

Group Variables

1. Now here for this lab come to your latest file directory. In my case it is exc7.
2. Then do a listing of it. Now create new directory with name group_vars and a file in that with name 'all'.

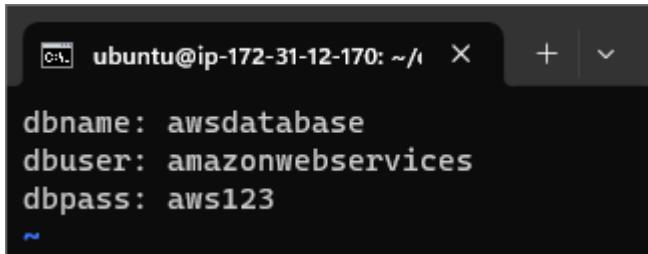
```
mkdir group_vars  
vim group_vars/all
```



```
ubuntu@ip-172-31-12-170:~/demodirectory/exc7$ ls  
demodb.yaml demousrkey.pem inventory  
ubuntu@ip-172-31-12-170:~/demodirectory/exc7$ mkdir group_vars  
ubuntu@ip-172-31-12-170:~/demodirectory/exc7$ vim group_vars/all
```

3. In the 'all' file you have to write all the three variables that you have defined in the demodb.yaml file just keep the values different.

```
dbname: awsdatabase  
dbuser: amazonwebservices  
dbpass: aws123
```



```
ubuntu@ip-172-31-12-170:~/demodirectory/exc7$ cat group_vars/all  
dbname: awsdatabase  
dbuser: amazonwebservices  
dbpass: aws123  
~
```

4. So, now when you run the ansible playbook command. You'll see that it is using the playbook variables which means that playbook has the higher priority.

```
ansible-playbook -i inventory demodb.yaml
```

```

ubuntu@ip-172-31-12-170:~/demodirectory/exc7$ vim group_vars/all
ubuntu@ip-172-31-12-170:~/demodirectory/exc7$ ansible-playbook -i inventory demodb.yaml

PLAY [DBserver setup] ****
TASK [Gathering Facts] ****
ok: [demodb01]

TASK [debug] ****
ok: [demodb01] => {
    "msg": " The dbname is ansible"
}

TASK [debug] ****
ok: [demodb01] => {
    "dbuser": "demouser"
}

```

5. Now you need to open the playbook and comment out the variables in it. After that come back and run the ansible playbook command. So, what will happen it will try to find the variables but it will be unable to find those in the playbook because they are commented out. So, this time it will go to the group_vars directory and pick the variables from there.
6. Remember to comment out only these variables and you are good to go.

```

---
- name: DBserver setup
  hosts: dbservers
  become: yes
  #vars:
    #dbname: ansible
    #dbuser: demouser
    #dbpass: demo

```

7. This time you will see that the message has changed which means that it has picked the variables from group_vars.

vim demodb.yaml
ansible-playbook -i inventory demodb.yaml

```

ubuntu@ip-172-31-12-170:~/demodirectory/exc7$ vim demodb.yaml
ubuntu@ip-172-31-12-170:~/demodirectory/exc7$ ansible-playbook -i inventory demodb.yaml

PLAY [DBserver setup] ****
TASK [Gathering Facts] ****
ok: [demodb01]

TASK [debug] ****
ok: [demodb01] => {
    "msg": " The dbname is awsdatabase"
}

TASK [debug] ****
ok: [demodb01] => {
    "dbuser": "amazonwebservices"
}

```

8. Now create a new exercise and get into it.
9. In this new exercise remove the playbook and remove the group_vars/all.

```
cp -r exc7/ exc8
cd exc8
rm -rf demodb.yaml
rm -rf group_vars
```

```
ubuntu@ip-172-31-12-170:~/demodirectory/exc7$ ls
demodb.yaml  demousrkey.pem  group_vars  inventory
ubuntu@ip-172-31-12-170:~/demodirectory/exc7$ cd ..
ubuntu@ip-172-31-12-170:~/demodirectory$ ls
exc1  exc2  exc3  exc4  exc5  exc6  exc7
ubuntu@ip-172-31-12-170:~/demodirectory$ cp -r exc7/ exc8
ubuntu@ip-172-31-12-170:~/demodirectory$ ls
exc1  exc2  exc3  exc4  exc5  exc6  exc7  exc8
ubuntu@ip-172-31-12-170:~/demodirectory$ cd exc8

ubuntu@ip-172-31-12-170:~/demodirectory/exc8$ rm -rf demodb.yaml
ubuntu@ip-172-31-12-170:~/demodirectory/exc8$ rm -rf group_vars
ubuntu@ip-172-31-12-170:~/demodirectory/exc8$ ls
demousrkey.pem  inventory
ubuntu@ip-172-31-12-170:~/demodirectory/exc8$ |
```

10. After that you need to create a new playbook.

```
vim demo_prec.yaml
```

```
ubuntu@ip-172-31-12-170:~/demodirectory/exc8$ vim demo_prec.yaml
```

11. After write this code into it. Then just save and quit.

```
---
- name: Understanding vars
  hosts: all
  become: yes
  gather_facts: False
  vars:
    USRNM: demouser
    COMM: ansible playbook variable
  tasks:
    - name: create user
      ansible.builtin.user:
        name: "{{USRNM}}"
        comment: "{{COMM}}"
```

```
ubuntu@ip-172-31-12-170: ~/i + | v

---
- name: Understanding vars
  hosts: all
  become: yes
  gather_facts: False
  vars:
    USRNM: demouser
    COMM: ansible playbook variable
  tasks:
    - name: create user
      ansible.builtin.user:
        name: "{{USRNM}}"
        comment: "{{COMM}}"
```

12. Now run the ansible playbook command.

```
ansible-playbook -i inventory demo_prec.yaml
```

```
ubuntu@ip-172-31-12-170:~/demodirectory/exc8$ ansible-playbook -i inventory demo_prec.yaml

PLAY [Understanding vars] ****
TASK [create user] ****
changed: [demodb01]
changed: [demoweb01]
changed: [demoweb02]

PLAY RECAP ****
demodb01 : ok=1    changed=1    unreachable=0    failed=0    skipped=0    rescued=0
            ignored=0
demoweb01 : ok=1    changed=1    unreachable=0    failed=0    skipped=0    rescued=0
            ignored=0
demoweb02 : ok=1    changed=1    unreachable=0    failed=0    skipped=0    rescued=0
            ignored=0

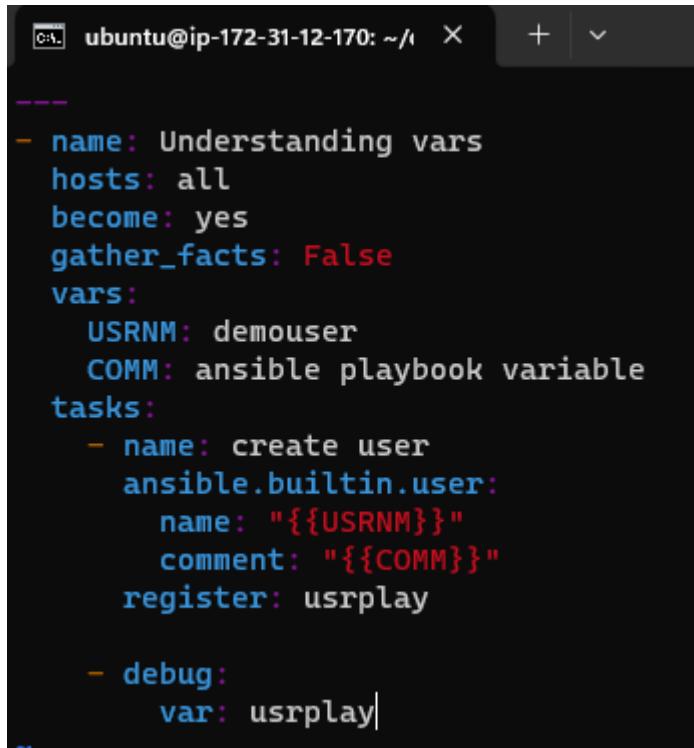
ubuntu@ip-172-31-12-170:~/demodirectory/exc8$
```

13. Now you need to update the code.

```
---
- name: Understanding vars
  hosts: all
  become: yes
  gather_facts: False
  vars:
    USRNM: demouser
    COMM: ansible playbook variable
  tasks:
    - name: create user
      ansible.builtin.user:
```

```
name: "{{USRNM}}"
comment: "{{COMM}}"
register: usrplay
```

```
- debug:
  var: usrplay
```



A screenshot of a terminal window titled "ubuntu@ip-172-31-12-170: ~/" showing an Ansible playbook file. The file contains a single play with one task. The task creates a user with the name {{USRNM}} and comment {{COMM}}, registers the result to a variable named "usrplay", and includes a debug step to output the value of "usrplay".

```
---  
- name: Understanding vars  
  hosts: all  
  become: yes  
  gather_facts: False  
  vars:  
    USRNM: demouser  
    COMM: ansible playbook variable  
  tasks:  
    - name: create user  
      ansible.builtin.user:  
        name: "{{USRNM}}"
        comment: "{{COMM}}"
      register: usrplay  
  
    - debug:  
      var: usrplay
```

14. And run the ansible playbook command. You will see that you getting the output for all three of your machines.

```
ansible-playbook -i inventory demo_prec.yaml
```

```

ubuntu@ip-172-31-12-170:~/demodirectory/exc8$ ansible-playbook -i inventory demo_prec.yaml

PLAY [Understanding vars] ****
TASK [create user] ****
ok: [demodb01]
ok: [demoweb02]

TASK [debug] ****
"usrplay": {
    "ansible_facts": {
        "discovered_interpreter_python": "/usr/bin/python3"
    },
    "append": false,
    "changed": false,
    "comment": "ansible playbook variable",
    "failed": false,
    "group": 1001,
    "home": "/home/demouser",
    "move_home": false,
    "name": "demouser",
    "shell": "/bin/bash",
    "state": "present",
    "uid": 1001
}
ok: [demoweb02] => {
"usrplay": {
    "ansible_facts": {
        "discovered_interpreter_python": "/usr/bin/python3"
    },
    "append": false,
    "changed": false,
    "comment": "ansible playbook variable",
    "failed": false,
    "group": 1001,
    "home": "/home/demouser",
    "move_home": false,
    "name": "demouser",
    "shell": "/bin/bash",
    "state": "present",
    "uid": 1001
}
}
ok: [demodb01] => {
"usrplay": {
    "ansible_facts": {
        "discovered_interpreter_python": "/usr/bin/python3"
    },
    "append": false,
    "changed": false,
}
}

```

15. Now if you make the little change in the code, you will get the same output but this time it will be smaller.

```

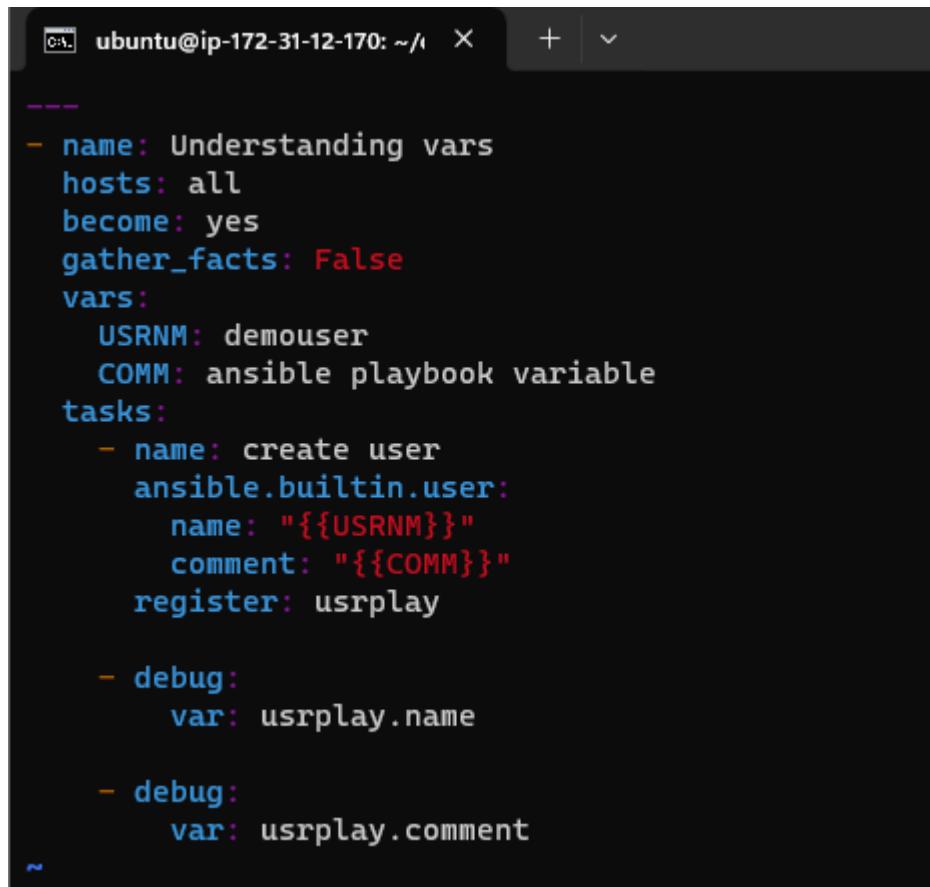
---
- name: Understanding vars
  hosts: all
  become: yes
  gather_facts: False
  vars:
    USRNM: demouser
    COMM: ansible playbook variable
  tasks:
    - name: create user
      ansible.builtin.user:
        name: "{{USRNM}}"
        comment: "{{COMM}}"
        register: usrplay

    - debug:

```

```
var: usrplay.name

- debug:
  var: usrplay.comment
```



The screenshot shows a terminal window with the title bar "ubuntu@ip-172-31-12-170: ~/" and a status bar with a "+" and a downward arrow. The terminal content displays an Ansible playbook file:

```
---
- name: Understanding vars
  hosts: all
  become: yes
  gather_facts: False
  vars:
    USRNM: demouser
    COMM: ansible playbook variable
  tasks:
    - name: create user
      ansible.builtin.user:
        name: "{{USRNM}}"
        comment: "{{COMM}}"
        register: usrplay

    - debug:
        var: usrplay.name

    - debug:
        var: usrplay.comment
```

16. Now run the ansible playbook command.

```
ansible-playbook -i inventory demo_prec.yaml
```

```
ubuntu@ip-172-31-12-170:~/demodirectory/exc8$ ansible-playbook -i inventory demo_prec.yaml

PLAY [Understanding vars] ****
TASK [create user] ****
ok: [demodb01]
ok: [demoweb01]
ok: [demoweb02]

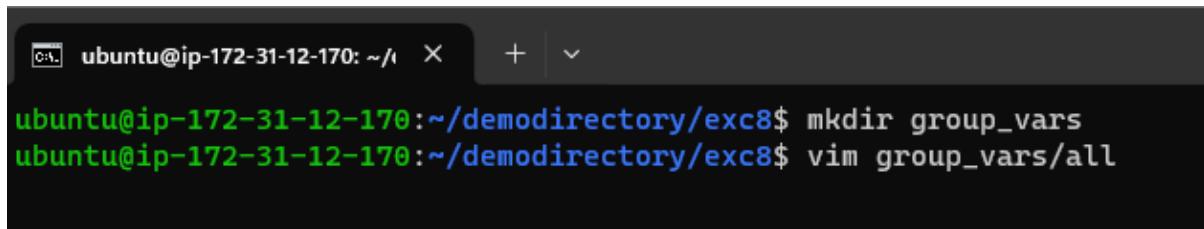
TASK [debug] ****
ok: [demoweb01] => {
    "usrplay.name": "demouser"
}
ok: [demoweb02] => {
    "usrplay.name": "demouser"
}
ok: [demodb01] => {
    "usrplay.name": "demouser"
}

TASK [debug] ****
ok: [demoweb01] => {
    "usrplay.comment": "ansible playbook variable"
}
ok: [demoweb02] => {
    "usrplay.comment": "ansible playbook variable"
}
ok: [demodb01] => {
    "usrplay.comment": "ansible playbook variable"
}

PLAY RECAP ****
demodb01 : ok=3    changed=0    unreachable=0    failed=0    skipped=0    rescued=0
    ignored=0
demoweb01 : ok=3    changed=0    unreachable=0    failed=0    skipped=0    rescued=0
    ignored=0
demoweb02 : ok=3    changed=0    unreachable=0    failed=0    skipped=0    rescued=0
    ignored=0
```

17. Now make a directory for group_vars and make a file in it with name all.

```
mkdir group_vars
vim group_vars/all
```



```
ubuntu@ip-172-31-12-170:~/demodirectory/exc8$ mkdir group_vars
ubuntu@ip-172-31-12-170:~/demodirectory/exc8$ vim group_vars/all
```

18. Now you need to write the variables.

USRNM: awsuser
COMM: variable from aws console.

```
ubuntu@ip-172-31-12-170: ~/ | + | ~
USRNM: awsuser
COMM: varialbe from aws console
~
```

```
ubuntu@ip-172-31-12-170:~/demodirectory/exc8$ cat group_vars/all
USRNM: awsuser
COMM: varialbe from aws console
ubuntu@ip-172-31-12-170:~/demodirectory/exc8$ cat demo_prec.yaml
---
- name: Understanding vars
  hosts: all
  become: yes
  gather_facts: False
  vars:
    USRNM: demouser
    COMM: ansible playbook variable
  tasks:
    - name: create user
      ansible.builtin.user:
        name: "{{USRNM}}"
        comment: "{{COMM}}"
      register: usrplay

    - debug:
        var: usrplay.name

    - debug:
        var: usrplay.comment
ubuntu@ip-172-31-12-170:~/demodirectory/exc8$ |
```

19. Now if you run the ansible playbook command. You will see it is again getting the variables from playbook not from group vars. It means that playbook variables has more precedence.

```
ansible-playbook -I inventory demo_prec.yaml
```

```

ubuntu@ip-172-31-12-170:~/demodirectory/exc8$ ansible-playbook -i inventory demo_prec.yaml

PLAY [Understanding vars] ****
TASK [create user] ****
ok: [demoweb01]
ok: [demodb01]
ok: [demoweb02]

TASK [debug] ****
ok: [demoweb01] => {
    "usrplay.name": "demouser"
}
ok: [demoweb02] => {
    "usrplay.name": "demouser"
}
ok: [demodb01] => {
    "usrplay.name": "demouser"
}

TASK [debug] ****
ok: [demoweb01] => {
    "usrplay.comment": "ansible playbook variable"
}
ok: [demoweb02] => {
    "usrplay.comment": "ansible playbook variable"
}
ok: [demodb01] => {
    "usrplay.comment": "ansible playbook variable"
}

PLAY RECAP ****
demodb01          : ok=3    changed=0    unreachable=0    failed=0    skipped=0    rescued=0
    ignored=0
demoweb01         : ok=3    changed=0    unreachable=0    failed=0    skipped=0    rescued=0
    ignored=0
demoweb02         : ok=3    changed=0    unreachable=0    failed=0    skipped=0    rescued=0
    ignored=0

ubuntu@ip-172-31-12-170:~/demodirectory/exc8$
```

20. Now open your playbook and comment out the variables in it. Then run the ansible playbook command again you will see that it has picked the variables from group vars.

```

vim demo_prec.yaml
ansible-playbook -i inventory demo_prec.yaml
```

```
ubuntu@ip-172-31-12-170:~/demodirectory/exc8$ vim demo_prec.yaml
ubuntu@ip-172-31-12-170:~/demodirectory/exc8$ ansible-playbook -i inventory demo_prec.yaml

PLAY [Understanding vars] ****
TASK [create user] ****
changed: [demodb01]
changed: [demoweb01]
changed: [demoweb02]

TASK [debug] ****
ok: [demoweb01] => {
    "usrplay.name": "awsuser"
}
ok: [demoweb02] => {
    "usrplay.name": "awsuser"
}
ok: [demodb01] => {
    "usrplay.name": "awsuser"
}

TASK [debug] ****
ok: [demoweb01] => {
    "usrplay.comment": "varialbe from aws console"
}
ok: [demoweb02] => {
    "usrplay.comment": "varialbe from aws console"
}
ok: [demodb01] => {
    "usrplay.comment": "varialbe from aws console"
}

PLAY RECAP ****
demodb01           : ok=3    changed=1    unreachable=0    failed=0    skipped=0    rescued=0
                     ignored=0
demoweb01          : ok=3    changed=1    unreachable=0    failed=0    skipped=0    rescued=0
                     ignored=0
demoweb02          : ok=3    changed=1    unreachable=0    failed=0    skipped=0    rescued=0
                     ignored=0
```