Name: Denzel Dinglasan	Date Performed: 06/11/2023
Course/Section:CPE 232 - CPE31S6	Date Submitted: 06/11/2023
Instructor: Dr. Jonathan Vidal Taylar	Semester and SY: 1st Sem 2023-2024

Midterm Skills Exam: **Install, Configure, and Manage Log Monitoring tools**

1. Objectives

Create and design a workflow that installs, configure and manage enterprise availability, performance and log monitoring tools using Ansible as an Infrastructure as Code (IaC) tool.

2. Instructions

1. Create a repository in your GitHub account and label it CPE_MIDEXAM_SURNAME.
2. Clone the repository and do the following:
   2.1. Create an Ansible playbook that does the following with an input of a config.yaml file and arranged Inventory file:
   2.2. Install and configure Elastic Stack in separate hosts (Elastic Search, Kibana, Logstash) • Install Nagios in one host
   2.3. Install Grafana,Prometheus and Influxdb in seperate hosts (Influxdb,Grafana,Prometheus)
   2.4. Install Lamp Stack in separate hosts (Httpd + Php,Mariadb)
3. Document all your tasks using this document. Provide proofs of all the ansible playbooks codes and successful installations.
4. Document the push and commit from the local repository to GitHub.
5. Finally, paste also the link of your GitHub repository in the documentation.

3. Output (screenshots and explanations)

Create a new repository

A repository contains all project files, including the revision history. Already have a project repository elsewhere? Import a repository.

Required fields are marked with an asterisk (*).

Owner * **Repository name ***

ddinglasan / Ξ_MIDTERM_DINGLASAN

CPE_MIDTERM_DINGLASAN is available.

Great repository names are short and memorable. Need inspiration? How about **urban-barnacle** ?

Description (optional)

Public
Anyone on the internet can see this repository. You choose who can commit.

Private
You choose who can see and commit to this repository.

Initialize this repository with:

Add a README file
This is where you can write a long description for your project. Learn more about READMEs.

Add .gitignore

.gitignore template: None

Choose which files not to track from a list of templates. Learn more about ignoring files.

Choose a license

License: None

A license tells others what they can and can't do with your code. Learn more about licenses.

This will set `main` as the default branch. Change the default name in your settings.

You are creating a public repository in your personal account.

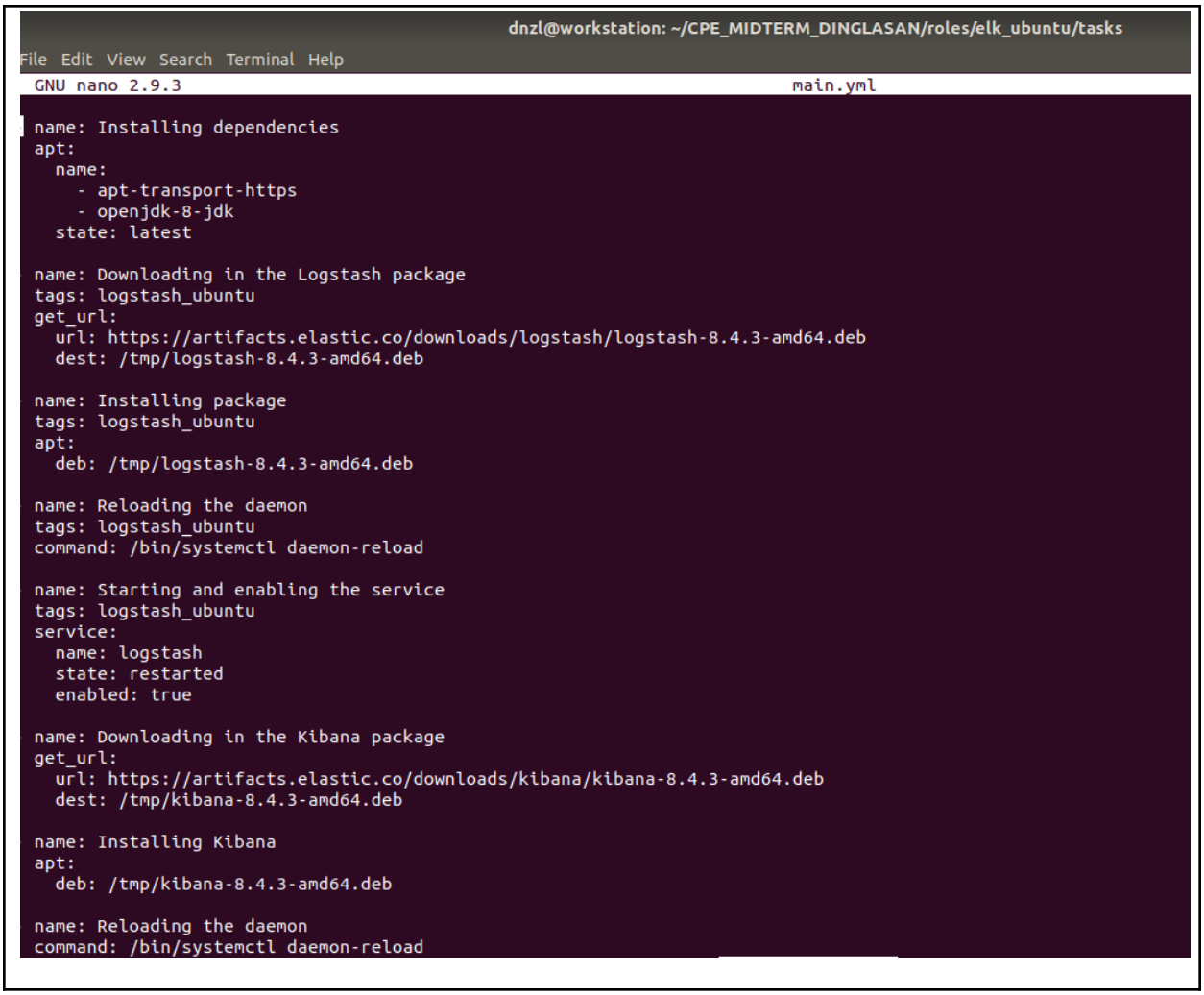
Create repository

```
dnzl@workstation:~$ git clone https://github.com/ddinglasan/CPE_MIDTERM_DINGLASAN.git
Cloning into 'CPE_MIDTERM_DINGLASAN'...
remote: Enumerating objects: 3, done.
remote: Counting objects: 100% (3/3), done.
remote: Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
Unpacking objects: 100% (3/3), done.
```

Step 2: Create the basic files needed(ansible.cfg & inventory) and create the roles needed for the Ubuntu and CentOS computer with the main.yml file for their own tasks. Also created a task.yml file to run the tasks of the roles.

```
dnzl@workstation:~/CPE_MIDTERM_DINGLASAN$ tree
.
├── ansible.cfg
├── files
│   ├── grafana.repo
│   ├── influxdb.repo
│   ├── kibana.repo
│   ├── logstash.repo
│   └── prometheus.service
├── inventory
├── README.md
├── roles
│   ├── elk_centos
│   │   └── tasks
│   │       └── main.yml
│   ├── elk_ubuntu
│   │   └── tasks
│   │       └── main.yml
│   ├── ipg_centos
│   │   └── tasks
│   │       └── main.yml
│   ├── ipg_ubuntu
│   │   └── tasks
│   │       └── main.yml
│   ├── ls_centos
│   │   └── tasks
│   │       └── main.yml
│   ├── ls_ubuntu
│   │   └── tasks
│   │       └── main.yml
│   └── nagios_ubuntu
│       └── tasks
│           └── main.yml
└── task.yml
```

Step 3: Paste this on the main.yml of the elk_ubuntu role.


dnzl@workstation: ~/CPE_MIDTERM_DINGLASAN/roles/elk_ubuntu/tasks
File Edit View Search Terminal Help
GNU nano 2.9.3 main.yml
name: Installing dependencies
apt:
name:
- apt-transport-https
- openjdk-8-jdk
state: latest
name: Downloading in the Logstash package
tags: logstash_ubuntu
get_url:
url: https://artifacts.elastic.co/downloads/logstash/logstash-8.4.3-amd64.deb
dest: /tmp/logstash-8.4.3-amd64.deb
name: Installing package
tags: logstash_ubuntu
apt:
deb: /tmp/logstash-8.4.3-amd64.deb
name: Reloading the daemon
tags: logstash_ubuntu
command: /bin/systemctl daemon-reload
name: Starting and enabling the service
tags: logstash_ubuntu
service:
name: logstash
state: restarted
enabled: true
name: Downloading in the Kibana package
get_url:
url: https://artifacts.elastic.co/downloads/kibana/kibana-8.4.3-amd64.deb
dest: /tmp/kibana-8.4.3-amd64.deb
name: Installing Kibana
apt:
deb: /tmp/kibana-8.4.3-amd64.deb
name: Reloading the daemon
command: /bin/systemctl daemon-reload

```
- name: Starting and enabling the service
  tags: logstash_ubuntu
  service:
    name: logstash
    state: restarted
    enabled: true

- name: Downloading in the Kibana package
  get_url:
    url: https://artifacts.elastic.co/downloads/kibana/kibana-8.4.3-amd64.deb
    dest: /tmp/kibana-8.4.3-amd64.deb

- name: Installing Kibana
  apt:
    deb: /tmp/kibana-8.4.3-amd64.deb

- name: Reloading the daemon
  command: /bin/systemctl daemon-reload

- name: Making sure that Kibana service is started and enabled
  service:
    name: kibana
    state: restarted
    enabled: true

- name: Downloading in the elastic search package
  get_url:
    url: https://artifacts.elastic.co/downloads/elasticsearch/elasticsearch-8.4.3-amd64.deb
    dest: /tmp/elasticsearch-8.4.3-amd64.deb

- name: Installing package
  apt:
    deb: /tmp/elasticsearch-8.4.3-amd64.deb

- name: Enabling elastic search service
  tags: es_ubuntu
  service:
    name: elasticsearch
    enabled: yes

- name: Modifying service file
  tags: es_ubuntu
  replace:
    path: /usr/lib/systemd/system/elasticsearch.service
    regexp: "TimeoutStartSec=75"
    replace: "TimeoutStartSec=500"

- name: Starting and enabling the deamon
  shell: |
    sudo systemctl enable elasticsearch.service
    sleep 10
    sudo systemctl start elasticsearch.service
  ignore_errors: yes
```

Step 4: Paste this on the main.yml of the elk_centos role.

dnzl@workstation: ~/CPE_MIDTERM_DINGLASAN/roles/elk_centos/tasks

File Edit View Search Terminal Help

GNU nano 2.9.3 main.yml

```
- name: Downloading the source file of Elasticsearch
  tags: es_ubuntu
  get_url:
    url: https://artifacts.elastic.co/downloads/elasticsearch/elasticsearch-8.4.3-x86_64.rpm
    dest: /tmp/elasticsearch-8.4.3-x86_64.rpm

- name: Installing Elasticsearch
  tags: es_ubuntu
  yum:
    name: /tmp/elasticsearch-8.4.3-x86_64.rpm
    state: present

- name: Enabling Elasticsearch service
  tags: es_ubuntu
  service:
    name: elasticsearch
    enabled: yes

- name: Modifying service file
  tags: es_ubuntu
  replace:
    path: /usr/lib/systemd/system/elasticsearch.service
    regexp: "TimeoutStartSec=75"
    replace: "TimeoutStartSec=300"

- name: Opening port for elastic search
  tags: es_ubuntu
  shell: |
    sudo firewall-cmd --permanent --zone=public --add-port=9200/tcp
    sleep 10
    sudo firewall-cmd --reload

- name: Enabling elastic search service
  tags: es_ubuntu
  shell: |
    systemctl enable elasticsearch.service
    sleep 10
    systemctl start elasticsearch.service
  ignore_errors: yes
```

```
- name: Downloading and installing public signing key
  tags: logstash_ubuntu
  rpm_key:
    state: present
    key: https://artifacts.elastic.co/GPG-KEY-elasticsearch

- name: Creeating a repo file for Logstash
  tags: logstash_ubuntu
  copy:
    src: logstash.repo
    dest: /etc/yum.repos.d/logstash.repo
    owner: root
    group: root
    mode: 0777

- name: Updating repo
  tags: logstash_ubuntu
  dnf:
    update_cache: yes
```

```
- name: Installing Logstash and its dependencies
  tags: logstash_ubuntu
  dnf:
    name:
      - logstash
    state: latest

- name: Opening port for Logstash
  tags: logstash_ubuntul, elk_install
  shell: |
    sudo firewall-cmd --permanent --zone=public --add-port=9600/tcp
    sleep 10
    sudo firewall-cmd --reload

- name: Making sure that logstash is stared and enabled
  tags: logstash_ubuntu, service, logstash_service, elk_service
  service:
    name: logstash
    state: restarted
```

```
    enabled: true

- name: Downloading and installing public signing key
  tags: kibana_ubuntu, kibana_install, elk_install
  rpm_key:
    state: present
    key: https://artifacts.elastic.co/GPG-KEY-elasticsearch

- name: Adding Kibana to the RPM repository
  tags: kibana_ubuntu, kibana_install, elk_install
  copy:
    src: kibana.repo
    dest: /etc/yum.repos.d/kibana.repo
    owner: root
    group: root
    mode: 777

- name: Updating the repository once again
  tags: kibana_ubuntu, kibana_install, elk_install
```

```
  yum:
    name:
      - kibana
    state: latest

- name: Opening port for Kibana
  tags: kibana_ubuntu, kibana_installl, elk_install
  firewalld:
    port: 5601/tcp
    zone: public
    permanent: yes
    state: enabled

- name: Making sure that Kibana is started and enabled
  tags: kibana_ubuntu, elk_service, kibana_service, service
  service:
    name: kibana
    state: restarted
    enabled: true
```

Step 3: Paste this on the main.yml of the lpg_ubuntu role.

dnzl@workstation: ~/CPE_MIDTERM_DINGLASAN/roles/ipg_ubuntu/tasks

File Edit View Search Terminal Help

GNU nano 2.9.3 main.yml

```
- name: Installing dependencies
  apt:
    name:
      - apt-transport-https
      - software-properties-common
      - wget
    state: latest

- name: Adding Influxdb in the repository
  shell: |
    wget -q https://repos.influxdata.com/influxdb.key
    sleep 5
    echo '23a1c8836f0afc5ed24e0486339d7cc8f6790b83886c4c96995b88a061c5bb5d influxdb.key' | sha256sum -c && cat influxdb.key | gpg --dearmor | sudo tee$
    sleep 5
    echo 'deb [signed-by=/etc/apt/trusted.gpg.d/influxdb.gpg] https://repos.influxdata.com/debian stable main' | sudo tee /etc/apt/sources.list.d/infl$

- name: Installing Influxdb
  apt:
    name:
      - influxdb

- name: Making sure that the Influxd is enabled and started
  service:
    name: influxdb
    state: started
    enabled: true

- name: Adding Grafana Repo
  shell: |
    sudo wget -q -O /usr/share/keyrings/grafana.key https://packages.grafana.com/gpg.key

- name: Update repo
  shell: |
    sudo apt-get update

- name: Updating the repo and isntalling grafana
  apt:
    name:
      - grafana
```

```
- name: Reloading the daemon
  shell: |
    sudo systemctl daemon-reload

- name: Making sure that the Grafana server is started and enabled
  service:
    name: grafana-server
    state: restarted
    enabled: true

- name: Creating a directory (where the downloaded files will be stored)
  tags: directory
  file:
    path: ~/prometheus
    state: directory

- name: Downloading and extracting Prometheus
  tags: source
  unarchive:
    src: https://github.com/prometheus/prometheus/releases/download/v2.39.1/prometheus-2.39.1.linux-amd64.tar.gz
    dest: ~/prometheus
    remote_src: yes
    mode: 0777
    owner: root
    group: root

- name: Stopping the Prometheus service if its exist
  shell: |
    sudo systemctl stop prometheus >> /dev/null
  ignore_errors: yes

- name: Adding the Prometheus executables to a PATH
  tags: executables
  shell: |
    cd ~/prometheus/prometheus*
    cp -r . /usr/local/bin/prometheus

- name: Copying the Prometheus service file
  tags: servicefile
```

```
  copy:
    src: prometheus.service
    dest: /etc/systemd/system/
    owner: root
    group: root
    mode: 777

- name: Making sure that Prometheus service is started and enabled
  tags: serviceon
  service:
    name: prometheus
    state: started
    enabled: true
```

Step 4: Paste this on the main.yml of the ipg_centos role.

dnzl@workstation: ~/CPE_MIDTERM_DINGLASAN/roles/ipg_centos/tasks

File Edit View Search Terminal Help

GNU nano 2.9.3 main.yml

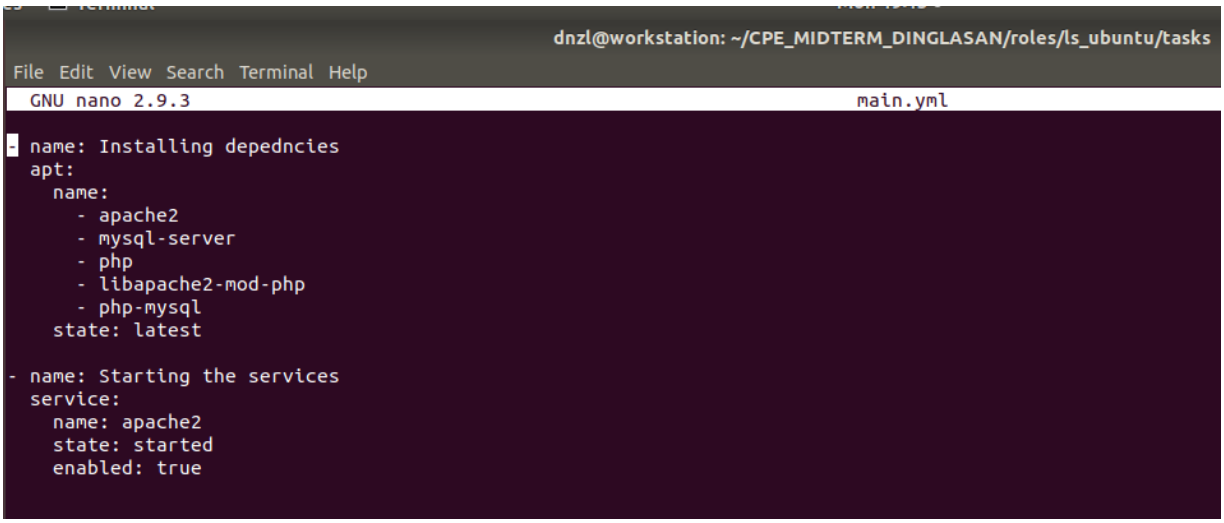
```
- name: Copying the Influxdb repository file
  unarchive:
    src: https://dl.influxdata.com/influxdb/releases/influxdb2-2.4.0-linux-amd64.tar.gz
    dest: /tmp/
    remote_src: yes
    mode: 0777
    owner: root
    group: root

- name: Adding the executables to the PATH
  shell:
    cd /tmp/influxdb2*
    sudo cp influxdb2-2.4.0-linux-amd64/influxd /usr/local/bin/

- name: Downloading Grafana package
  get_url:
    url: https://dl.grafana.com/enterprise/release/grafana-enterprise-9.2.2-1.x86_64.rpm
    dest: /tmp/grafana-enterprise-9.2.2-1.x86_64.rpm

- name: Installing Grafana
  yum:
    name: /tmp/grafana-enterprise-9.2.2-1.x86_64.rpm

- name: Enabling Grafana service
  service:
    name: grafana-server
    enabled: yes

- name: Modifying service file
  tags: es_ubuntu
  replace:
    path: /usr/lib/systemd/system/grafana-server.service
    regexp: "TimeoutStartSec=75"
    replace: "TimeoutStartSec=500"

- name: Making sure that Grafana service is started and enabled
  service:
    name: grafana-server
    enabled: true
    state: started
```

```
- name: Creating a directory for Prometheus package
  tags: directory
  file:
    path: ~/prometheus
    state: directory

- name: Downloading and extracting Prometheus
  tags: source
  unarchive:
    src: https://github.com/prometheus/prometheus/releases/download/v2.39.1/prometheus-2.39.1.linux-amd64.tar.gz
    dest: ~/prometheus
    remote_src: yes
    mode: 0777
    owner: root
    group: root

- name: Stopping the Prometheus service if exists
  shell:
    sudo systemctl stop prometheus >> /dev/null
```

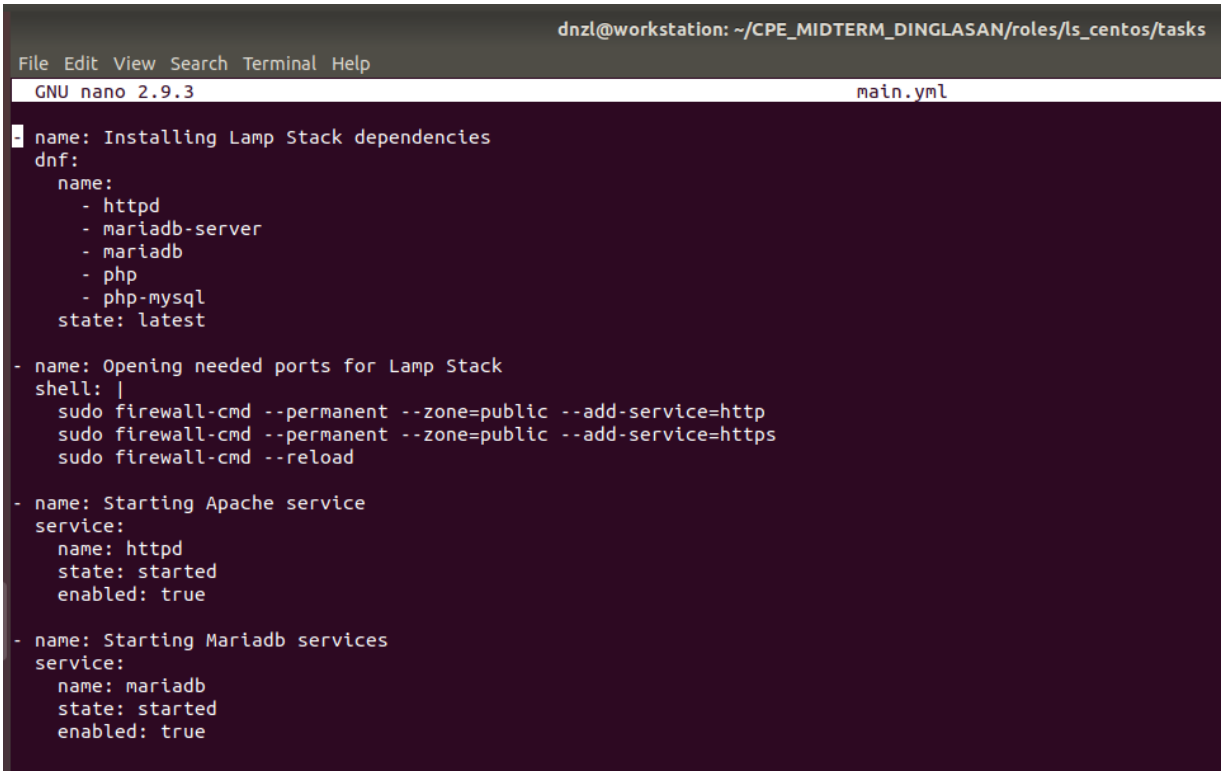
```
    sudo systemctl stop prometheus >> /dev/null
  ignore_errors: yes

- name: Adding the Prometheus executables to a PATH
  tags: executables
  shell: |
    cd ~/prometheus/prometheus*
    cp -r . /usr/local/bin/prometheus
  ignore_errors: yes


- name: Copying the Prometheus service file
  tags: servicefile
  copy:
    src: prometheus.service
    dest: /etc/systemd/system/
    owner: root
    group: root
    mode: 777
```

```
- name: Making sure that Prometheus service is started and enabled
  service:
    name: prometheus
    state: restarted
    enabled: true
```

Step 3: Paste this on the main.yml of the ls_ubuntu role.



Step 4: Paste this on the main.yml of the ls_centos role.



dnzl@workstation: ~/CPE_MIDTERM_DINGLASAN/roles/nagios_ubuntu/tasks

File Edit View Search Terminal Help

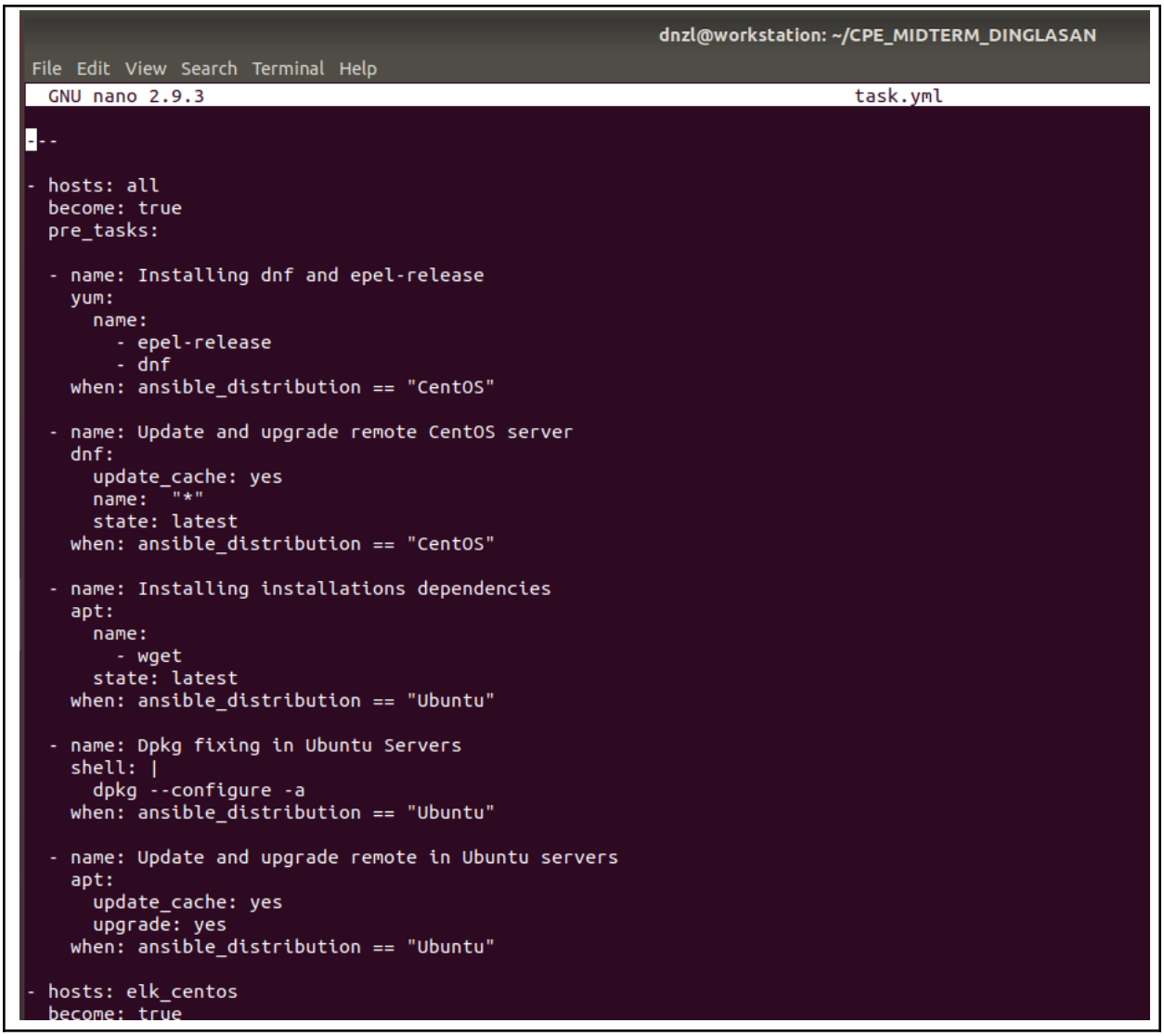
GNU nano 2.9.3 main.yml

```
---
- name: nagios libraries and dependencies (Ubuntu)
  tags: ubuntu, dependencies, libraries
  apt:
    name:
      - autoconf
      - libc6
      - gcc
      - make
      - wget
      - unzip
      - apache2
      - php
      - libapache2-mod-php7.2
      - libgd-dev
      - openssl
      - libssl-dev
      - bc
      - gawk
      - dc
      - build-essential
      - snmp
      - libnet-snmp-perl
      - gettext
      - python3
      - python3-pip
    state: latest

- name: passlib package
  pip:
    name: passlib

- name: nagios directory PATH
  file:
    path: ~/nagios
    state: directory

- name: downloading nagios
  unarchive:
    src: https://github.com/NagiosEnterprises/nagioscore/archive/nagios-4.4.6.tar.gz
    dest: ~/nagios
    remote_src: yes
    mode: 0777
    owner: root
    group: root

- name: downloading nagios plugins
  unarchive:
    src: https://github.com/nagios-plugins/nagios-plugins/archive/release-2.3.3.tar.gz
    dest: ~/nagios
    remote_src: yes
    mode: 0777
    owner: root
    group: root

- name: install, compile, adding users and groups
  shell: |
    cd ~/nagios/nagioscore-*
    sudo ./configure --with-httpd-conf=/etc/apache2/sites-enabled
    sudo make all
```

```
    sudo make install-groups-users
    sudo usermod -a -G nagios www-data
    sudo make install
    sudo make install-daemoninit
    sudo make install-commandmode
    sudo make install-config
    sudo make install-webconf
    sudo a2enmod rewrite
    sudo a2enmod cgi

- name: compile and install plugins
  shell: |
    cd ~/nagios/nagios-plugins*
    ./tools/setup
    ./configure
    make
    make install

- name: adding users to nagios
  community.general.htpasswd:
    path: /usr/local/nagios/etc/htpasswd.users
    name: dnzl
    password: dnglsn

- name: Nagios Start/Enable Check
  service:
    name: nagios
    state: restarted
    enabled: true

- name: Apache/httpd Start/Enable check
  service:
    name: apache2
    state: restarted
    enabled: true
```

```
dnzl@workstation:~/CPE_MIDTERM_DINGLASAN/files$ cat grafana.repo
[grafana]
name=grafana
baseurl=https://packages.grafana.com/oss/rpm
repo_gpgcheck=1
enabled=1
gpgcheck=1
gpgkey=https://packages.grafana.com/gpg.key
sslverify=1
sslcacert=/etc/pki/tls/certs/ca-bundle.crt
```

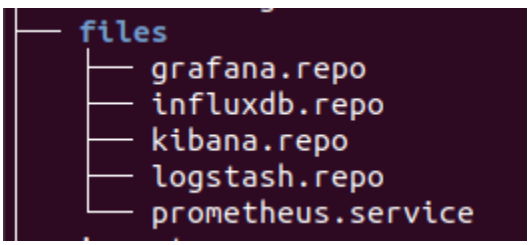
```
dnzl@workstation:~/CPE_MIDTERM_DINGLASAN/files$ cat influxdb.repo
[influxdb]
name = InfluxDB Repository - RHEL \$releasever
baseurl =
https://repos.influxdata.com/rhel/\$releasever/\$basearch/stable
enabled = 1
gpgcheck = 1
gpgkey = https://repos.influxdata.com/influxdb.key
dnzl@workstation:~/CPE_MIDTERM_DINGLASAN/files$ cat kibana.repo
[kibana-8.x]
name=Kibana repository for 8.x packages
baseurl=https://artifacts.elastic.co/packages/8.x/yum
gpgcheck=1
gpgkey=https://artifacts.elastic.co/GPG-KEY-elasticsearch
enabled=1
autorefresh=1
type=rpm-md

dnzl@workstation:~/CPE_MIDTERM_DINGLASAN/files$ cat logstash.repo
[logstash-8.x]
name=Elastic repository for 8.x packages
baseurl=https://artifacts.elastic.co/packages/8.x/yum
gpgcheck=1
gpgkey=https://artifacts.elastic.co/GPG-KEY-elasticsearch
enabled=1
autorefresh=1
type=rpm-md
dnzl@workstation:~/CPE_MIDTERM_DINGLASAN/files$ cat prometheus.service
[Unit]
Description=Prometheus Service
After=network.target

[Service]
Type=simple
ExecStart=/usr/local/bin/prometheus/prometheus --config.file=/usr/local/bin/prometheus/prometheus.yml

[Install]
WantedBy=multi-user.target
```


dnzl@workstation: ~/CPE_MIDTERM_DINGLASAN
File Edit View Search Terminal Help
GNU nano 2.9.3 task.yml

- hosts: all
become: true
pre_tasks:
- name: Installing dnf and epel-release
yum:
name:
- epel-release
- dnf
when: ansible_distribution == "CentOS"
- name: Update and upgrade remote CentOS server
dnf:
update_cache: yes
name: "*"
state: latest
when: ansible_distribution == "CentOS"
- name: Installing installations dependencies
apt:
name:
- wget
state: latest
when: ansible_distribution == "Ubuntu"
- name: Dpkg fixing in Ubuntu Servers
shell: |
dpkg --configure -a
when: ansible_distribution == "Ubuntu"
- name: Update and upgrade remote in Ubuntu servers
apt:
update_cache: yes
upgrade: yes
when: ansible_distribution == "Ubuntu"
- hosts: elk_centos
become: true

```
  roles:
    - elk_centos

- hosts: elk_ubuntu
  become: true
  roles:
    - elk_ubuntu

- hosts: nagios_ubuntu
  become: true
  roles:
    - nagios_ubuntu

- hosts: ipg_centos
  become: true
  roles:
    - ipg_centos

- hosts: ipg_ubuntu
  become: true
  roles:
    - ipg_ubuntu

- hosts: ls_centos
  become: true
  roles:
    - ls_centos

- hosts: ls_ubuntu
  become: true
  roles:
    - ls_ubuntu
```

Step 5: Create a files directory and then create these files in the files directory.



Step 6: Paste this on the said files.

Step 6: Run the playbook with the command *ansible-playbook –ask-become-pass task.yml*

GitHub link:

https://github.com/ddinglasan/CPE_MIDTERM_DINGLASAN.git

Conclusions: (link your conclusion from the objective)