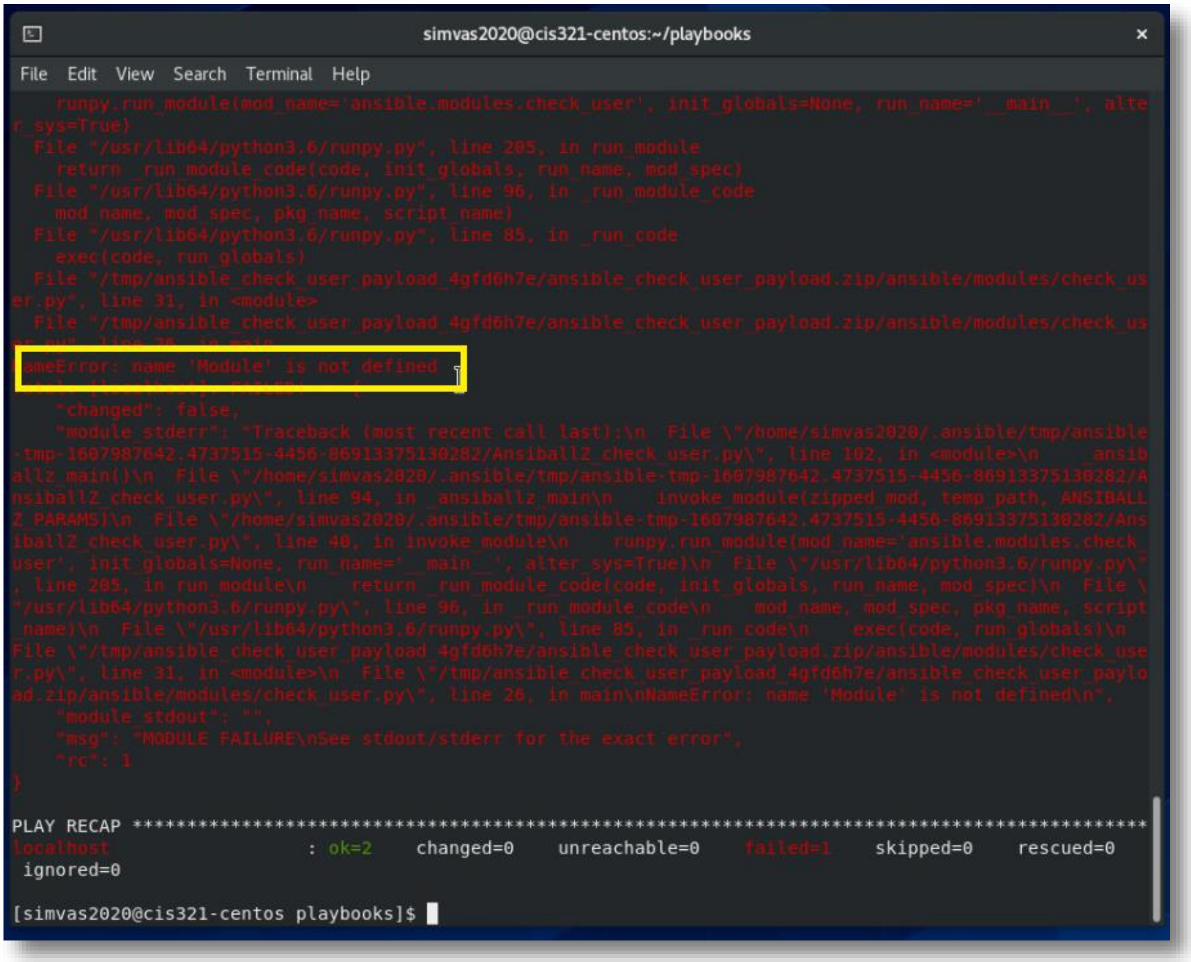
```
SyntaxError: invalid syntax

[simvas2020@cis321-centos playbooks]$ nano library/check_user.py
[simvas2020@cis321-centos playbooks]$ python3 -m py_compile library/check_user.py
[simvas2020@cis321-centos playbooks]$ ansible-playbook check_user.yml

PLAY [localhost] ********************************************************************************

TASK [Gathering Facts] **************************************************************************
ok: [localhost]

TASK [Check if user simvas2020 exists] **********************************************************
ok: [localhost]

TASK [Check if user does_not_exist exists] ******************************************************
An exception occurred during task execution. To see the full traceback, use -vvv. The error was: NameErr
or: name 'Module' is not defined
fatal: [localhost]: FAILED! => {"changed": false, "module_stderr": "Traceback (most recent call last):\n
  File \"/home/simvas2020/.ansible/tmp/ansible-tmp-1607987198.0014715-4251-61273947218280/AnsiballZ_chec
k_user.py\", line 102, in <module>\n    _ansiballz_main()\n  File \"/home/simvas2020/.ansible/tmp/ansibl
e-tmp-1607987198.0014715-4251-61273947218280/AnsiballZ_check_user.py\", line 94, in _ansiballz_main\n
 invoke_module(zipped_mod, temp_path, ANSIBALLZ_PARAMS)\n  File \"/home/simvas2020/.ansible/tmp/ansible-
tmp-1607987198.0014715-4251-61273947218280/AnsiballZ_check_user.py\", line 40, in invoke_module\n    run
py.run_module(mod_name='ansible.modules.check_user', init_globals=None, run_name='__main__', alter_sys=T
rue)\n  File \"/usr/lib64/python3.6/runpy.py\", line 205, in run_module\n    return _run_module_code(cod
e, init_globals, run_name, mod_spec)\n  File \"/usr/lib64/python3.6/runpy.py\", line 96, in _run_module_
code\n    mod_name, mod_spec, pkg_name, script_name)\n  File \"/usr/lib64/python3.6/runpy.py\", line 85,
 in _run_code\n    exec(code, run_globals)\n  File \"/tmp/ansible_check_user_payload_ki0tqd5z/ansible_ch
eck_user_payload.zip/ansible/modules/check_user.py\", line 31, in <module>\n  File \"/tmp/ansible_check_
user_payload_ki0tqd5z/ansible_check_user_payload.zip/ansible/modules/check_user.py\", line 26, in main\n
NameError: name 'Module' is not defined\n", "module_stdout": "", "msg": "MODULE FAILURE\nSee stdout/stde
rr for the exact error", "rc": 1}

PLAY RECAP **************************************************************************************
localhost                  : ok=2    changed=0    unreachable=0    failed=1    skipped=0    rescued=0
 ignored=0

[simvas2020@cis321-centos playbooks]$
```

8. You can try to correct the error where it occurs by looking at the error message Ansible provided you with, or you can use the –vvv option. Type:

```
ansible-playbook –vvv check_user.yml
```



```
[simvas2020@cis321-centos playbooks]$ ansible-playbook -vvv check_user.yml
ansible-playbook 2.10.3
  config file = None
  configured module search path = ['/home/simvas2020/.ansible/plugins/modules
```

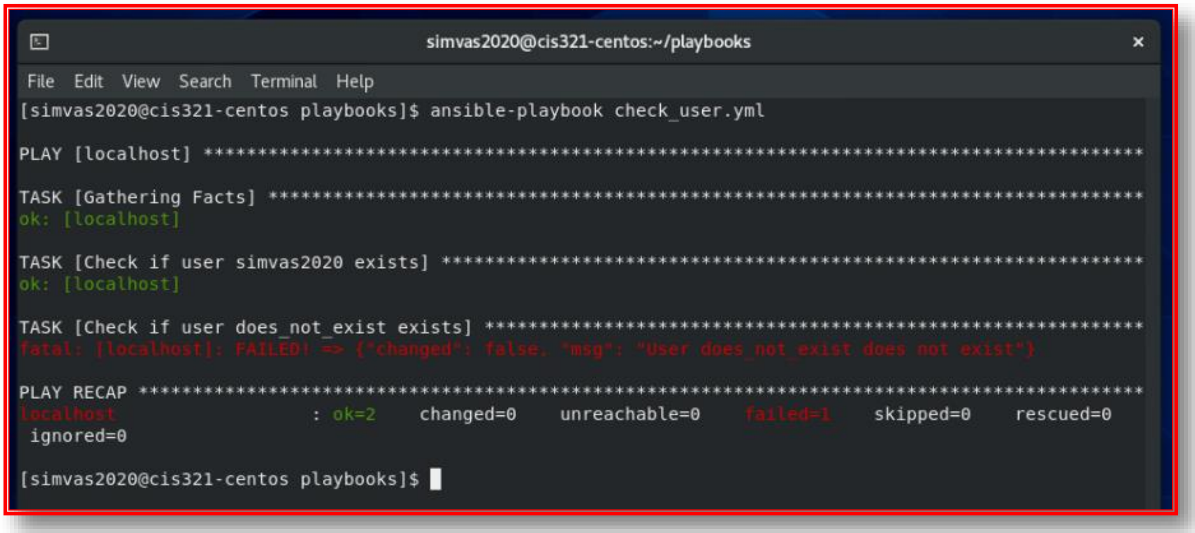


simvas2020@cis321-centos:~/playbooks
File Edit View Search Terminal Help

```
    runpy.run_module(mod_name='ansible.modules.check_user', init_globals=None, run_name='__main__', alte
r_sys=True)
  File "/usr/lib64/python3.6/runpy.py", line 205, in run_module
    return _run_module_code(code, init_globals, run_name, mod_spec)
  File "/usr/lib64/python3.6/runpy.py", line 96, in _run_module_code
    mod_name, mod_spec, pkg_name, script_name)
  File "/usr/lib64/python3.6/runpy.py", line 85, in _run_code
    exec(code, run_globals)
  File "/tmp/ansible_check_user_payload_4gfd6h7e/ansible_check_user_payload.zip/ansible/modules/check_us
er.py", line 31, in <module>
  File "/tmp/ansible_check_user_payload_4gfd6h7e/ansible_check_user_payload.zip/ansible/modules/check_us
er.py", line 26, in main
NameError: name 'Module' is not defined
fatal: [localhost]: FAILED! => {
    "changed": false,
    "module_stderr": "Traceback (most recent call last):\n  File \"/home/simvas2020/.ansible/tmp/ansible
-tmp-1607987642.4737515-4456-86913375130282/AnsiballZ_check_user.py\", line 102, in <module>\n    _ansib
allz_main()\n  File \"/home/simvas2020/.ansible/tmp/ansible-tmp-1607987642.4737515-4456-86913375130282/A
nsiballZ_check_user.py\", line 94, in _ansiballz_main\n    invoke_module(zipped_mod, temp_path, ANSIBALL
Z_PARAMS)\n  File \"/home/simvas2020/.ansible/tmp/ansible-tmp-1607987642.4737515-4456-86913375130282/Ans
iballZ_check_user.py\", line 40, in invoke_module\n    runpy.run_module(mod_name='ansible.modules.check_
user', init_globals=None, run_name='__main__', alter_sys=True)\n  File \"/usr/lib64/python3.6/runpy.py\"
, line 205, in run_module\n    return _run_module_code(code, init_globals, run_name, mod_spec)\n  File \
"/usr/lib64/python3.6/runpy.py\", line 96, in _run_module_code\n    mod_name, mod_spec, pkg_name, script
_name)\n  File \"/usr/lib64/python3.6/runpy.py\", line 85, in _run_code\n    exec(code, run_globals)\n
File \"/tmp/ansible_check_user_payload_4gfd6h7e/ansible_check_user_payload.zip/ansible/modules/check_use
r.py\", line 31, in <module>\n  File \"/tmp/ansible_check_user_payload_4gfd6h7e/ansible_check_user_paylo
ad.zip/ansible/modules/check_user.py\", line 26, in main\nNameError: name 'Module' is not defined\n",
    "module_stdout": "",
    "msg": "MODULE FAILURE\nSee stdout/stderr for the exact error",
    "rc": 1
}

PLAY RECAP *********************************************************************************
localhost                  : ok=2    changed=0    unreachable=0    failed=1    skipped=0    rescued=0
 ignored=0

[simvas2020@cis321-centos playbooks]$ 
```

Correct these errors as well and run your playbook.

9. Your output should resemble the one below:



simvas2020@cis321-centos:~/playbooks
File Edit View Search Terminal Help

```
[simvas2020@cis321-centos playbooks]$ ansible-playbook check_user.yml

PLAY [localhost] ***************************************************************************

TASK [Gathering Facts] *********************************************************************
ok: [localhost]

TASK [Check if user simvas2020 exists] *****************************************************
ok: [localhost]

TASK [Check if user does_not_exist exists] *************************************************
fatal: [localhost]: FAILED! => {"changed": false, "msg": "User does not exist does not exist"}

PLAY RECAP *********************************************************************************
localhost                  : ok=2    changed=0    unreachable=0    failed=1    skipped=0    rescued=0
 ignored=0

[simvas2020@cis321-centos playbooks]$ 
```

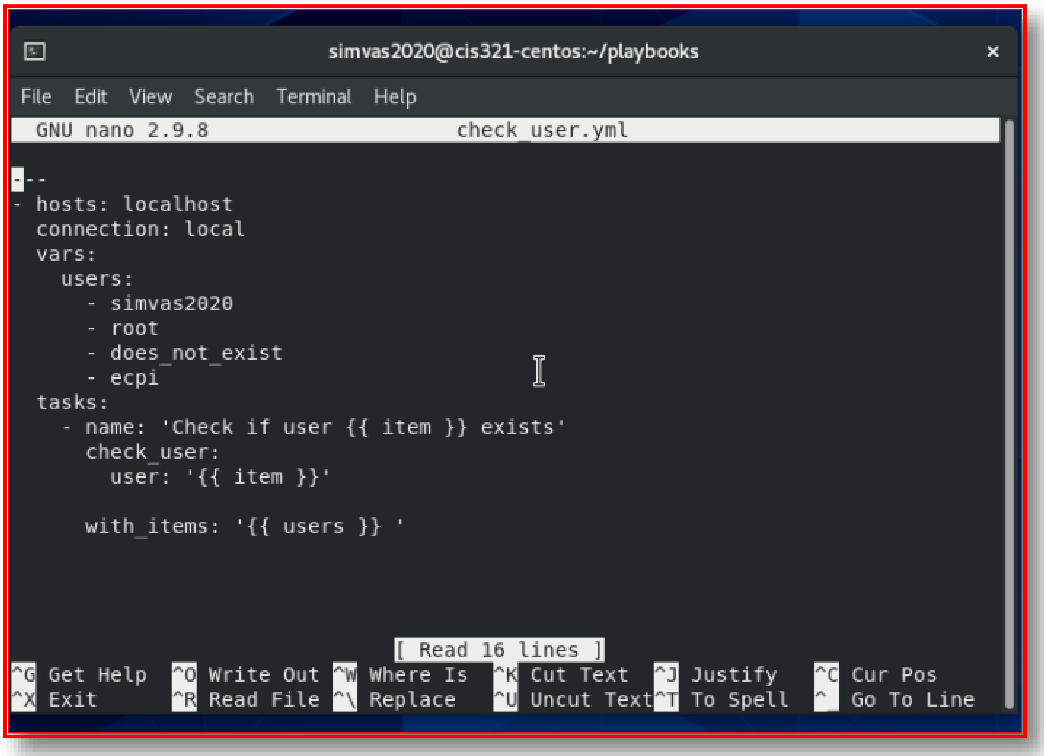
Take a screenshot for your lab report.

10. Let us slightly change the playbook, so that it loops through a list of users.
    a. First off, let us make a copy of the .yml file as a backup. Type:

```
cp check_user.yml check_user.bkp
```

    b. Edit the check_user.yml with nano. The goal is to use a list of users that the playbook can loop through. After you modified the file, save it with Ctrl+S, close with Ctrl+X, and check syntax again.

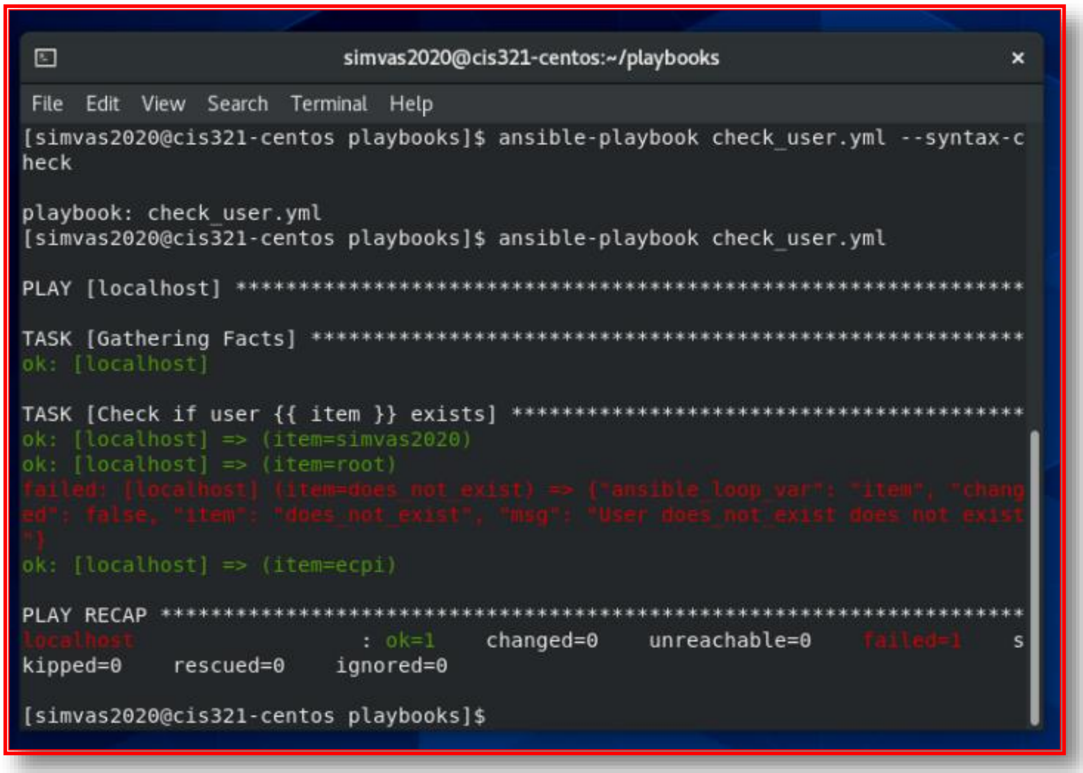
```
---
- hosts: localhost
  connection: local
  vars:
    users:
      - simvas2020
      - root
      - does_not_exist
      - ecpi

  tasks:
    - name: 'Check if {{ item }} exists'
      check_user:
        user: '{{ item }}'
      with_items: '{{ users }}'
```


simvas2020@cis321-centos:~/playbooks
File Edit View Search Terminal Help

```
  GNU nano 2.9.8                    check_user.yml

---
- hosts: localhost
  connection: local
  vars:
    users:
      - simvas2020
      - root
      - does_not_exist
      - ecpi
  tasks:
    - name: 'Check if user {{ item }} exists'
      check_user:
        user: '{{ item }}'

      with_items: '{{ users }} '




                                [ Read 16 lines ]
^G Get Help   ^O Write Out  ^W Where Is   ^K Cut Text   ^J Justify    ^C Cur Pos
^X Exit       ^R Read File  ^\ Replace    ^U Uncut Text ^T To Spell   ^_ Go To Line
```

Take a screenshot for your lab report.

c. After you corrected any new errors you might have made, run the playbook again. Your output should look like the one below.



simvas2020@cis321-centos:~/playbooks
File Edit View Search Terminal Help

```
[simvas2020@cis321-centos playbooks]$ ansible-playbook check_user.yml --syntax-c
heck

playbook: check_user.yml
[simvas2020@cis321-centos playbooks]$ ansible-playbook check_user.yml

PLAY [localhost] ***************************************************************

TASK [Gathering Facts] *********************************************************
ok: [localhost]

TASK [Check if user {{ item }} exists] *****************************************
ok: [localhost] => (item=simvas2020)
ok: [localhost] => (item=root)
failed: [localhost] (item=does_not_exist) => {"ansible_loop_var": "item", "chang
ed": false, "item": "does_not_exist", "msg": "User does_not_exist does not exist
"}
ok: [localhost] => (item=ecpi)

PLAY RECAP *********************************************************************
localhost                  : ok=1    changed=0    unreachable=0    failed=1    s
kipped=0    rescued=0    ignored=0

[simvas2020@cis321-centos playbooks]$
```

Take a screenshot for your lab report.

Guided Practice Questions

In your **Guided Practice Lab Report**, in addition to the screenshots, include answers for the following questions about this learning activity. Some may require research.

1. List at least two formatting/syntax guidelines for YAML and two for Python.
2. What were some of the errors you received, and how did you fix them? After researching the syntax/formatting guidelines for question 1, can you explain why you received the errors and how you fixed them? **(An explanation is necessary – do not answer YES/NO.)**
3. What is an Ansible custom module?