

Name: Planta, Calvin Earl L.	Date Performed: 10/10/25
Course/Section: CPE 212 - CPE31S2	Date Submitted: 10/10/25
Instructor: Engr. Robin Valenzuela	Semester and SY: 1st Sem SY 25-26
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	
<ol style="list-style-type: none"> 1. Create a repository in your GitHub account and label it CPE_MIDEXAM_SURNAME. 2. Clone the repository and do the following: <ol style="list-style-type: none"> 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 separate 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)	

1 **General**

Owner * Calvin-Earl / **Repository name *** CPE212_MIDTERM
 CPE212_MIDTERM is available.

Great repository names are short and memorable. How about [upgraded-broccoli](#)?

Description
0 / 350 characters

2 **Configuration**

Choose visibility * Public

Choose who can see and commit to this repository

Add README On
READMEs can be used as longer descriptions. [About READMEs](#)

Add .gitignore No .gitignore
.gitignore tells git which files not to track. [About ignoring files](#)

Add license No license
Licenses explain how others can use your code. [About licenses](#)

[Create repository](#)

2.1.

ansible.cfg

```
GNU nano 7.2                               ansible.cfg
[defaults]
inventory = inventory.yaml
private_key_file = ~/.ssh/ansible
```

inventory.yaml

```
GNU nano 7.2                                         inventory.yaml *
```

```
[elasticstack]
192.168.56.108
192.168.56.117

[nagios]
192.168.56.107

[lampstack]
192.168.56.108
192.168.56.117
```

I grouped the hosts based on the midterm tasks.

2.2.

main.yml (elasticstack role)

```
CPE212_MIDTERM > roles > elasticstack > tasks > ! main.yml
1  ---
2  - name: add elasticstack prerequisites (Ubuntu)
3    apt:
4      name: apt-transport-https
5      state: present
6      when: ansible_distribution == "Ubuntu"
7
8  - name: add elasticstack prerequisites (CentOS)
9    dnf:
10   name: curl
11   state: present
12   when: ansible_distribution == "CentOS"
13
14 - name: add elasticsearch GPG key (Ubuntu)
15   apt_key:
16     url: https://artifacts.elastic.co/GPG-KEY-elasticsearch
17     state: present
18   when: ansible_distribution == "Ubuntu"
19
20 - name: import elasticstack repository (Ubuntu)
21   apt_repository:
22     repo: "deb https://artifacts.elastic.co/packages/8.x/apt stable main"
23     state: present
24   when: ansible_distribution == "Ubuntu"
25
26 - name: Add Elasticsearch Yum Repository
27   yum_repository:
28     name: elasticsearch
29     description: Elasticsearch repository for 8.x packages
30     baseurl: https://artifacts.elastic.co/packages/8.x/yum
31     gpgcheck: yes
32     gpgkey: https://artifacts.elastic.co/GPG-KEY-elasticsearch
33     enabled: yes
34   when: ansible_distribution == "CentOS"
```

```
CPE212_MIDTERM > roles > elasticstack > tasks > ! main.yml
35
36 - name: install elasticstack for Ubuntu (Elasticsearch, Kibana, Logstash)
37   apt:
38     name:
39       - elasticsearch
40       - kibana
41       - logstash
42     state: present
43   when: ansible_distribution == "Ubuntu"
44
45 - name: install elasticstack for CentOS (Elasticsearch, Kibana, Logstash)
46   dnf:
47     name:
48       - elasticsearch
49       - kibana
50       - logstash
51     state: latest
52   when: ansible_distribution == "CentOS"
53
```

```
CPE212_MIDTERM > roles > elasticstack > tasks > ! main.yml
54   - name: start installed services (Ubuntu)
55     service:
56       name:
57         - elasticsearch
58         - kibana
59         - logstash
60       state: restarted
61       enabled: yes
62     when: ansible_distribution == "Ubuntu"
63
64   - name: start installed services (CentOS)
65     service:
66       name:
67         - elasticsearch
68         - kibana
69         - logstash
70       state: restarted
71       enabled: yes
72     when: ansible_distribution == "CentOS"
```

This playbook installs the Elasticstack (elasticsearch, kibana, logstash) for both Ubuntu and CentOS servers. For the packages to be installed, their respective GPG keys and repositories need to be imported in order for the system to recognize the packages to be installed. Afterwards, it also starts all the services for both servers.

main.yml (nagios role)

```
GNU nano 7.2                               main.yml *
```

```
---
- name: install nagios for one host (workstation)
  apt:
    name:
      - nagios4
    state: present
    update_cache: yes
  when: ansible_distribution == "Ubuntu"

- name: start nagios for ubuntu
  tags: nagios, start, ubuntu
  service:
    name: nagios4
    state: started
    enabled: true
  when: ansible_distribution == "Ubuntu"
```

This playbook installs and starts the nagios4 monitoring tool to one host. I chose the workstation node for this installation.

main.yml (lampstack role)

```
GNU nano 7.2                               main.yml *
```

```
---
- name: install lampstack (Ubuntu)
  apt:
    name:
      - apache2
      - libapache2-mod-php
      - mariadb-server
    state: latest
  when: ansible_distribution == "Ubuntu"

- name: install lampstack (CentOS)
  dnf:
    name:
      - httpd
      - php
      - mariadb-server
    state: latest
  when: ansible_distribution == "CentOS"
```

```
GNU nano 7.2                                     main.yml *
```

```
- httpd
- php
- mariadb-server
  state: latest
when: ansible_distribution == "CentOS"

- name: start mariadb service (Ubuntu)
  service:
    name: mariadb
    state: restarted
    enabled: yes
  when: ansible_distribution == "Ubuntu"

- name: start mariadb service (CentOS)
  service:
    name: mariadb
    state: restarted
    enabled: yes
  when: ansible_distribution == "CentOS"
```

This playbook installs the Lampstack (Apache2/httpd, php, mariadb) for both Ubuntu and CentOS nodes. It also starts the mariadb service for both servers as well.

play.yml

```
CPE212_MIDTERM > ! play.yml
1   ---
2   - hosts: elasticstack
3     become: true
4     roles:
5       - elasticstack
6
7   - hosts: nagios
8     become: true
9     roles:
10    - nagios
11
12  - hosts: lampstack
13    become: true
14    roles:
15      - lampstack
```

This is the playbook to be run, where it calls the roles to do the tasks from their respective main.yml files. The concept of roles can be seen in this play, as I only called the roles, rather than listing it all in this playbook.

3.

PLAY

```
vbearl@workstation:~/CPE212_MIDTERM$ ansible-playbook play.yml -K
BECOME password:

PLAY [elasticstack] ****
TASK [Gathering Facts] ****
ok: [192.168.56.108]
ok: [192.168.56.117]

TASK [elasticstack : add elasticstack prerequisites (Ubuntu)] ****
skipping: [192.168.56.117]
ok: [192.168.56.108]

TASK [elasticstack : add elasticstack prerequisites (CentOS)] ****
skipping: [192.168.56.108]
ok: [192.168.56.117]
```

```
TASK [elasticstack : add elasticsearch GPG key (Ubuntu)] ****
skipping: [192.168.56.117]
ok: [192.168.56.108]

TASK [elasticstack : import elasticstack repository (Ubuntu)] ****
skipping: [192.168.56.117]
ok: [192.168.56.108]

TASK [elasticstack : Add Elasticsearch Yum Repository] ****
skipping: [192.168.56.108]
ok: [192.168.56.117]

TASK [elasticstack : install elasticstack for Ubuntu (Elasticsearch, Kibana, Logstash)] ***
skipping: [192.168.56.117]
ok: [192.168.56.108]

TASK [elasticstack : install elasticstack for CentOS (Elasticsearch, Kibana, Logstash)] ***
skipping: [192.168.56.108]
ok: [192.168.56.117]
```

```
TASK [elasticstack : start elasticsearch (Ubuntu)] ****
skipping: [192.168.56.117]
changed: [192.168.56.108]

TASK [elasticstack : start kibana (Ubuntu)] ****
skipping: [192.168.56.117]
changed: [192.168.56.108]

TASK [elasticstack : start logstash (Ubuntu)] ****
skipping: [192.168.56.117]
changed: [192.168.56.108]

TASK [elasticstack : start elasticsearch (CentOS)] ****
skipping: [192.168.56.108]
changed: [192.168.56.117]

TASK [elasticstack : start kibana (CentOS)] ****
skipping: [192.168.56.108]
changed: [192.168.56.117]

TASK [elasticstack : start logstash (CentOS)] ****
skipping: [192.168.56.108]
changed: [192.168.56.117]
```

```
PLAY [nagios] ****
TASK [Gathering Facts] ****
ok: [192.168.56.107]

TASK [nagios : install nagios for one host (workstation)] ****
ok: [192.168.56.107]

TASK [nagios : start nagios for ubuntu] ****
ok: [192.168.56.107]
```

```
PLAY [lampstack] ****
TASK [Gathering Facts] ****
ok: [192.168.56.108]
ok: [192.168.56.117]

TASK [lampstack : install lampstack (Ubuntu)] ****
skipping: [192.168.56.117]
ok: [192.168.56.108]

TASK [lampstack : install lampstack (CentOS)] ****
skipping: [192.168.56.108]
changed: [192.168.56.117]

TASK [lampstack : start mariadb service (Ubuntu)] ****
skipping: [192.168.56.117]
changed: [192.168.56.108]

TASK [lampstack : start mariadb service (CentOS)] ****
skipping: [192.168.56.108]
changed: [192.168.56.117]
```

```
PLAY RECAP ****
192.168.56.107      : ok=3    changed=0    unreachable=0    failed=0    s
kipped=0  rescued=0  ignored=0
192.168.56.108      : ok=11   changed=4    unreachable=0    failed=0    s
kipped=8  rescued=0  ignored=0
192.168.56.117      : ok=10   changed=5    unreachable=0    failed=0    s
kipped=9  rescued=0  ignored=0
```

As observed, all the plays listed in the playbook were successful. In all servers, all the packages were installed and started.

PROOFS

Ubuntu



this is a sample html

I was able to access the apache2 homepage, confirming that the apache2 package was successfully installed. (Note: in the previous lab activity, I changed the homepage for httpd to a sample html document that I created, which explains why this page shows up rather than the usual apache homepage.)

```
vbearl@server1:~$ sudo systemctl status mariadb
[sudo] password for vbearl:
● mariadb.service - MariaDB 10.11.13 database server
  Loaded: loaded (/usr/lib/systemd/system/mariadb.service; enabled; preset: >
  Active: active (running) since Fri 2025-10-10 08:35:35 UTC; 2h 17min ago
    Docs: man:mariadb(8)
          https://mariadb.com/kb/en/library/systemd/
   Main PID: 1456 (mariadbd)
     Status: "Taking your SQL requests now..."
       Tasks: 11 (limit: 30383)
      Memory: 108.3M (peak: 112.9M)
        CPU: 1.949s
      CGroup: /system.slice/mariadb.service
              └─1456 /usr/sbin/mariadbd

Oct 10 08:35:34 server1 mariadb[1456]: 2025-10-10  8:35:34 0 [Note] Plugin 'FE>
Oct 10 08:35:34 server1 mariadb[1456]: 2025-10-10  8:35:34 0 [Note] InnoDB: Lo>
Oct 10 08:35:34 server1 mariadb[1456]: 2025-10-10  8:35:34 0 [Warning] You nee>
Oct 10 08:35:34 server1 mariadb[1456]: 2025-10-10  8:35:34 0 [Note] Server soc>
Oct 10 08:35:34 server1 mariadb[1456]: 2025-10-10  8:35:34 0 [Note] InnoDB: Bu>
Oct 10 08:35:34 server1 mariadb[1456]: 2025-10-10  8:35:34 0 [Note] /usr/sbin/>
Oct 10 08:35:34 server1 mariadb[1456]: Version: '10.11.13-MariaDB-0ubuntu0.24.>
Oct 10 08:35:35 server1 systemd[1]: Started mariadb.service - MariaDB 10.11.13 >
```

mariadb service

```
vbearl@server1:~$ sudo systemctl status elasticsearch
[sudo] password for vbearl:
● elasticsearch.service - Elasticsearch
  Loaded: loaded (/usr/lib/systemd/system/elasticsearch.service; enabled; pr>
  Active: activating (start) since Fri 2025-10-10 11:00:43 UTC; 31s ago
    Docs: https://www.elastic.co...
   Main PID: 1362 (java)
     Tasks: 45 (limit: 9204)
    Memory: 4.2G (peak: 4.2G)
      CPU: 50.998s
    CGroup: /system.slice/elasticsearch.service
            ├─1362 /usr/share/elasticsearch/jdk/bin/java -Xms4m -Xmx64m -XX:+U...
            ├─2241 /usr/share/elasticsearch/jdk/bin/java -Des.networkaddress.c...
            └─2530 /usr/share/elasticsearch/modules/x-pack-ml/platform/linux-x...

Oct 10 11:00:43 server1 systemd[1]: Starting elasticsearch.service - Elasticsea...
lines 1-14/14 (END)
```

elasticsearch service

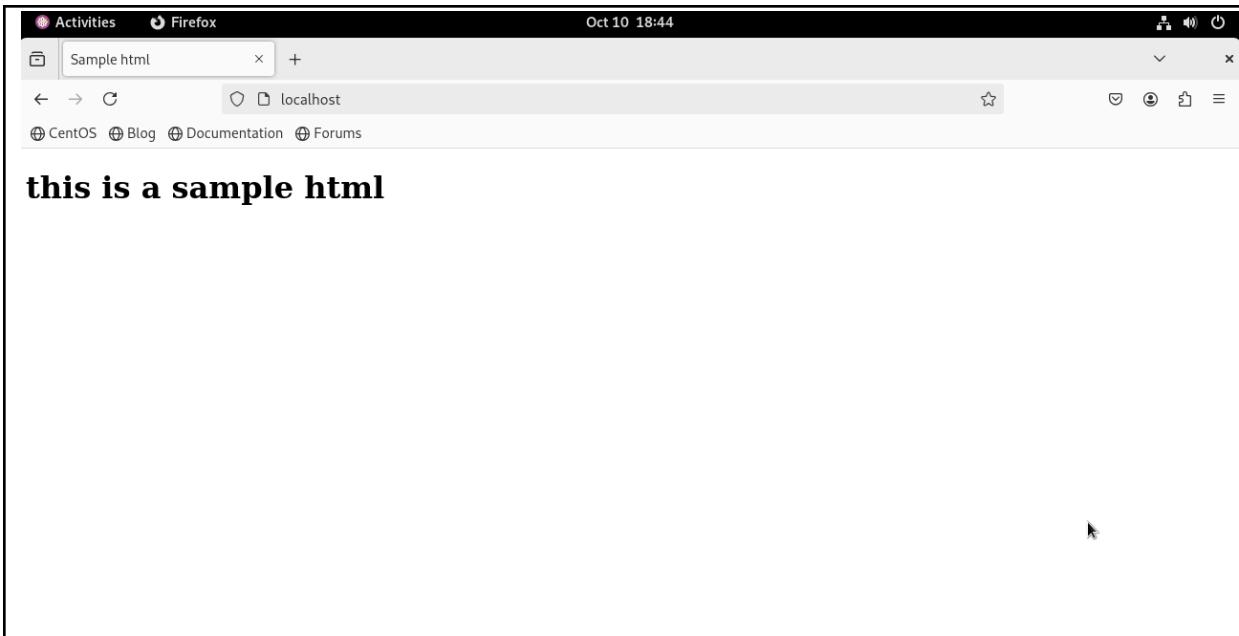
```
vbearl@server1:~$ sudo systemctl status kibana
● kibana.service - Kibana
  Loaded: loaded (/usr/lib/systemd/system/kibana.service; enabled; preset: en>
  Active: active (running) since Fri 2025-10-10 11:00:43 UTC; 1min 5s ago
    Docs: https://www.elastic.co...
    Main PID: 1367 (node)
      Tasks: 11 (limit: 9204)
     Memory: 458.7M (peak: 594.4M)
        CPU: 24.214s
       CGroup: /system.slice/kibana.service
               └─1367 /usr/share/kibana/bin/../node/glibc-217/bin/node /usr/share/...
Oct 10 11:00:53 server1 kibana[1367]: Native global console methods have been o...
Oct 10 11:00:58 server1 kibana[1367]: [2025-10-10T11:00:58.405+00:00][INFO ][ro...
Oct 10 11:00:58 server1 kibana[1367]: [2025-10-10T11:00:58.479+00:00][INFO ][no...
Oct 10 11:01:25 server1 kibana[1367]: [2025-10-10T11:01:25.173+00:00][INFO ][pl...
Oct 10 11:01:25 server1 kibana[1367]: [2025-10-10T11:01:25.272+00:00][INFO ][ht...
Oct 10 11:01:25 server1 kibana[1367]: [2025-10-10T11:01:25.436+00:00][INFO ][pl...
Oct 10 11:01:25 server1 kibana[1367]: [2025-10-10T11:01:25.477+00:00][INFO ][pr...
Oct 10 11:01:25 server1 kibana[1367]: [2025-10-10T11:01:25.520+00:00][INFO ][ro...
Oct 10 11:01:31 server1 kibana[1367]: i Kibana has not been configured.
Oct 10 11:01:31 server1 kibana[1367]: Go to http://localhost:5601/?code=402709 >
Lines 1-21/21 (END)
```

kibana service

```
vbearl@server1:~$ sudo systemctl status logstash
● logstash.service - logstash
  Loaded: loaded (/usr/lib/systemd/system/logstash.service; enabled; preset:>
  Active: active (running) since Fri 2025-10-10 11:02:09 UTC; 1s ago
    Main PID: 3928 (java)
      Tasks: 24 (limit: 9204)
     Memory: 152.1M (peak: 152.1M)
        CPU: 2.960s
       CGroup: /system.slice/logstash.service
               └─3928 /usr/share/logstash/jdk/bin/java -Xms1g -Xmx1g -Djava.awt.h...
Oct 10 11:02:09 server1 systemd[1]: logstash.service: Scheduled restart job, re>
Oct 10 11:02:09 server1 systemd[1]: Started logstash.service - logstash.
Oct 10 11:02:09 server1 logstash[3928]: Using bundled JDK: /usr/share/logstash/>
Lines 1-13/13 (END)
```

logstash service

CentOS



I was able to access the httpd homepage, confirming that the httpd package was successfully installed. (Note: in the previous lab activity, I changed the homepage for httpd to a sample html document that I created, which explains why this page shows up rather than the usual apache homepage.

```
[vbearl@centOS ~]$ sudo systemctl status mariadb
● mariadb.service - MariaDB 10.5 database server
  Loaded: loaded (/usr/lib/systemd/system/mariadb.service; enabled; preset: >
  Active: active (running) since Fri 2025-10-10 16:36:17 PST; 2h 7min ago
    Docs: man:mariadb(8)
          https://mariadb.com/kb/en/library/systemd/
   Main PID: 1188 (mariadb)
     Status: "Taking your SQL requests now..."
      Tasks: 8 (limit: 22979)
     Memory: 8.1M (peak: 95.0M)
        CPU: 1.878s
       CGroup: /system.slice/mariadb.service
                 └─1188 /usr/libexec/mariadb --basedir=/usr

Oct 10 16:36:16 centOS systemd[1]: Starting MariaDB 10.5 database server...
Oct 10 16:36:16 centOS mariadb-check-socket[1002]: Socket file /var/lib/mysql/m>
Oct 10 16:36:16 centOS mariadb-check-socket[1002]: No process is using /var/lib/>
Oct 10 16:36:16 centOS mariadb-prepare-db-dir[1085]: Database MariaDB is probab>
Oct 10 16:36:16 centOS mariadb-prepare-db-dir[1085]: If this is not the case, m>
Oct 10 16:36:17 centOS systemd[1]: Started MariaDB 10.5 database server.
```

mariadb service

```
[vbearl@centOS ~]$ sudo systemctl status elasticsearch
● elasticsearch.service - Elasticsearch
   Loaded: loaded (/usr/lib/systemd/system/elasticsearch.service; enabled; pr>
   Active: active (running) since Fri 2025-10-10 18:39:19 PST; 7min ago
     Docs: https://www.elastic.co...
     Main PID: 14681 (java)
        Tasks: 109 (limit: 22979)
       Memory: 901.0M (peak: 2.3G)
          CPU: 54.907s
        CGroup: /system.slice/elasticsearch.service
                  └─14681 /usr/share/elasticsearch/jdk/bin/java -Xms4m -Xmx64m -XX:+>
                     ├─14745 /usr/share/elasticsearch/jdk/bin/java -Des.networkaddress.=>
                     └─14769 /usr/share/elasticsearch/modules/x-pack-ml/platform/linux->

Oct 10 18:38:55 centOS systemd[1]: Starting Elasticsearch...
Oct 10 18:39:19 centOS systemd[1]: Started Elasticsearch.
Lines 1-15/15 (END)
```

elasticsearch service

```
[vbearl@centOS ~]$ sudo systemctl status kibana
● kibana.service - Kibana
   Loaded: loaded (/usr/lib/systemd/system/kibana.service; enabled; preset: d>
   Active: active (running) since Fri 2025-10-10 18:39:30 PST; 7min ago
     Docs: https://www.elastic.co...
     Main PID: 15057 (node)
        Tasks: 11 (limit: 22979)
       Memory: 305.8M (peak: 523.6M)
          CPU: 17.060s
        CGroup: /system.slice/kibana.service
                  └─15057 /usr/share/kibana/bin/.../node/glibc-217/bin/node /usr/shar>

Oct 10 18:39:33 centOS kibana[15057]: Native global console methods have been o>
Oct 10 18:39:35 centOS kibana[15057]: [2025-10-10T18:39:35.523+08:00] [INFO ] [ro>
Oct 10 18:39:35 centOS kibana[15057]: [2025-10-10T18:39:35.560+08:00] [INFO ] [no>
Oct 10 18:39:50 centOS kibana[15057]: [2025-10-10T18:39:49.964+08:00] [INFO ] [pl>
Oct 10 18:39:50 centOS kibana[15057]: [2025-10-10T18:39:50.109+08:00] [INFO ] [ht>
Oct 10 18:39:50 centOS kibana[15057]: [2025-10-10T18:39:50.201+08:00] [INFO ] [pl>
Oct 10 18:39:50 centOS kibana[15057]: [2025-10-10T18:39:50.229+08:00] [INFO ] [pr>
Oct 10 18:39:50 centOS kibana[15057]: [2025-10-10T18:39:50.259+08:00] [INFO ] [ro>
Oct 10 18:39:58 centOS kibana[15057]: i Kibana has not been configured.
Oct 10 18:39:58 centOS kibana[15057]: Go to http://localhost:5601/?code=348840 >
Lines 1-21/21 (END)
```

kibana service

```
[vbearl@centOS ~]$ sudo systemctl status logstash
● logstash.service - logstash
   Loaded: loaded (/usr/lib/systemd/system/logstash.service; enabled; preset:>)
   Active: active (running) since Fri 2025-10-10 18:47:20 PST; 5s ago
     Main PID: 18513 (java)
        Tasks: 25 (limit: 22979)
       Memory: 325.6M (peak: 325.8M)
          CPU: 14.865s
        CGroup: /system.slice/logstash.service
                  └─18513 /usr/share/logstash/jdk/bin/java -Xms1g -Xmx1g -Djava.awt.>
Oct 10 18:47:20 centOS systemd[1]: Started logstash.
Oct 10 18:47:20 centOS logstash[18513]: Using bundled JDK: /usr/share/logstash/>
lines 1-12/12 (END)
```

logstash service

Workstation

The screenshot shows a web browser window with the URL <http://localhost/nagios4/>. The page is titled "Nagios®". The left sidebar contains navigation links for "General", "Current Status", "Reports", and "System". The "Current Status" section is expanded, showing links for "Tactical Overview", "Map (Legacy)", "Hosts", "Services", "Host Groups", "Service Groups", and "Problems". The "Reports" section is also expanded, showing links for "Availability", "Trends (Legacy)", "Alerts", "History", "Summary", and "Histogram (Legacy)". The "System" section shows links for "Comments", "Downtime", "Process Info", and "Performance Info". On the right side of the page, there is a large "Nagios® Core™" logo and the text "Nagios® Core™ Version 4.4.6 April 28, 2020". Below this, there is a copyright notice: "Copyright © 2010-2020 Nagios Core Development Team and Community Contributors. Copyright © 1999-2009 Ethan Galstad. See the THANKS file for more information on contributors." and a note about the GNU General Public License.

I was able to access the nagios homepage, confirming that the nagios4 package was successful.

4.

```
vbearl@workstation:~/CPE212_MIDTERM$ git status
On branch main
Your branch is up to date with 'origin/main'.

Untracked files:
  (use "git add <file>..." to include in what will be committed)
    ansible.cfg
    inventory.yaml
    play.yml
    roles/
nothing added to commit but untracked files present (use "git add" to track)
vbearl@workstation:~/CPE212_MIDTERM$ git add .
vbearl@workstation:~/CPE212_MIDTERM$ git commit -m "midterm files"
>
[main 710357c] midterm files
 6 files changed, 170 insertions(+)
 create mode 100644 ansible.cfg
 create mode 100644 inventory.yaml
 create mode 100644 play.yml
 create mode 100644 roles/elasticsearch/tasks/main.yml
 create mode 100644 roles/lampstack/tasks/main.yml
 create mode 100644 roles/nagios/tasks/main.yml
vbearl@workstation:~/CPE212_MIDTERM$ git push origin main
```

Here, I committed all the necessary files created for this midterm skills exam. I then pushed them so I could store them in my repository.

```
● vbearl@workstation:~/CPE212_MIDTERM$ git push origin main
Enumerating objects: 16, done.
Counting objects: 100% (16/16), done.
Delta compression using up to 5 threads
Compressing objects: 100% (9/9), done.
Writing objects: 100% (15/15), 1.74 KiB | 1.74 MiB/s, done.
Total 15 (delta 1), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (1/1), done.
To https://github.com/Calvin-Earl/CPE212_MIDTERM.git
 0133721..710357c  main -> main
```

Note: I realized that I cloned this link in VSCode, which explains why I couldn't push it using the linux terminal, as it asks for a username and password prompt. Here is a legitimate documentation of the files being pushed to my github repository.

The screenshot shows a GitHub repository page for 'CPE212_MIDTERM'. At the top, it indicates 'main' branch, 1 Branch, 0 Tags, and a search bar with 'Go to file' and a 't' icon. Below the header, a commit from 'Calvin-Earl' titled 'midterm files' is shown, made 5 minutes ago. The commit details five files: 'roles' (midterm files), 'README.md' (Initial commit, 2 hours ago), 'ansible.cfg' (midterm files, 5 minutes ago), 'inventory.yaml' (midterm files, 5 minutes ago), and 'play.yml' (midterm files, 5 minutes ago). A link to the 'README' file is provided below the commit list.

CPE212_MIDTERM

GitHub link:

[Calvin-Earl/CPE212_MIDTERM](https://github.com/Calvin-Earl/CPE212_MIDTERM)

Conclusions: (link your conclusion from the objective)

In conclusion, I was able to apply my learnings in Automating Server Management by demonstrating how to install packages on separate hosts (Ubuntu, CentOS) while applying the concept of roles. In this midterm skills exam, I created an ansible playbook that installs a handful of packages and starts each of their services. While doing so, I also learned how to install packages out of my linux repository index, which is done by importing its repository in order for the system to recognize the packages, and adding its GPG key, which verifies the imported repository as a trusted repository. Overall, in this midterm skills exam, I was able to validate my knowledge in ansible by importing third-party repositories along with their GPG keys, install multiple packages from the imported repositories, and start each of their services while demonstrating the concept of roles.