Lab Work Task. Day 2

Task

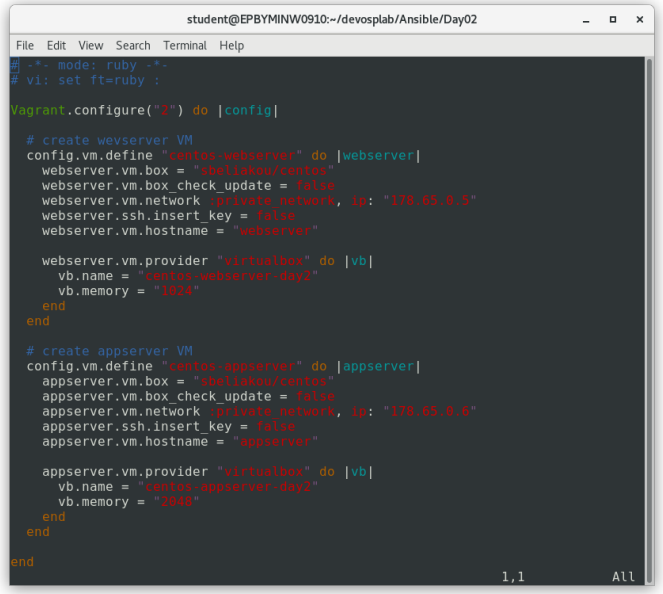
A) On Control Machine:

Create folder ~/ansible/day2. All working files should be placed there (Vagrantfile, playbooks, roles folder with custom roles, inventory)

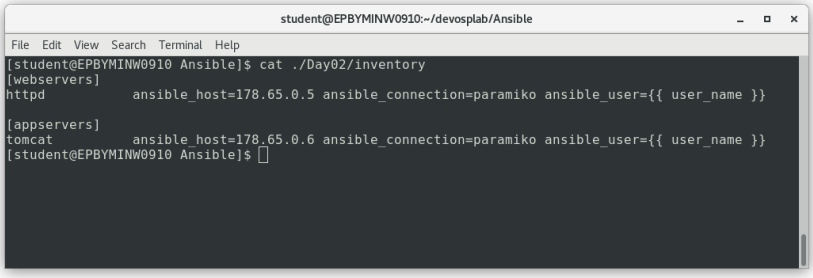
Picture 1

B) Spin up 2 VMs form single Vagranfile. Compose Inventory:

- Specify groups;
- Define group vars;
- Define host vars;
- Define ansible connection settings (common for all hosts/groups in inventory).



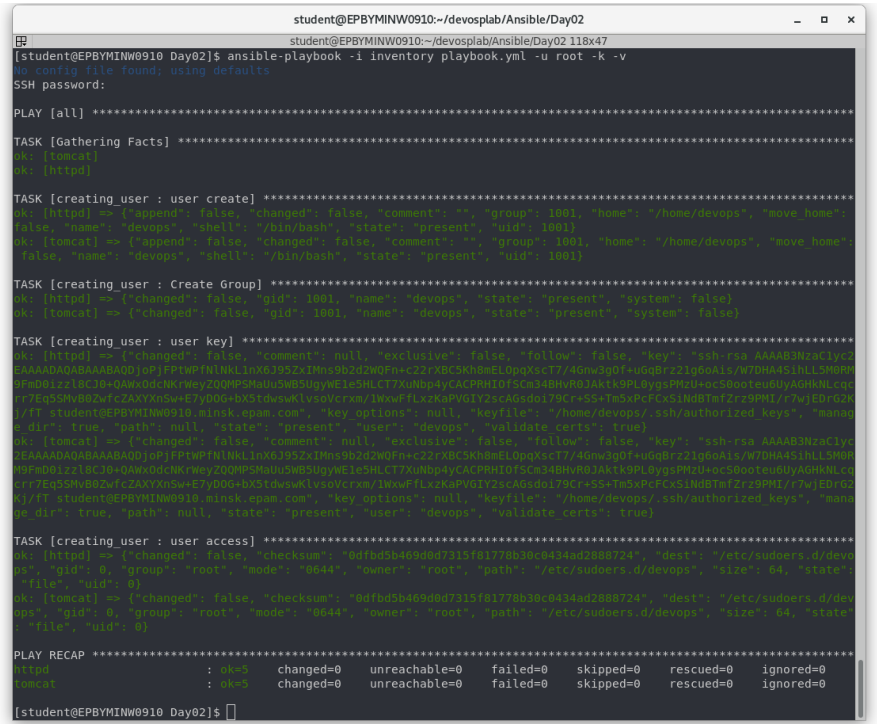
Picture 2 – Vagrantfile



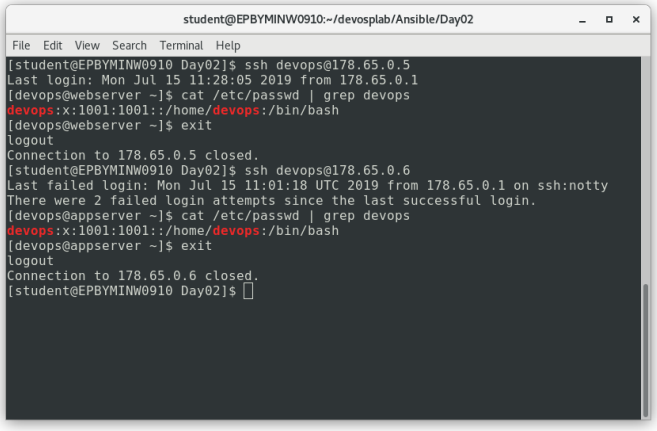
Picture 2.1 – cat inventory file

C) Develop following Roles:

- common (provides common system updates/configuration/so on);
- devops_user (creates “devops:devops” user, updates authorized keys file, configures sudoers, disables requiretty setting for this user);

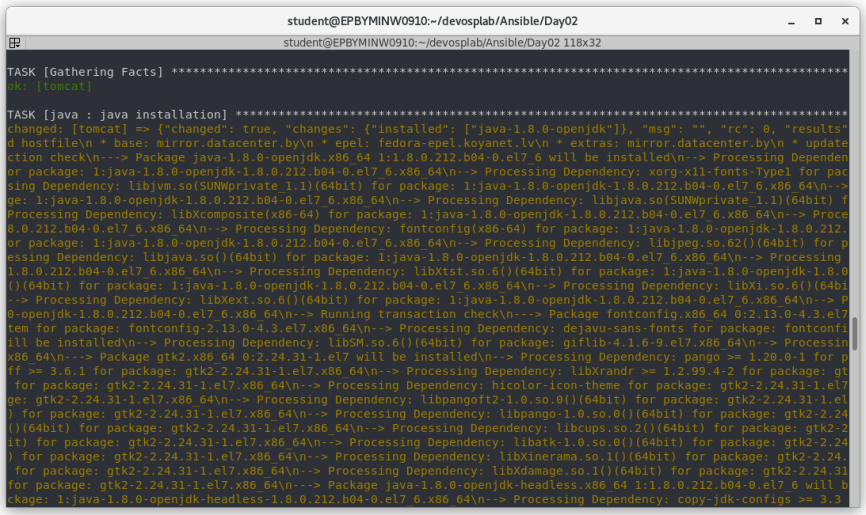

student@EPBYMINW0910:~/devosplab/Ansible/Day02
student@EPBYMINW0910:~/devosplab/Ansible/Day02 118x47
[student@EPBYMINW0910 Day02]$ ansible-playbook -i inventory playbook.yml -u root -k -v
No config file found; using defaults
SSH password:
PLAY [all] ***
TASK [Gathering Facts] ***
ok: [tomcat]
ok: [httpd]
TASK [creating_user : user create] ***
ok: [httpd] => {"append": false, "changed": false, "comment": "", "group": 1001, "home": "/home/devops", "move_home": false, "name": "devops", "shell": "/bin/bash", "state": "present", "uid": 1001}
ok: [tomcat] => {"append": false, "changed": false, "comment": "", "group": 1001, "home": "/home/devops", "move_home": false, "name": "devops", "shell": "/bin/bash", "state": "present", "uid": 1001}
TASK [creating_user : Create Group] ***
ok: [httpd] => {"changed": false, "gid": 1001, "name": "devops", "state": "present", "system": false}
ok: [tomcat] => {"changed": false, "gid": 1001, "name": "devops", "state": "present", "system": false}
TASK [creating_user : user key] ***
ok: [httpd] => {"changed": false, "comment": null, "exclusive": false, "follow": false, "key": "ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAABAQDjoPjFPtWPfNlNkL1nX6J95ZxIMns9b2d2WQFn+c22rXBC5Kh8mELOpqXscT7/4Gnw3gOf+uGqBrz21g6oAis/W7DHA4SihLL5M0RM9FmD0izzl8CJ0+QAWxOdcNKrWeyZOQMP5MaUu5WB5UgyWE1e5HLCT7XuNbp4yCACPRHIOfSCm34BHvR0JAktk9PL0ygsPMzU+ocS0ooteu6UyAGHkNLcqcrr7Eq5SMvB0ZwfcZAXYXn5W+E7yDOG+bX5tdwswKlvsoVcrxm/1WxwFfLxzKaPVGIY2scAGsdoi79Cr+S5+Tm5xPcFCxSiNdBTmfZrz9PMI/r7wjEDrG2Kj/fT student@EPBYMINW0910.minsk.epam.com", "key_options": null, "keyfile": "/home/devops/.ssh/authorized_keys", "manage_dir": true, "path": null, "state": "present", "user": "devops", "validate_certs": true}
ok: [tomcat] => {"changed": false, "comment": null, "exclusive": false, "follow": false, "key": "ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAABAQDjoPjFPtWPfNlNkL1nX6J95ZxIMns9b2d2WQFn+c22rXBC5Kh8mELOpqXscT7/4Gnw3gOf+uGqBrz21g6oAis/W7DHA4SihLL5M0RM9FmD0izzl8CJ0+QAWxOdcNKrWeyZOQMP5MaUu5WB5UgyWE1e5HLCT7XuNbp4yCACPRHIOfSCm34BHvR0JAktk9PL0ygsPMzU+ocS0ooteu6UyAGHkNLcqcrr7Eq5SMvB0ZwfcZAXYXnSw+E7yDOG+bX5tdwswKlvsoVcrxm/1WxwFfLxzKaPVGIY2scAGsdoi79Cr+SS+Tm5xPcFCxSiNdBTmfZrz9PMI/r7wjEDrG2Kj/fT student@EPBYMINW0910.minsk.epam.com", "key_options": null, "keyfile": "/home/devops/.ssh/authorized_keys", "manage_dir": true, "path": null, "state": "present", "user": "devops", "validate_certs": true}
TASK [creating_user : user access] ***
ok: [httpd] => {"changed": false, "checksum": "0dfbd5b469d0d7315f81778b30c0434ad2888724", "dest": "/etc/sudoers.d/devops", "gid": 0, "group": "root", "mode": "0644", "owner": "root", "path": "/etc/sudoers.d/devops", "size": 64, "state": "file", "uid": 0}
ok: [tomcat] => {"changed": false, "checksum": "0dfbd5b469d0d7315f81778b30c0434ad2888724", "dest": "/etc/sudoers.d/devops", "gid": 0, "group": "root", "mode": "0644", "owner": "root", "path": "/etc/sudoers.d/devops", "size": 64, "state": "file", "uid": 0}
PLAY RECAP ***
httpd : ok=5 changed=0 unreachable=0 failed=0 skipped=0 rescued=0 ignored=0
tomcat : ok=5 changed=0 unreachable=0 failed=0 skipped=0 rescued=0 ignored=0
[student@EPBYMINW0910 Day02]$

Picture 3.1

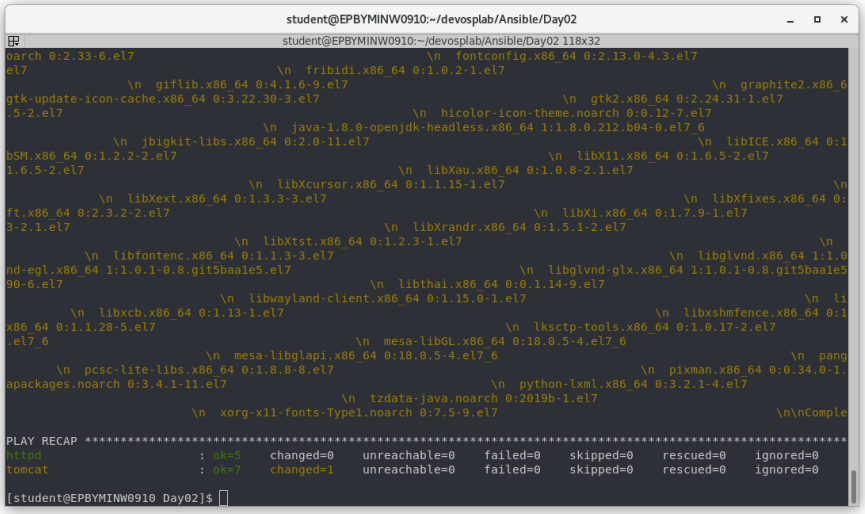


Picture 3.2 – Proof

– java (installs java, verifies java installation; supports java version customization);



Picture 3.3

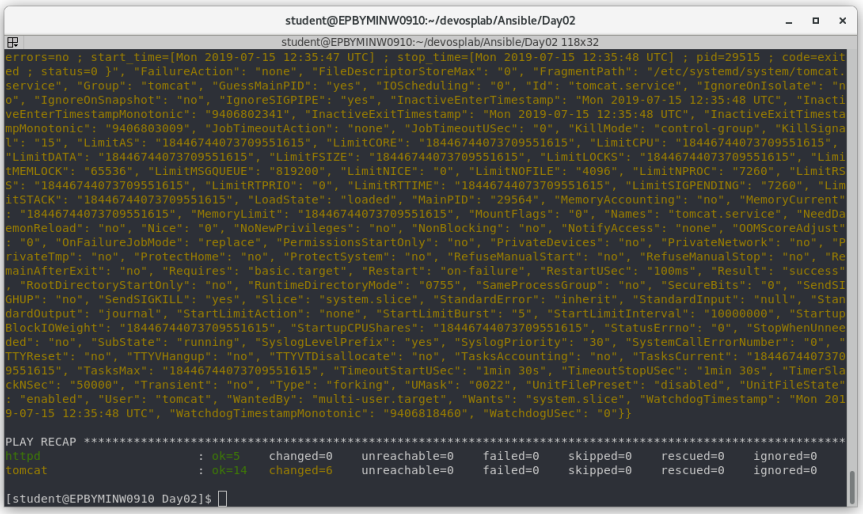


Picture 3.4

Picture 3.5 – Check Java version on appserver

– tomcat (installs tomcat from sources - war, creates tomcat user with custom uid, creates tomcat user's group with custom gid, configures systemd to start tomcat automatically, verifies tomcat installation; supports tomcat customization: JAVA_OPTS, version, etc; role has dependency on java role);

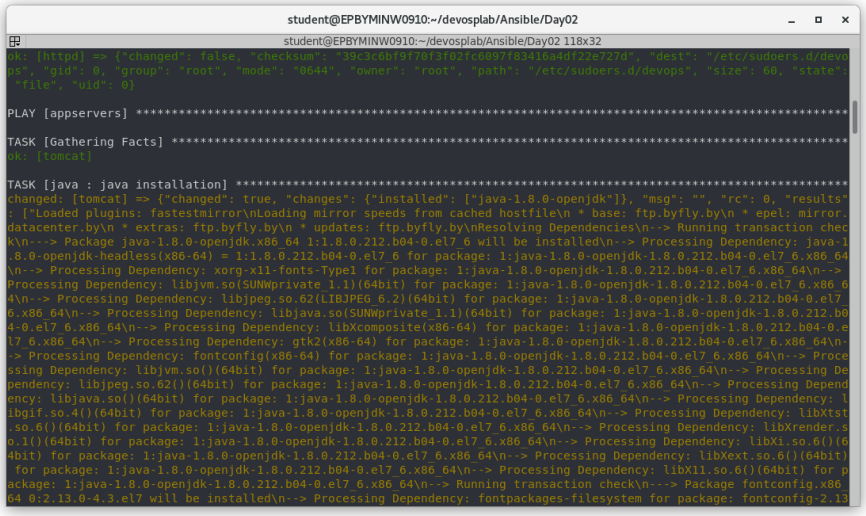
Picture 3.6



Picture 3.7

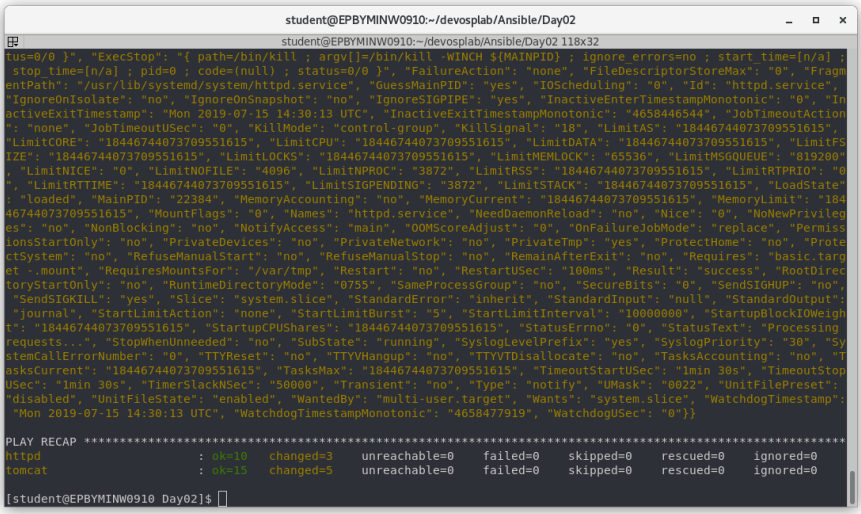
Add JAVA to /role/tomcat/meta/main.yml

```
dependencies:
- role: java
  java_version: 1.8.0
```

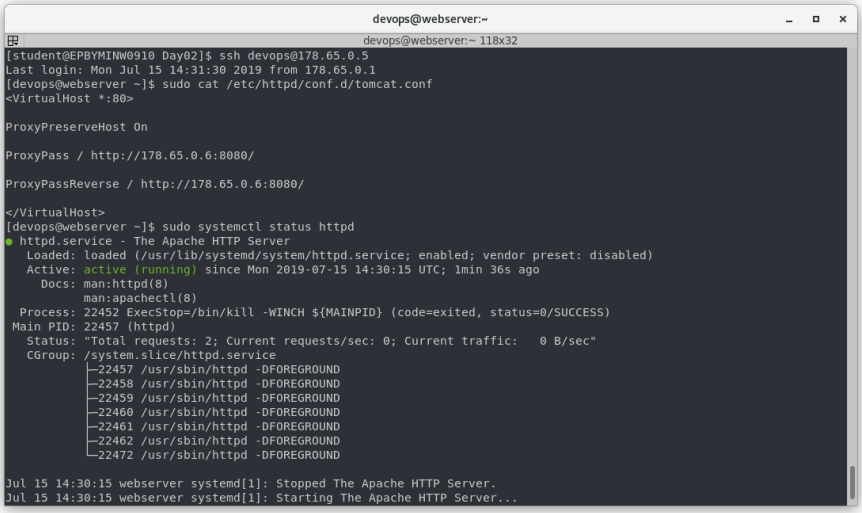


Picture 3.8

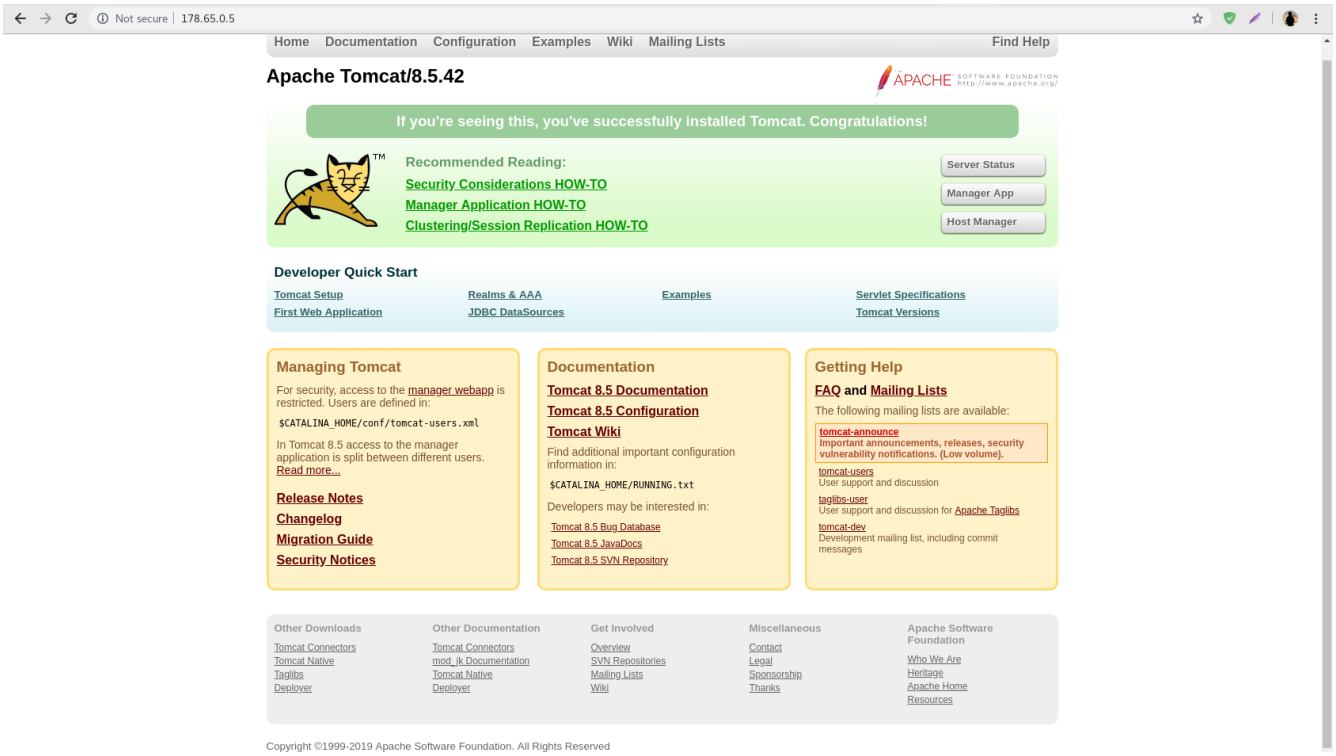
– httpd (installs apache httpd service, forwards requests to tomcat through custom VirtualHost, verifies http installation and running state).



Picture 3.8

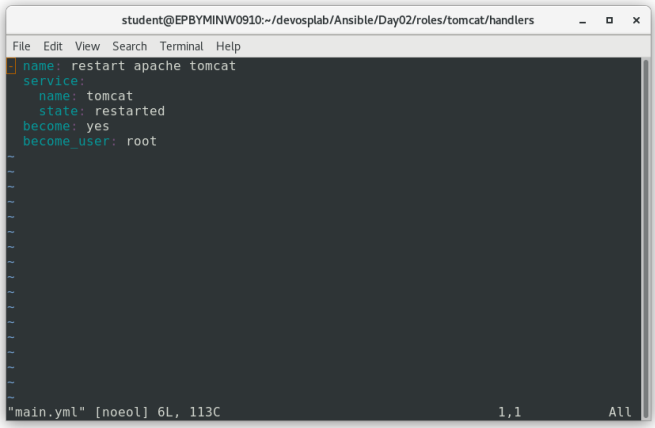


Picture 3.8 – Check proxy settings and service stratus

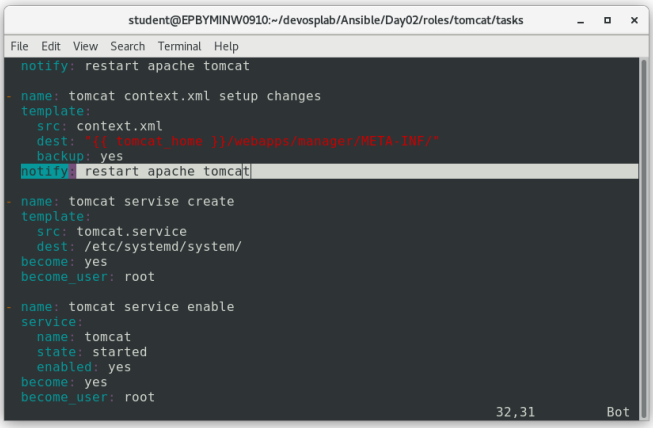


Picture 3.9 – Check results

D) Make sure that all necessary application state changes are being done with handlers on demand.



Picture 4.1



Picture 4.2

And httpd handlers you may see in /role/httpd/handlers/main.yml

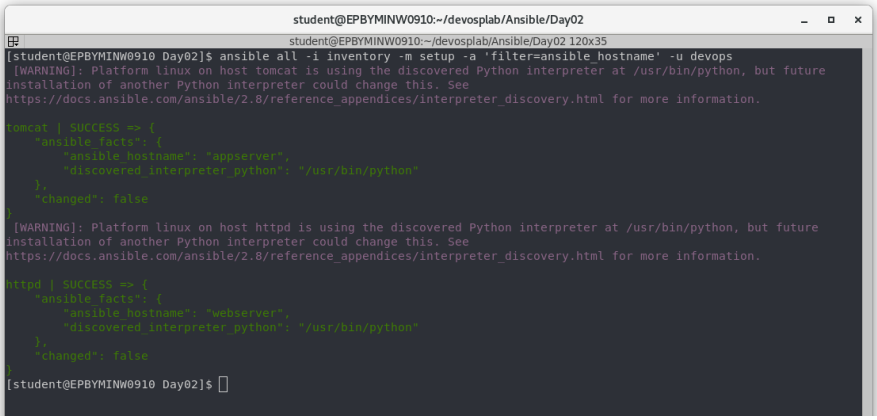
Picture 4.3 – Restart service after change proxy configs

E) All roles should leave facts about configured/provisioned software on the system (Verify system's configuration with ad-hoc command – setup module - after provisioning)

```
$ ansible all -i inventory -m setup -a 'filter=ansible_all_ipv4_addresses' -u devops
```

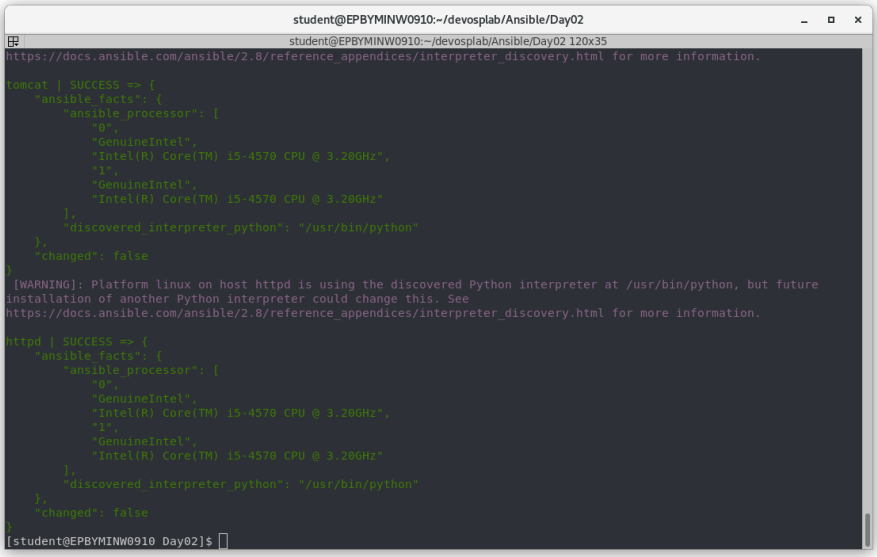
Picture 5.1 – IPv4 addresses

```
$ ansible all -i inventory -m setup -a 'filter=ansible_hostname' -u devops
```



Picture 5.2 – Hostname

```
$ ansible all -i inventory -m setup -a 'filter=ansible_processor' -u devops
```



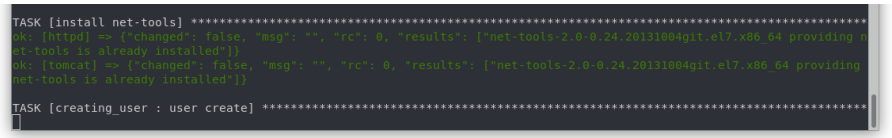
Picture 5.3 – Processor

```
$ ansible all -i inventory -m setup -a 'filter=ansible_processor_cores' -u devops
```

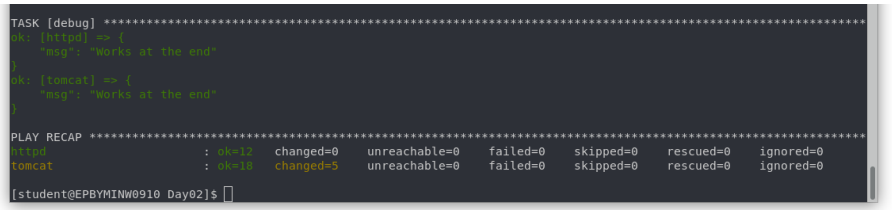
Picture 5.4 – Number processor cores

F) Develop provision.yml playbook:

- Sections pre_tasks, tasks, roles, post_tasks;

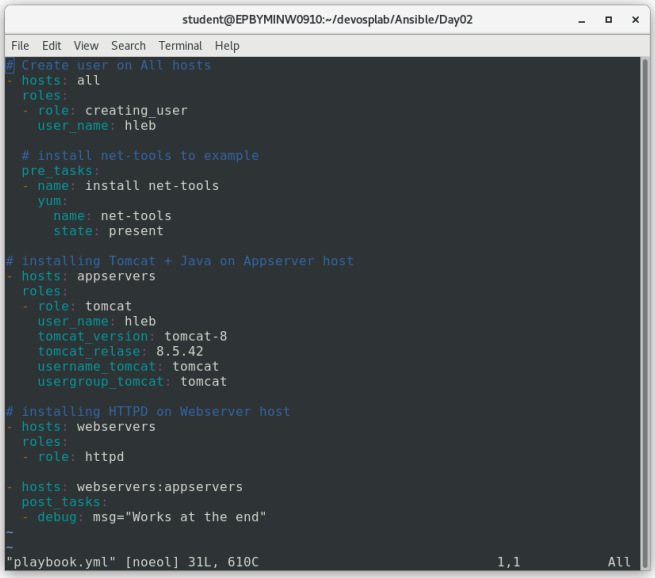


Picture 6.1 – Pre_tasks install net-tools



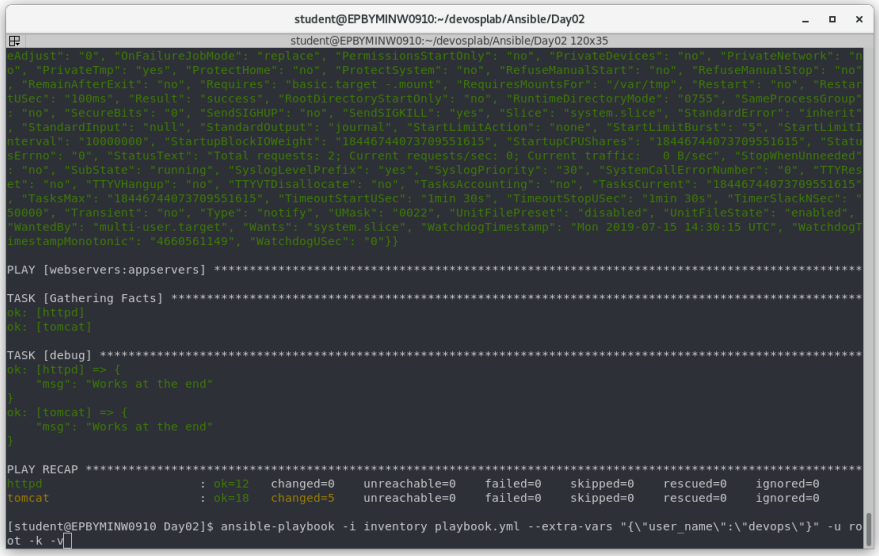
Picture 6.1 – Post_tasks debag info

- Specify tomcat version in playbook, inventory (group vars, host vars), extra vars – try different.

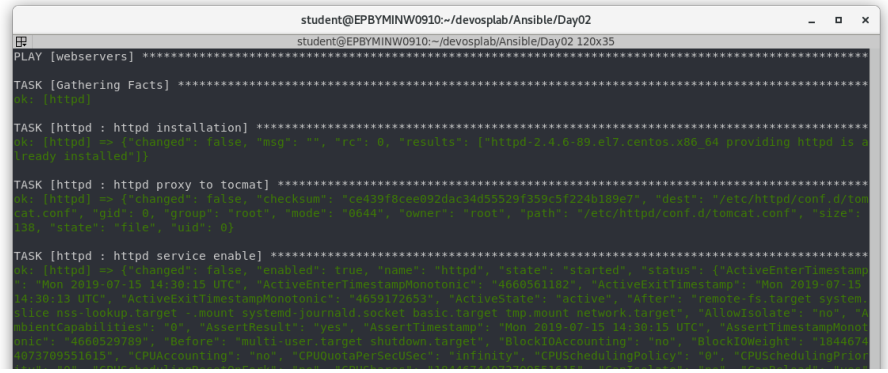


Picture 7.1 – Filed “user_name” is hleb

```
$ ansible-playbook -i inventory playbook.yml --extra-vars
"{\"user_name\":\"devops\"}" -u root -k -v
```



Picture 7.2 – Successful trying extra vars



Picture 7.3 – User not changed

This is useful for, among other things, setting the hosts group or the user for the playbook. Difference only in the precedence.

Here is the order of precedence from least to greatest (the last listed variables winning prioritization):

1. command line values (eg “-u user”)
2. role defaults
3. inventory file or script group vars
4. inventory group_vars/all
5. playbook group_vars/all
6. inventory group_vars/*
7. playbook group_vars/*
8. inventory file or script host vars
9. inventory host_vars/*
10. playbook host_vars/*
11. host facts / cached set_facts
12. play vars
13. play vars_prompt
14. play vars_files
15. role vars (defined in role/vars/main.yml)
16. block vars (only for tasks in block)
17. task vars (only for the task)
18. include_vars
19. set_facts / registered vars
20. role (and include_role) params
21. include params
22. extra vars (always win precedence)