Name: Daniela Marie D. Rabang	**Date Performed:** 11/06/2023
Course/Section: CPE232/CPE31S4	**Date Submitted:** 11/06/2023
Instructor: Dr. Jonathan V. Taylar	**Semester and SY:** 1st sem 2023-2024

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

1. Objectives

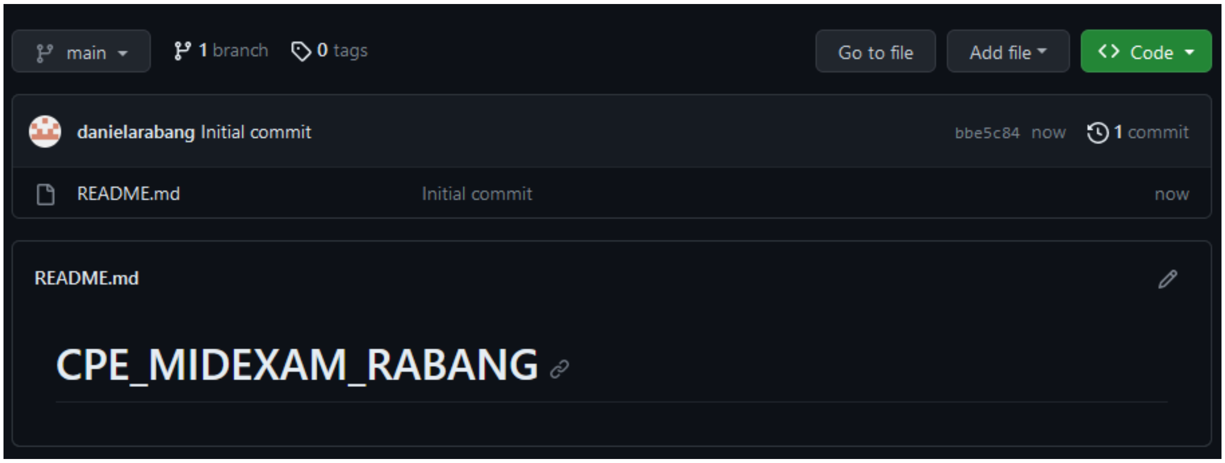
Create and design a workflow that installs, configure and manage enterprise availability, performance and log monitoring tools using Ansible as an Infrastructure as Code (IaC) tool.

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 repository.



danielarabang Initial commit bbe5c84 now 1 commit

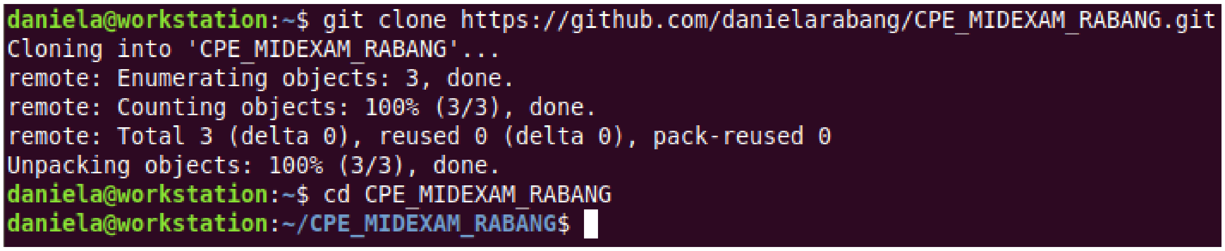
README.md Initial commit now

README.md

CPE_MIDEXAM_RABANG

- I have created a repository in my github account that is named as CPE_MIDEXAM_RABANG.

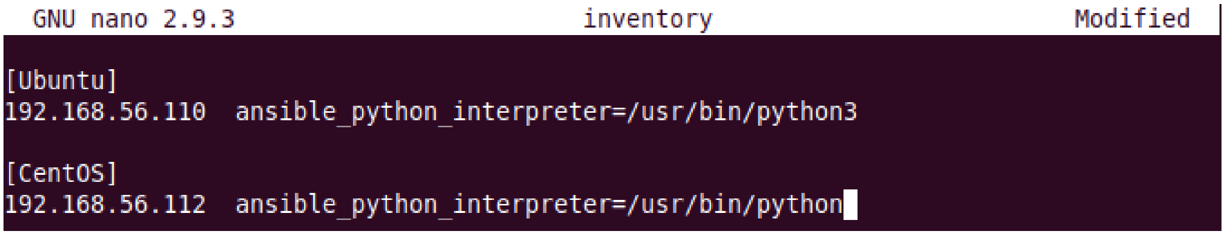
Clone the repository.



```
daniela@workstation:~$ git clone https://github.com/danielarabang/CPE_MIDEXAM_RABANG.git
Cloning into 'CPE_MIDEXAM_RABANG'...
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.
daniela@workstation:~$ cd CPE_MIDEXAM_RABANG
daniela@workstation:~/CPE_MIDEXAM_RABANG$ 
```

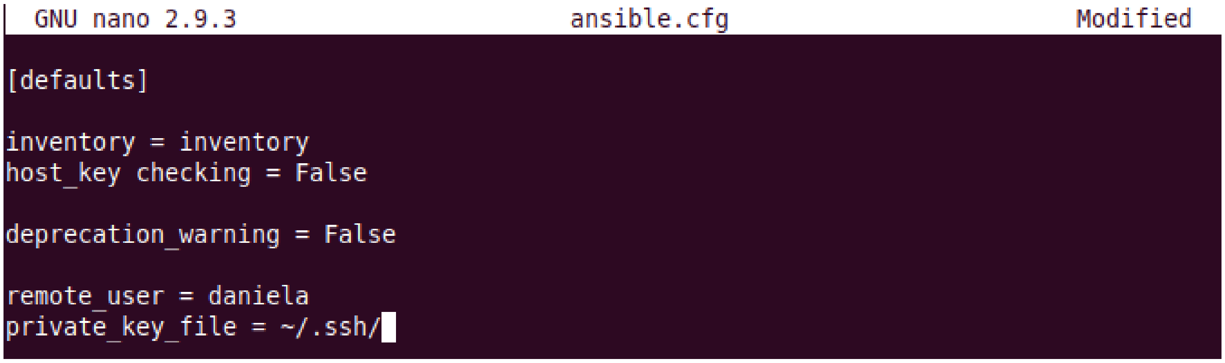
- I had cloned my repository into my workstation.

Create an inventory file.



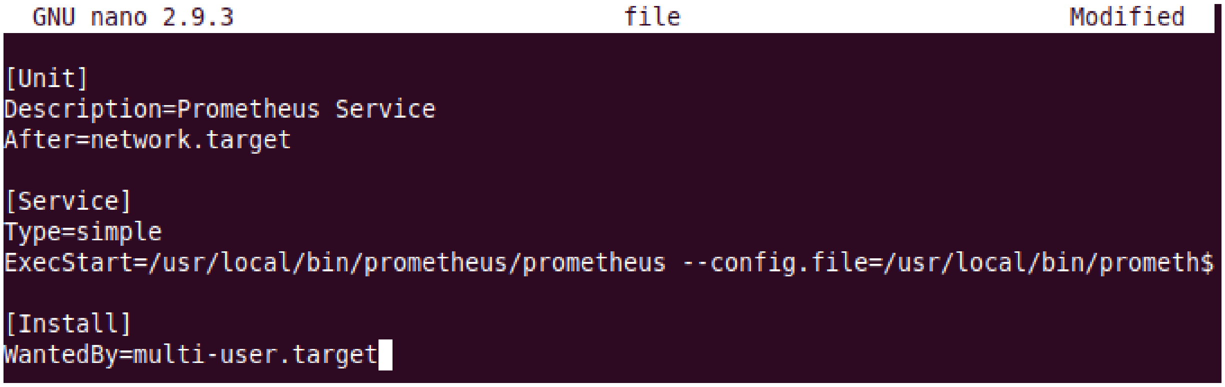
```
  GNU nano 2.9.3                        inventory                         Modified

[Ubuntu]
192.168.56.110  ansible_python_interpreter=/usr/bin/python3

[CentOS]
192.168.56.112  ansible_python_interpreter=/usr/bin/python
```

Create an ansible.cfg file.



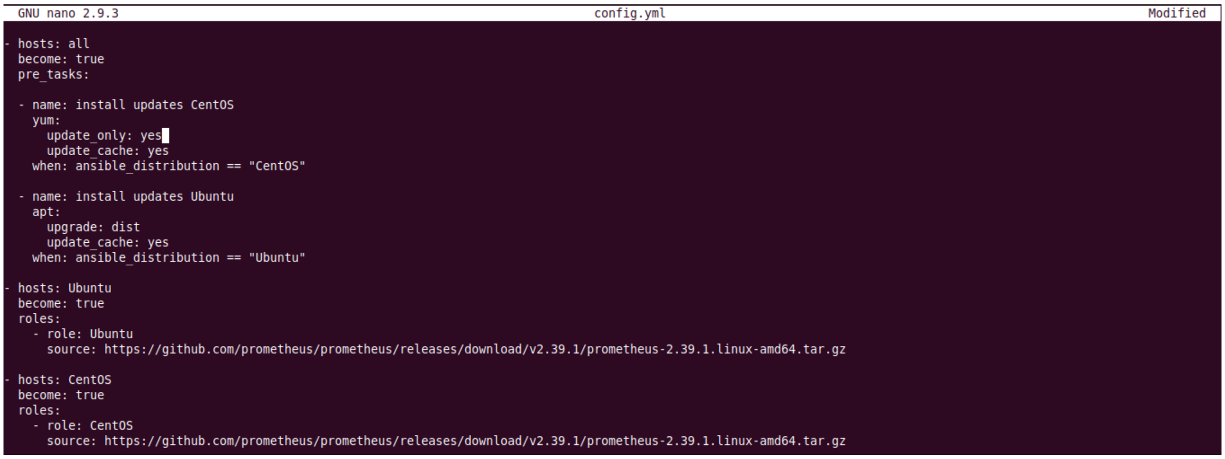
```
  GNU nano 2.9.3                        ansible.cfg                       Modified

[defaults]

inventory = inventory
host_key checking = False

deprecation_warning = False

remote_user = daniela
private_key_file = ~/.ssh/
```

Create a file



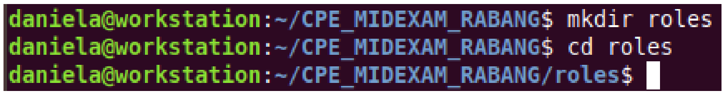
```
  GNU nano 2.9.3                          file                            Modified

[Unit]
Description=Prometheus Service
After=network.target

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

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

Create a config.yml



```
GNU nano 2.9.3                          config.yml                          Modified

- hosts: all
  become: true
  pre_tasks:

  - name: install updates CentOS
    yum:
      update_only: yes
      update_cache: yes
    when: ansible_distribution == "CentOS"

  - name: install updates Ubuntu
    apt:
      upgrade: dist
      update_cache: yes
    when: ansible_distribution == "Ubuntu"

- hosts: Ubuntu
  become: true
  roles:
    - role: Ubuntu
      source: https://github.com/prometheus/prometheus/releases/download/v2.39.1/prometheus-2.39.1.linux-amd64.tar.gz

- hosts: CentOS
  become: true
  roles:
    - role: CentOS
      source: https://github.com/prometheus/prometheus/releases/download/v2.39.1/prometheus-2.39.1.linux-amd64.tar.gz
```

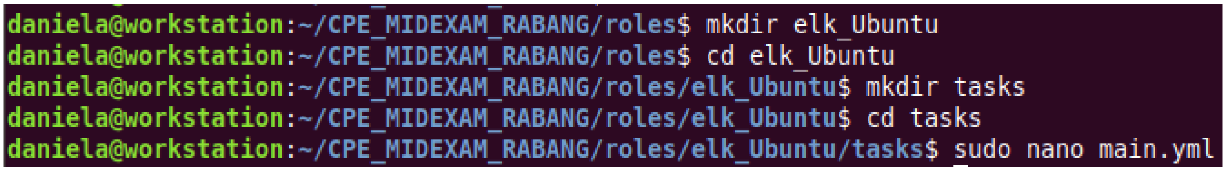
Create a directory called roles.



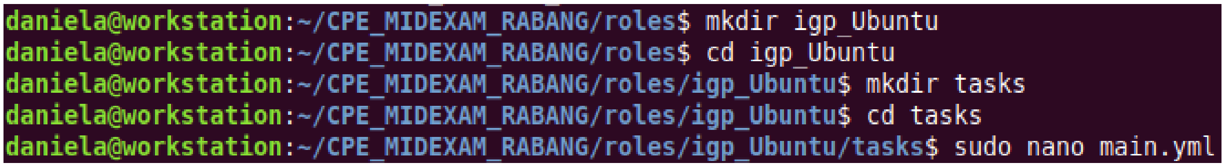
```
daniela@workstation:~/CPE_MIDEXAM_RABANG$ mkdir roles
daniela@workstation:~/CPE_MIDEXAM_RABANG$ cd roles
daniela@workstation:~/CPE_MIDEXAM_RABANG/roles$
```

In the roles directory create directory for the following packages.

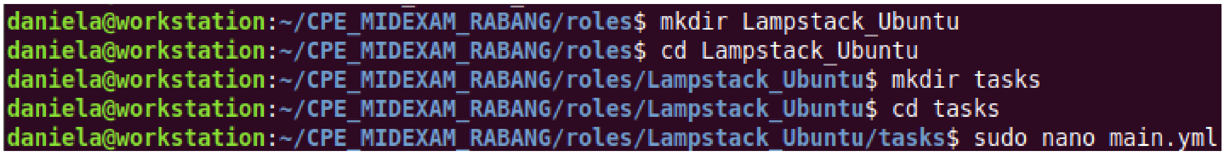
In each roles directory, create a directory that is called tasks.



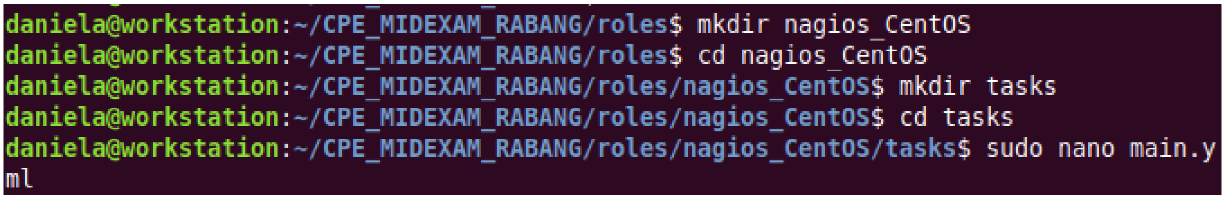
```
daniela@workstation:~/CPE_MIDEXAM_RABANG/roles$ mkdir elk_Ubuntu
daniela@workstation:~/CPE_MIDEXAM_RABANG/roles$ cd elk_Ubuntu
daniela@workstation:~/CPE_MIDEXAM_RABANG/roles/elk_Ubuntu$ mkdir tasks
daniela@workstation:~/CPE_MIDEXAM_RABANG/roles/elk_Ubuntu$ cd tasks
daniela@workstation:~/CPE_MIDEXAM_RABANG/roles/elk_Ubuntu/tasks$ sudo nano main.yml
```



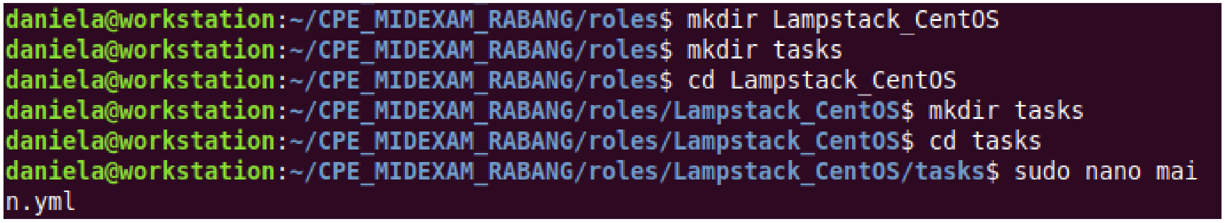
```
daniela@workstation:~/CPE_MIDEXAM_RABANG/roles$ mkdir igp_Ubuntu
daniela@workstation:~/CPE_MIDEXAM_RABANG/roles$ cd igp_Ubuntu
daniela@workstation:~/CPE_MIDEXAM_RABANG/roles/igp_Ubuntu$ mkdir tasks
daniela@workstation:~/CPE_MIDEXAM_RABANG/roles/igp_Ubuntu$ cd tasks
daniela@workstation:~/CPE_MIDEXAM_RABANG/roles/igp_Ubuntu/tasks$ sudo nano main.yml
```



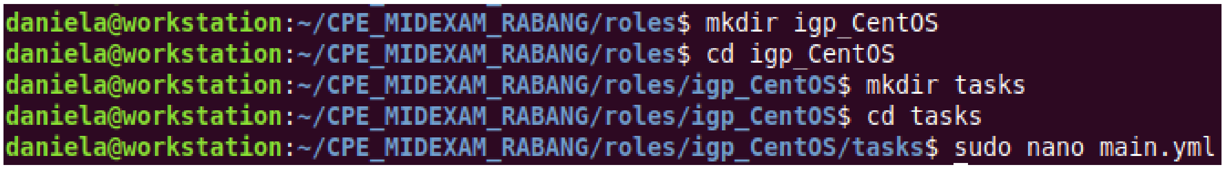
```
daniela@workstation:~/CPE_MIDEXAM_RABANG/roles$ mkdir Lampstack_Ubuntu
daniela@workstation:~/CPE_MIDEXAM_RABANG/roles$ cd Lampstack_Ubuntu
daniela@workstation:~/CPE_MIDEXAM_RABANG/roles/Lampstack_Ubuntu$ mkdir tasks
daniela@workstation:~/CPE_MIDEXAM_RABANG/roles/Lampstack_Ubuntu$ cd tasks
daniela@workstation:~/CPE_MIDEXAM_RABANG/roles/Lampstack_Ubuntu/tasks$ sudo nano main.yml
```



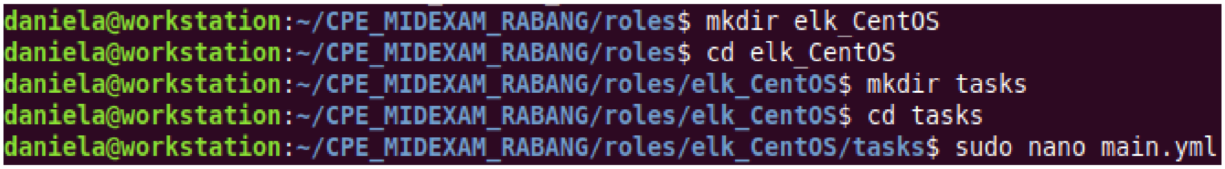
```
daniela@workstation:~/CPE_MIDEXAM_RABANG/roles$ mkdir nagios_CentOS
daniela@workstation:~/CPE_MIDEXAM_RABANG/roles$ cd nagios_CentOS
daniela@workstation:~/CPE_MIDEXAM_RABANG/roles/nagios_CentOS$ mkdir tasks
daniela@workstation:~/CPE_MIDEXAM_RABANG/roles/nagios_CentOS$ cd tasks
daniela@workstation:~/CPE_MIDEXAM_RABANG/roles/nagios_CentOS/tasks$ sudo nano main.y
ml
```



```
daniela@workstation:~/CPE_MIDEXAM_RABANG/roles$ mkdir Lampstack_CentOS
daniela@workstation:~/CPE_MIDEXAM_RABANG/roles$ mkdir tasks
daniela@workstation:~/CPE_MIDEXAM_RABANG/roles$ cd Lampstack_CentOS
daniela@workstation:~/CPE_MIDEXAM_RABANG/roles/Lampstack_CentOS$ mkdir tasks
daniela@workstation:~/CPE_MIDEXAM_RABANG/roles/Lampstack_CentOS$ cd tasks
daniela@workstation:~/CPE_MIDEXAM_RABANG/roles/Lampstack_CentOS/tasks$ sudo nano mai
n.yml
```



```
daniela@workstation:~/CPE_MIDEXAM_RABANG/roles$ mkdir igp_CentOS
daniela@workstation:~/CPE_MIDEXAM_RABANG/roles$ cd igp_CentOS
daniela@workstation:~/CPE_MIDEXAM_RABANG/roles/igp_CentOS$ mkdir tasks
daniela@workstation:~/CPE_MIDEXAM_RABANG/roles/igp_CentOS$ cd tasks
daniela@workstation:~/CPE_MIDEXAM_RABANG/roles/igp_CentOS/tasks$ sudo nano main.yml
```

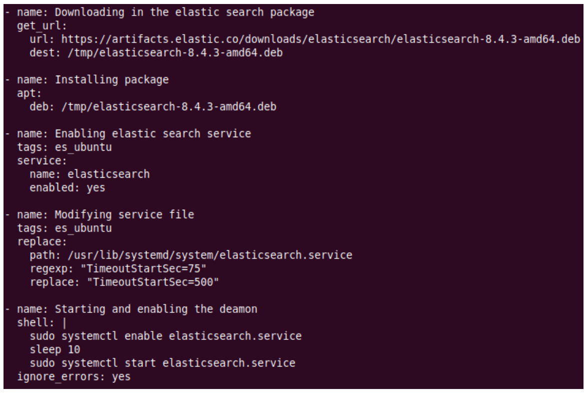


```
daniela@workstation:~/CPE_MIDEXAM_RABANG/roles$ mkdir elk_CentOS
daniela@workstation:~/CPE_MIDEXAM_RABANG/roles$ cd elk_CentOS
daniela@workstation:~/CPE_MIDEXAM_RABANG/roles/elk_CentOS$ mkdir tasks
daniela@workstation:~/CPE_MIDEXAM_RABANG/roles/elk_CentOS$ cd tasks
daniela@workstation:~/CPE_MIDEXAM_RABANG/roles/elk_CentOS/tasks$ sudo nano main.yml
```

In each directory tasks create a main.yml, run the playbook, and verify that the following are installed into the control nodes:

Ubuntu:

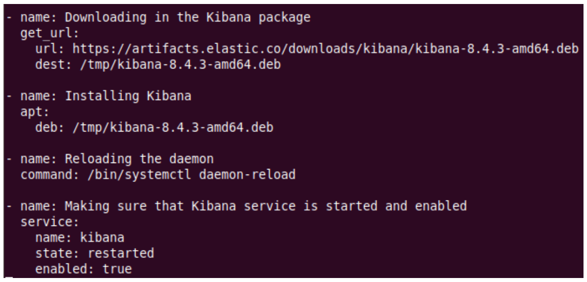
Elastic Search

Input	Process
	
Output	

```
- 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
```

Kibana

Input	Process

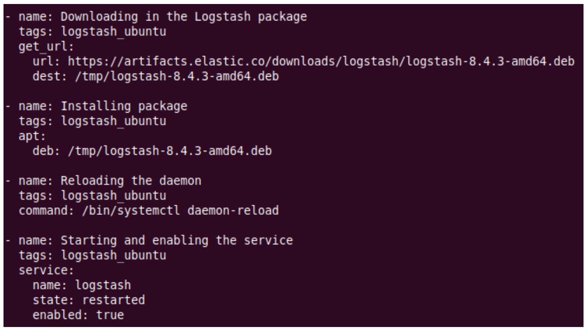


```
- name: Downloading in the Kibana package
  get_url:
    url: https://artifacts.elastic.co/downloads/kibana/kibana-8.4.3-amd64.deb
    dest: /tmp/kibana-8.4.3-amd64.deb

- 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
```

Output

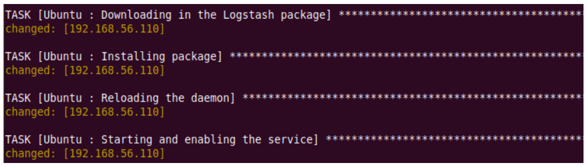
Logstash

Input



```
- 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
```

Process

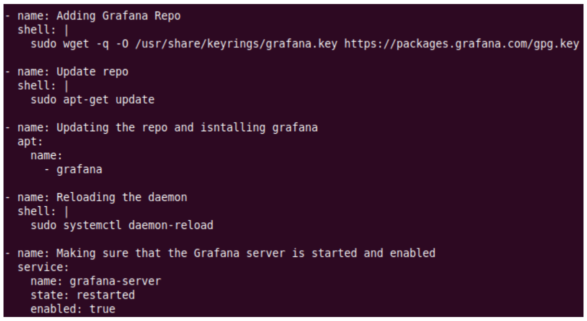


```
TASK [Ubuntu : Downloading in the Logstash package] **************************************
changed: [192.168.56.110]

TASK [Ubuntu : Installing package] *******************************************************
changed: [192.168.56.110]

TASK [Ubuntu : Reloading the daemon] *****************************************************
changed: [192.168.56.110]

TASK [Ubuntu : Starting and enabling the service] ****************************************
changed: [192.168.56.110]
```

Output

Grafana

Input

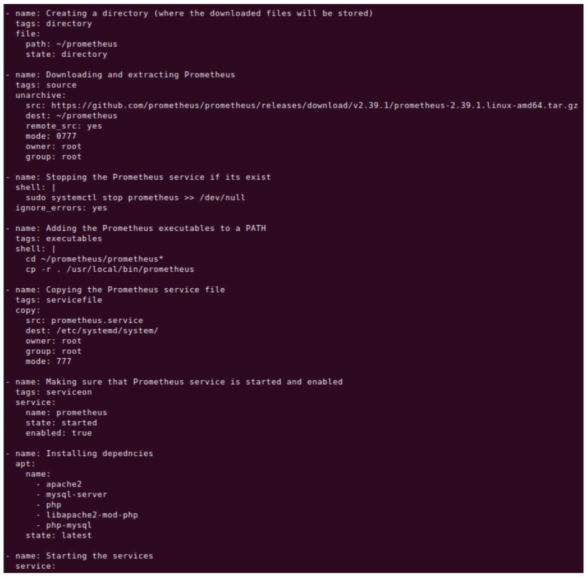


```
- name: Adding Grafana Repo
  shell: |
    sudo wget -q -O /usr/share/keyrings/grafana.key https://packages.grafana.com/gpg.key

- 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
```

Process

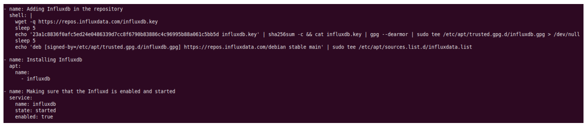
Output

Prometheus

Input	Process
	
Output	

```
- 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

- name: Installing depedncies
  apt:
    name:
      - apache2
      - mysql-server
      - php
      - libapache2-mod-php
      - php-mysql
    state: latest

- name: Starting the services
  service:
```

Influxdb

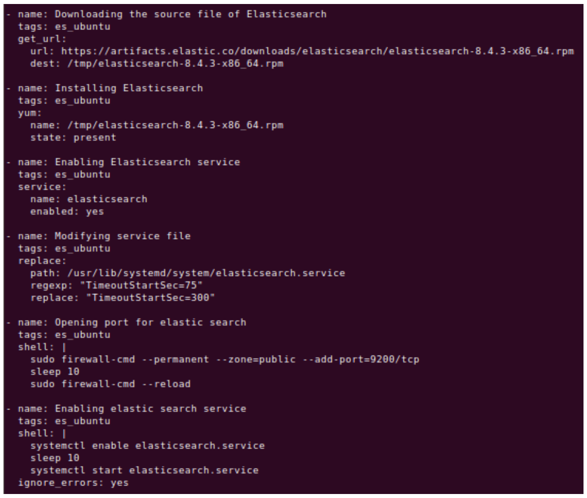
Input	Process
	
Output	

```
- 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 /etc/apt/trusted.gpg.d/influxdb.gpg > /dev/null
    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/influxdata.list

- name: Installing Influxdb
  apt:
    name:
      - influxdb

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

CentOS:

Elastic Search

Input	Process

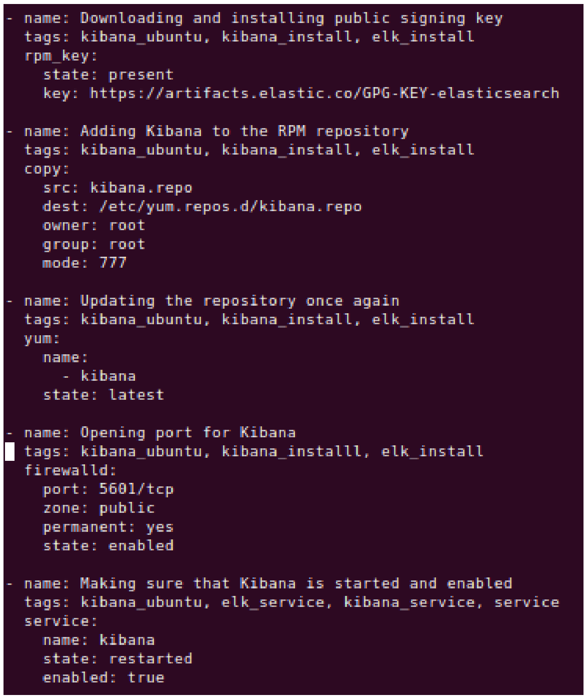


```
- name: Downloading the source file of Elasticsearch
  tags: es_ubuntu
  get_url:
    url: https://artifacts.elastic.co/downloads/elasticsearch/elasticsearch-8.4.3-x86_64.rpm
    dest: /tmp/elasticsearch-8.4.3-x86_64.rpm

- 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
```

Output

Kibana

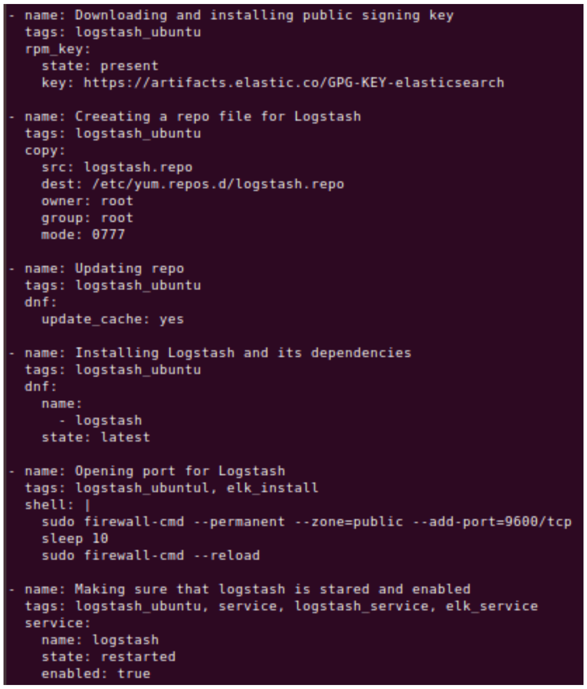
Input | **Process**



```
- 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
```

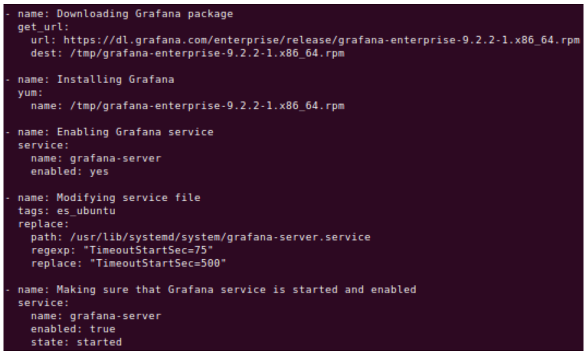
Output

Logstash

Input	Process
	
Output	

```
- 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
```

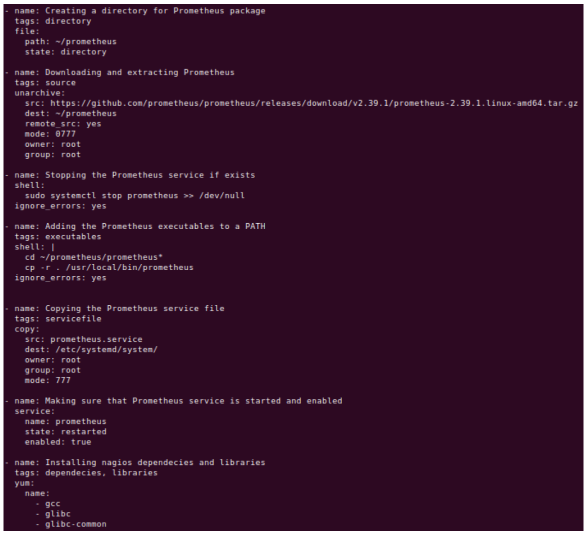
Grafana

Input	Process
	
Output	

```
- 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
```

Prometheus

<table>
<tr><th>Input</th><th>Process</th></tr>
<tr><td>

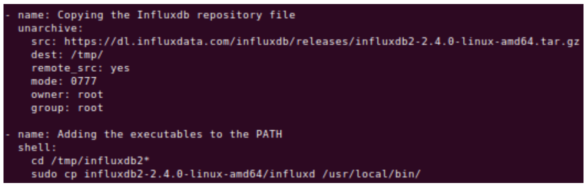


```
- name: Creating a directory for Prometheus package
  tags: directory
  file:
    path: ~/prometheus
    state: directory

- 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
  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

- name: Installing nagios dependecies and libraries
  tags: dependecies, libraries
  yum:
    name:
      - gcc
      - glibc
      - glibc-common
```

</td><td></td></tr>
<tr><th colspan="2">Output</th></tr>
<tr><td colspan="2"></td></tr>
</table>

Influxdb

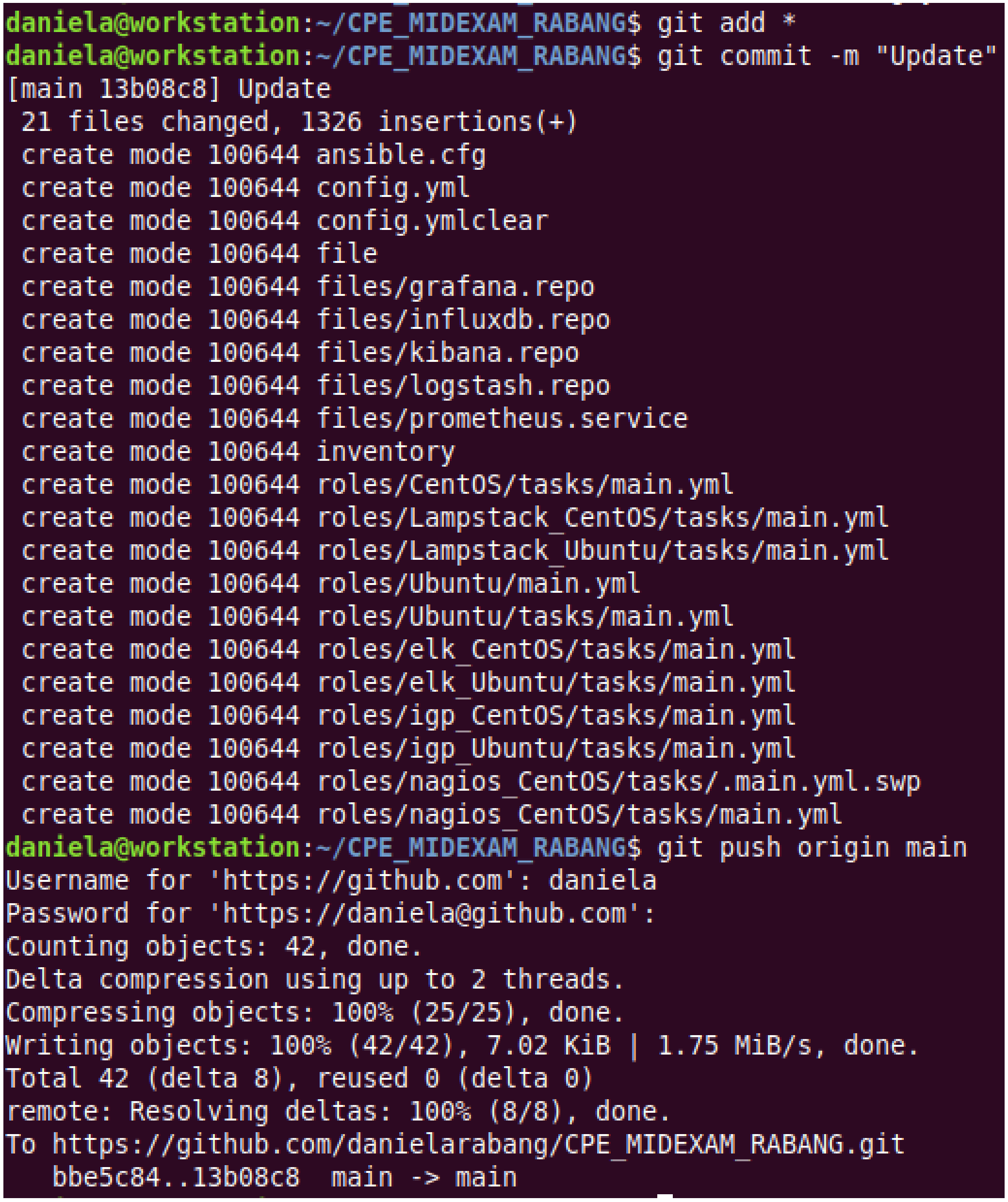
<table>
<tr><th>Input</th><th>Process</th></tr>
<tr><td>



```
- 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/
```

</td><td></td></tr>
<tr><th colspan="2">Output</th></tr>
<tr><td colspan="2"></td></tr>
</table>

Commit



```
daniela@workstation:~/CPE_MIDEXAM_RABANG$ git add *
daniela@workstation:~/CPE_MIDEXAM_RABANG$ git commit -m "Update"
[main 13b08c8] Update
 21 files changed, 1326 insertions(+)
 create mode 100644 ansible.cfg
 create mode 100644 config.yml
 create mode 100644 config.ymlclear
 create mode 100644 file
 create mode 100644 files/grafana.repo
 create mode 100644 files/influxdb.repo
 create mode 100644 files/kibana.repo
 create mode 100644 files/logstash.repo
 create mode 100644 files/prometheus.service
 create mode 100644 inventory
 create mode 100644 roles/CentOS/tasks/main.yml
 create mode 100644 roles/Lampstack_CentOS/tasks/main.yml
 create mode 100644 roles/Lampstack_Ubuntu/tasks/main.yml
 create mode 100644 roles/Ubuntu/main.yml
 create mode 100644 roles/Ubuntu/tasks/main.yml
 create mode 100644 roles/elk_CentOS/tasks/main.yml
 create mode 100644 roles/elk_Ubuntu/tasks/main.yml
 create mode 100644 roles/igp_CentOS/tasks/main.yml
 create mode 100644 roles/igp_Ubuntu/tasks/main.yml
 create mode 100644 roles/nagios_CentOS/tasks/.main.yml.swp
 create mode 100644 roles/nagios_CentOS/tasks/main.yml
daniela@workstation:~/CPE_MIDEXAM_RABANG$ git push origin main
Username for 'https://github.com': daniela
Password for 'https://daniela@github.com':
Counting objects: 42, done.
Delta compression using up to 2 threads.
Compressing objects: 100% (25/25), done.
Writing objects: 100% (42/42), 7.02 KiB | 1.75 MiB/s, done.
Total 42 (delta 8), reused 0 (delta 0)
remote: Resolving deltas: 100% (8/8), done.
To https://github.com/danielarabang/CPE_MIDEXAM_RABANG.git
   bbe5c84..13b08c8  main -> main
```

GitHub link:

https://github.com/danielarabang/CPE_MIDEXAM_RABANG.git

Conclusions: (link your conclusion from the objective)