Lab Work Task. Day 3.
Making Roles and playbooks better.
Custom Filters, Custom Modules

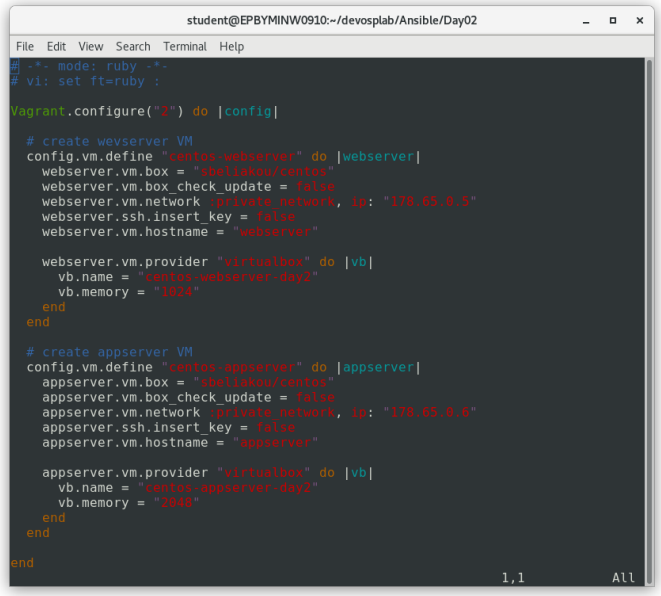
Task

A) On Control Machine:

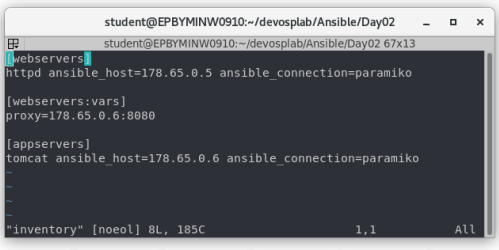
Create folder ~/ansible/day3. All working files should be placed there (Vagrantfile, playbooks, roles folder with custom roles, library – for modules and filter_plugins for filters folders, inventory)

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)



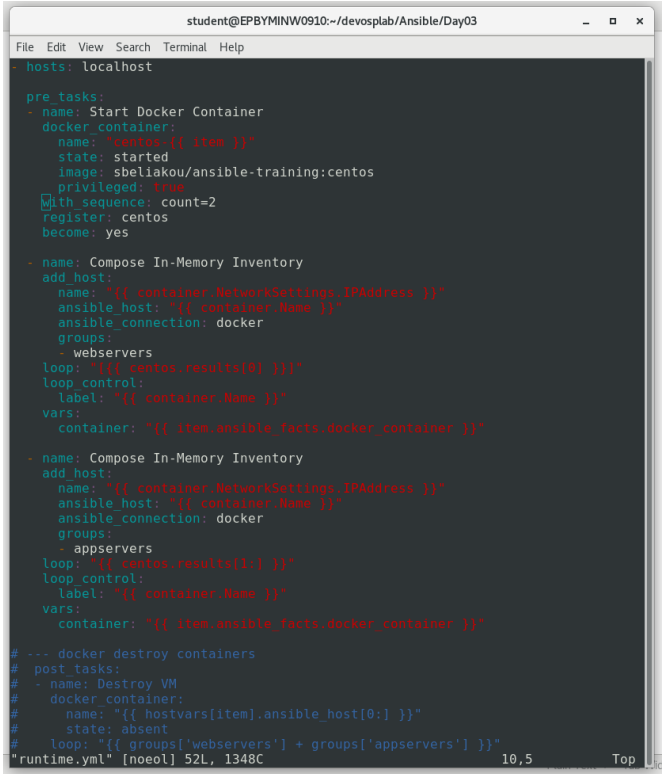
Picture 1.1 – Vagrant file



Picture 1.2 - Invantory

OR

Use example from repo to spin up environment with ansible and vagrantfile, register host in run-time inventory with all necessary parameters.



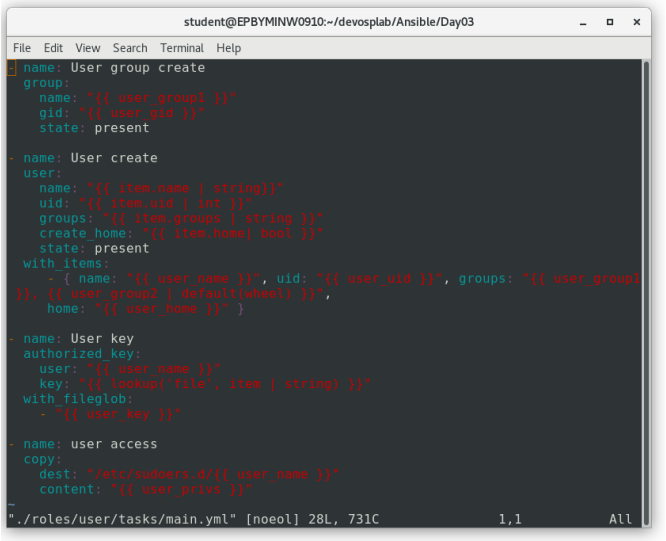
Picture 1.3 – Solution with docker

Picture 1.4 – Docker containers

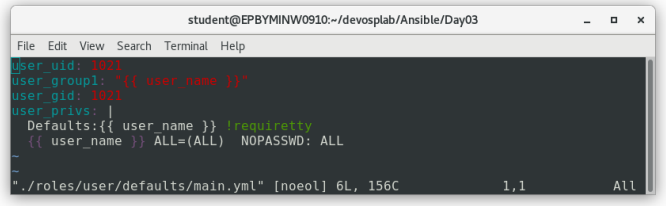
C) Develop role “user”, make sure it accepts following parameters:

- name (str) – user name to be created
- group (str) – user’s primary group
- groups (list) – user’s secondary groups
- uid (number) – user id
- gid (number) – group id
- home (str) – user’s home folder location, if not set – rely on system settings
- create_home (bool) – whether create home for user or no
- authorized_keys (list) - public ssh keys to be updated on the system / current user
- sudoers (str) – user’s sudoers priviliges

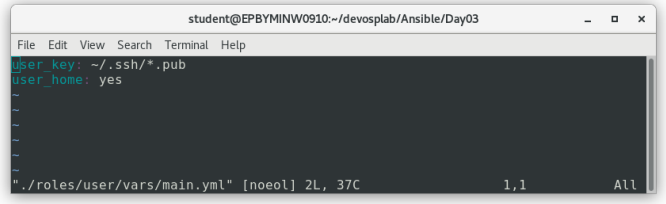
Develop and test all possible cases (passing different parameters)



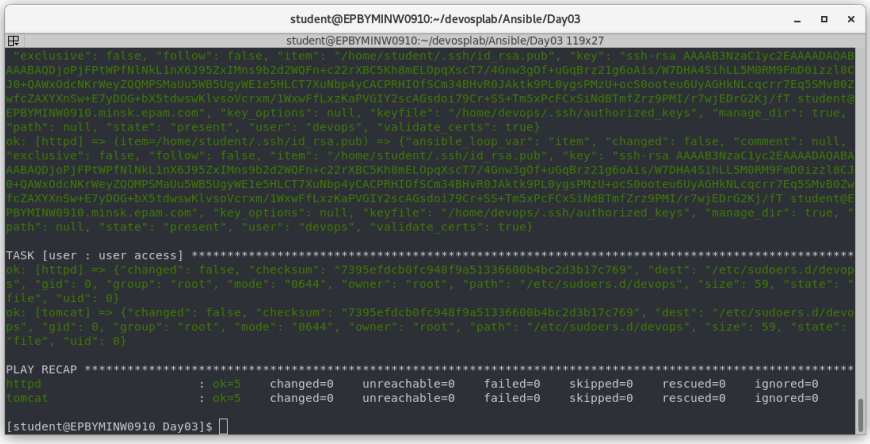
Picture 2.1 – User role ./roles/user/task/main.yml



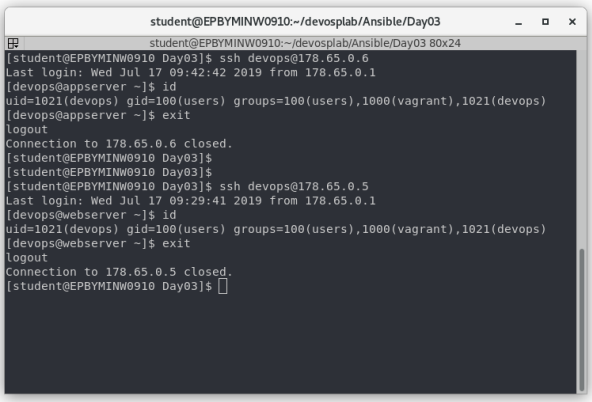
Picture 2.2 – User defaults ./roles/user/defaults/main.yml



Picture 2.3 – User vars ./roles/user/vars/main.yml



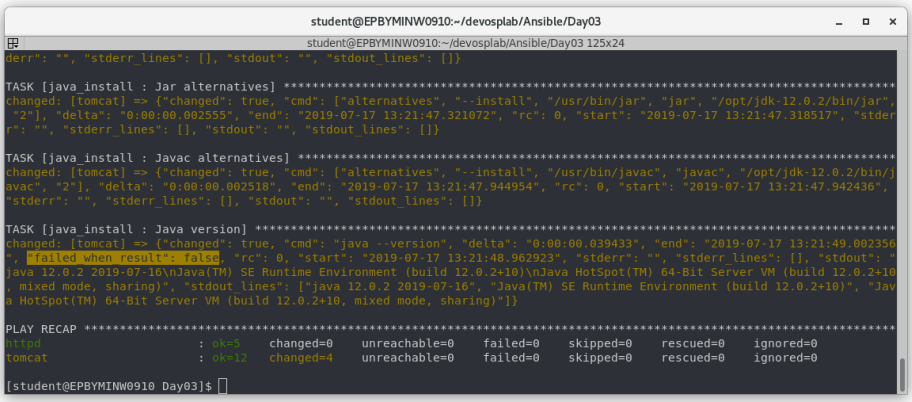
Picture 2.4 – Results playbook run



Picture 2.5 – Checks

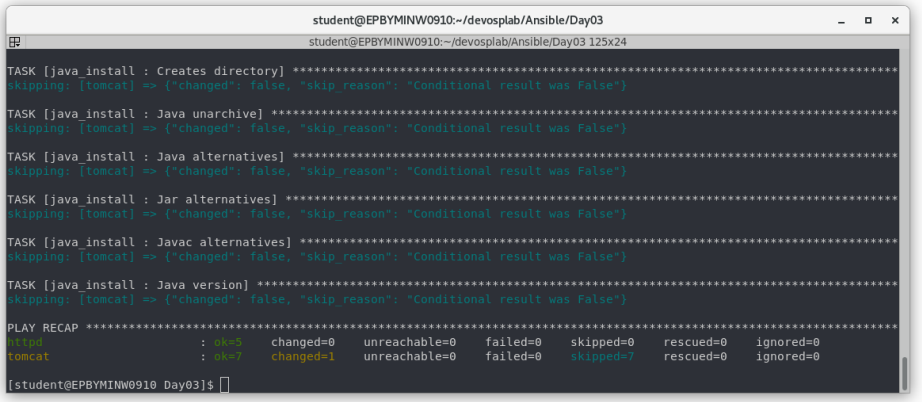
D) (Re-)Develop role “java”

- Role accepts parameter version (java version to be installed)
- Role takes appropriate distribution package (on Control Machine) for installation or reports/fails with custom message like “java version distribution hasn’t been found”
- Role creates directory on the system: /opt/oracle/java/{{ version }}/ (JAVA_HOME)
- Role registers java/javac/jar in alternatives
- Role validates that installed java is the same as it should be by configuration



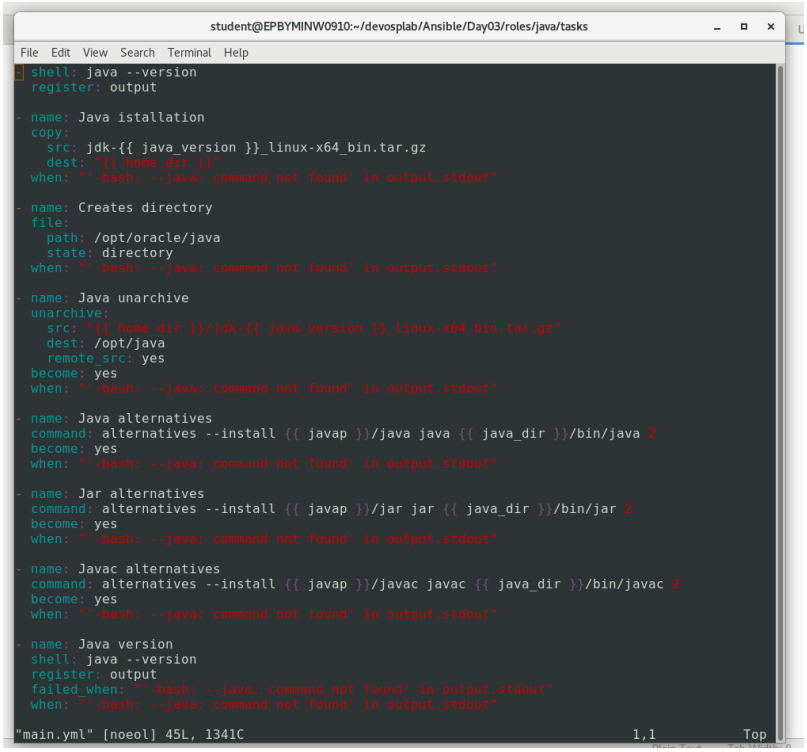
Picture 3.1 – Validates java

- Role skips installation steps if there's a fact with suitable java configuration



Picture 3.2 – Skip when java

Download several java distribution packages from Oracle site (*.tar.gz)

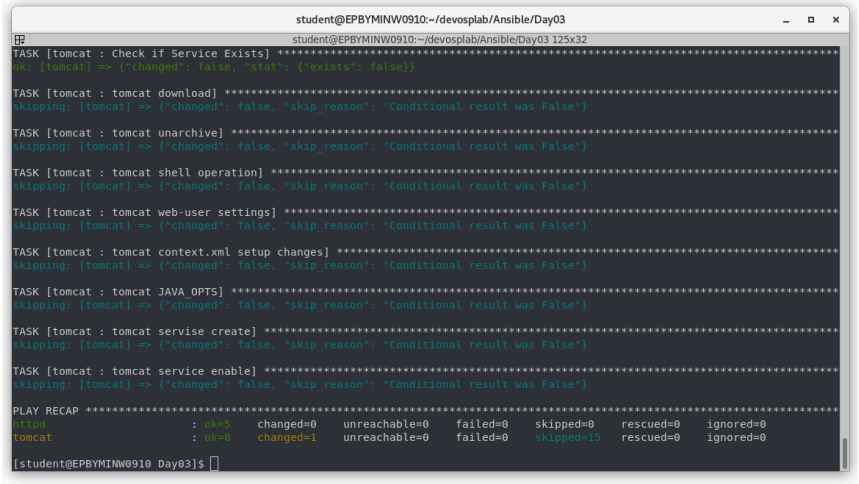


Picture 3.3 – /roles/java/tasks/main.yml

Picture 3.4 – /roles/java/defaults/main.yml

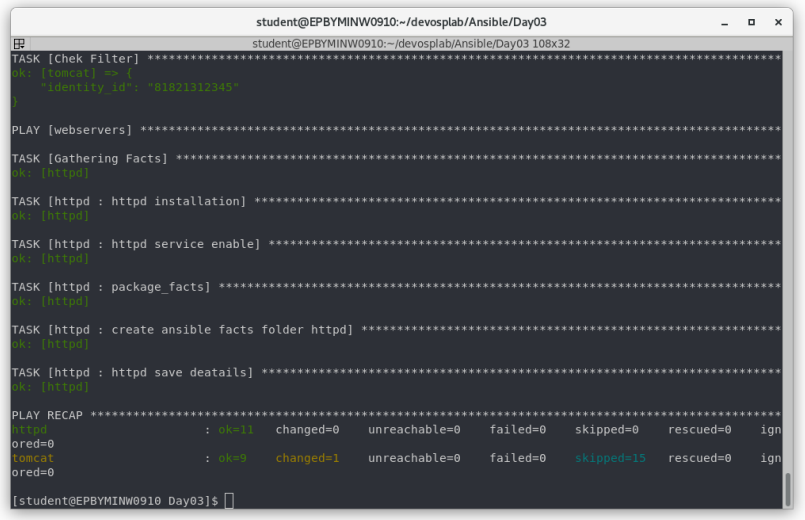
Picture 3.4 – /roles/java/vars/main.yml

E) Update Role “tomcat” from Homework 2 to skip installation/configuration if ansible fact reports that this configuration is already done

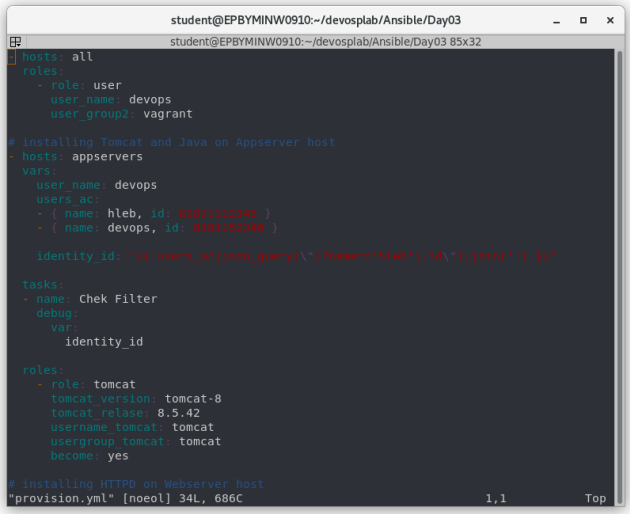


Picture 4 – Skipped installation if tomcat service stated

F) Develop custom filter which simplifies use case from slide #33 (to find identirti_id number more simpler way)



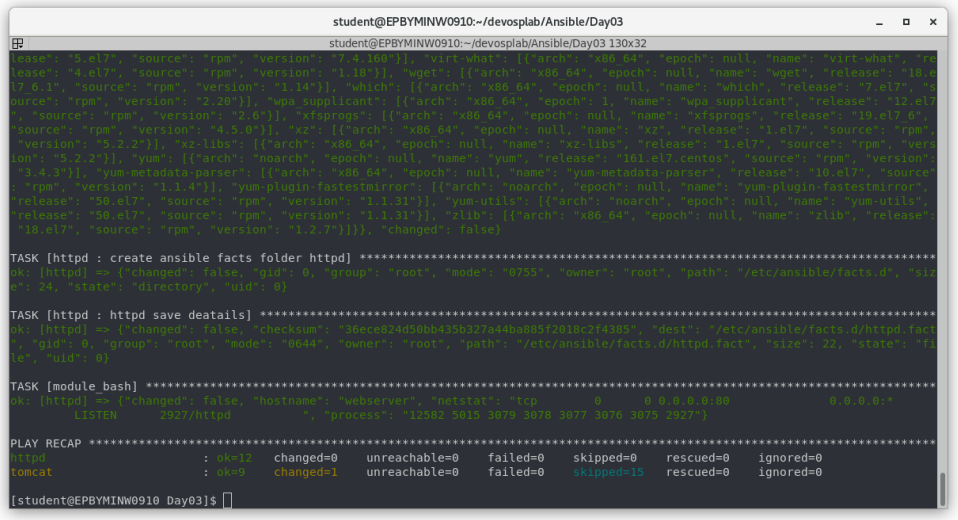
Picture 5.1 – Result



Picture 5.2 – Simpler way to use filter as variable

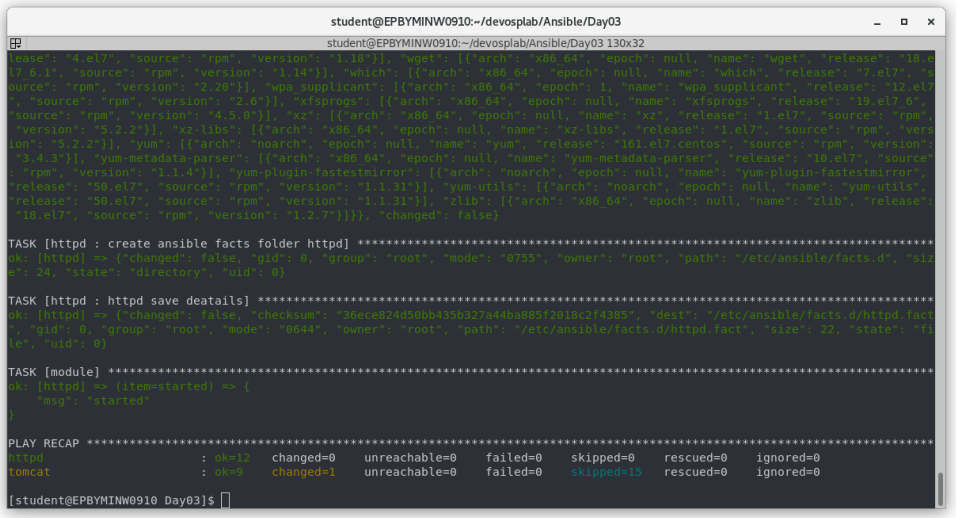
G) Create custom module on Bash or Python (up to you), which performs service's santy tests:

- Process is running as expected (by name, under user)
- Port is handled by proper process, in listening mode
- Web content of given url contains given regexp string
- Web server information (curl -IL url) contains given regexp strin



Picture 5.1 – Results

Use Loop Control to clean playbook output



Picture 5.2 – Results

Picture 5.3 – Playbook