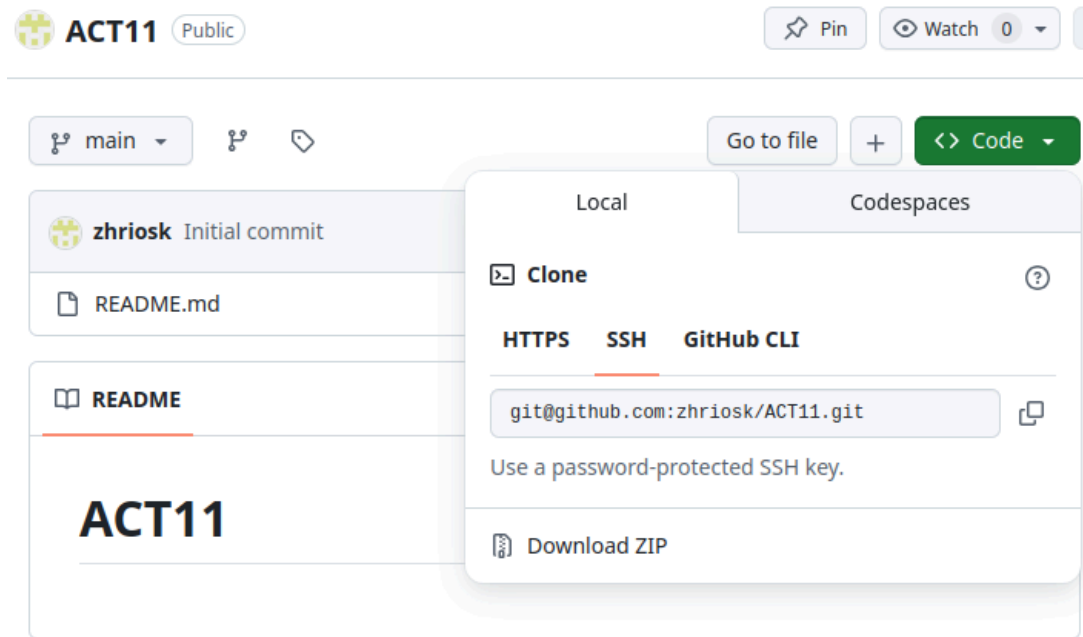


Name: Atian, Catherine Joy D.	Date Performed: Oct. 24, 2025
Course/Section: CPE 212 - CPE31S4	Date Submitted: Oct. 24, 2025
Instructor: Engr. Robin Valenzuela	Semester and SY: 1st SEM, A.Y. 2025 - 2026
Activity 11: Containerization	
1. Objectives	
Create a Dockerfile and form a workflow using Ansible as Infrastructure as Code (IaC) to enable Continuous Delivery process	
2. Discussion	
<p>Docker is an open platform for developing, shipping, and running applications. Docker enables you to separate your applications from your infrastructure so you can deliver software quickly. With Docker, you can manage your infrastructure in the same ways you manage your applications. By taking advantage of Docker's methodologies for shipping, testing, and deploying code quickly, you can significantly reduce the delay between writing code and running it in production.</p> <p>Source: https://docs.docker.com/get-started/overview/</p> <p>You may also check the difference between containers and virtual machines. Click the link given below.</p> <p>Source: https://docs.microsoft.com/en-us/virtualization/windowscontainers/about/containers-vs-vm</p>	
3. Tasks	
<ol style="list-style-type: none"> 1. Create a new repository for this activity. 2. Install Docker and enable the docker socket. 3. Add to Docker group to your current user. 4. Create a Dockerfile to install web and DB server. 5. Install and build the Dockerfile using Ansible. 6. Add, commit and push it to your repository. 	

4. Output (screenshots and explanations)



```
atian@ATIAN-Workstation:~$ git clone git@github.com:zhriosk/ACT11.git
Cloning into 'ACT11'...
remote: Enumerating objects: 3, done.
remote: Counting objects: 100% (3/3), done.
remote: Total 3 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
Receiving objects: 100% (3/3), done.
atian@ATIAN-Workstation:~$ ls
ACT11          Desktop      id_rsa       Music        Templates
ATIAN_PrelimExam Documents    id-rsa.pub   Pictures     Videos
CPE212         Downloads    id-rsa.pub   Public
CPE_MIDTERM_ATIAN id-rsa      MIDTERM_EXAM snap
atian@ATIAN-Workstation:~$
```

```

atian@ATIAN-Workstation:~/ACT11$ ansible-playbook --ask-become-pass -i inventory.ini -K install-doc
ker.yml
BECOME password:

PLAY [Install Docker on all nodes] *****

TASK [Gathering Facts] *****
ok: [server1]
ok: [server2]
ok: [CentOS]

TASK [Update APT package cache] *****
[WARNING]: Updating cache and auto-installing missing dependency: python3-apt
fatal: [CentOS]: FAILED! => {"changed": false, "cmd": "apt-get update", "msg": "[Errno 2] No such f
ile or directory: b'apt-get'", "rc": 2, "stderr": "", "stderr_lines": [], "stdout": "", "stdout_lin
es": []}
changed: [server2]
changed: [server1]

TASK [Install prerequisite packages] *****
ok: [server2]
ok: [server1]

TASK [Add Docker's official GPG key] *****
ok: [server2]
ok: [server1]

TASK [Add Docker repository] *****
ok: [server2]
ok: [server1]

TASK [Install Docker Engine] *****
ok: [server2]
fatal: [server1]: FAILED! => {"cache_update_time": 1761290036, "cache updated": true, "changed": fa
lse, "msg": "'/usr/bin/apt-get -y -o \"Dpkg::Options::=-force-confdef\" -o \"Dpkg::Options::=-for
ce-confold\"      install 'docker-ce=5:28.5.1-1~ubuntu.24.04~noble' failed: E: Could not get lock
/var/lib/dpkg/lock-frontent. It is held by process 16045 (apt-get)\nE: Unable to acquire the dpkg
frontend lock (/var/lib/dpkg/lock-frontent), is another process using it?\n", "rc": 100, "stderr":
"E: Could not get lock /var/lib/dpkg/lock-frontent. It is held by process 16045 (apt-get)\nE: Unabl

```

```

TASK [Install Docker SDK for Python] *****
fatal: [server2]: FAILED! => {"changed": false, "msg": "Unable to find any of pip3 to use. pip nee
ds to be installed."}

PLAY RECAP *****
CentOS                : ok=1    changed=0    unreachable=0    failed=1    skipped=0    rescued=
=0    ignored=0
server1               : ok=5    changed=1    unreachable=0    failed=1    skipped=0    rescued=
=0    ignored=0
server2               : ok=6    changed=1    unreachable=0    failed=1    skipped=0    rescued=
=0    ignored=0

atian@ATIAN-Workstation:~/ACT11$ docker --version
Command 'docker' not found, but can be installed with:
sudo snap install docker           # version 28.1.1+1, or
sudo apt install docker.io         # version 28.2.2-0ubuntu1~24.04.1
sudo apt install podman-docker     # version 4.9.3+ds1-1ubuntu0.2
See 'snap info docker' for additional versions.
atian@ATIAN-Workstation:~/ACT11$

```

```
atian@ATIAN-Workstation:~/ACT11$ ansible-playbook --ask-become-pass -i inventory.ini -K install-docker.yml
```

```
PLAY [Install Docker on all nodes] *****
```

```
TASK [Gathering Facts] *****
```

```
ok: [server1]
```

```
ok: [server2]
```

```
ok: [CentOS]
```

```
TASK [Update APT package cache] *****
```

```
[WARNING]: Updating cache and auto-installing missing dependency: python3-apt
```

```
fatal: [CentOS]: FAILED! => {"changed": false, "cmd": "apt-get update", "msg": "[Errno 2] No such file or directory: b'apt-get'", "rc": 2, "stderr": "", "stderr_lines": [], "stdout": "", "stdout_lines": []}
```

```
changed: [server2]
```

```
changed: [server1]
```

```
TASK [Ensure pip3 is installed] *****
```

```
ok: [server1]
```

```
ok: [server2]
```

```
TASK [Install Docker SDK for Python] *****
```

```
changed: [server2]
```

```
changed: [server1]
```

```
TASK [Install prerequisite packages] *****
```

```
ok: [server1]
```

```
ok: [server2]
```

```
TASK [Add Docker's official GPG key] *****
```

```
ok: [server2]
```

```
ok: [server1]
```

```
TASK [Add Docker repository] *****
```

```
ok: [server1]
```

```
ok: [server2]
```

```
TASK [Install Docker Engine] *****
```

```
Ln 6, Col 1 Spaces: 4
```

```
atian@ATIAN-Workstation:~/ACT11$ ansible-playbook --ask-become-pass -i inventory.ini -K install-docker.yml
```

```
ok: [server2]
```

```
TASK [Install Docker SDK for Python] *****
```

```
fatal: [server1]: FAILED! => {"changed": false, "cmd": ["/usr/bin/python3", "-m", "pip.__main__", "install", "docker"], "msg": "\n:stderr: error: externally-managed-environment\n\nx This environment is externally managed\n-> To install Python packages system-wide, try apt install\n  python3-xyz, where xyz is the package you are trying to\n  install.\n  \n  If you wish to install a non-Debian-packaged Python package,\n  create a virtual environment using python3 -m venv path/to/venv.\n  Then use path/to/venv/bin/python and path/to/venv/bin/pip. Make\n  sure you have python3 -full installed.\n  \n  If you wish to install a non-Debian packaged Python application,\n  it may be easiest to use pipx install xyz, which will manage a\n  virtual environment for you. Make sure you have pipx installed.\n  \n  See /usr/share/doc/python3.12/README.venv for more information.\n\nnote: If you believe this is a mistake, please contact your Python installation or OS distribution provider. You can override this, at the risk of breaking your Python installation or OS, by passing --break-system-packages.\n\nhint: See PEP 668 for the detailed specification.\n"}
fatal: [server2]: FAILED! => {"changed": false, "cmd": ["/usr/bin/python3", "-m", "pip.__main__", "install", "docker"], "msg": "\n:stderr: error: externally-managed-environment\n\nx This environment is externally managed\n-> To install Python packages system-wide, try apt install\n  python3-xyz, where xyz is the package you are trying to\n  install.\n  \n  If you wish to install a non-Debian-packaged Python package,\n  create a virtual environment using python3 -m venv path/to/venv.\n  Then use path/to/venv/bin/python and path/to/venv/bin/pip. Make\n  sure you have python3 -full installed.\n  \n  If you wish to install a non-Debian packaged Python application,\n  it may be easiest to use pipx install xyz, which will manage a\n  virtual environment for you. Make sure you have pipx installed.\n  \n  See /usr/share/doc/python3.12/README.venv for more information.\n\nnote: If you believe this is a mistake, please contact your Python installation or OS distribution provider. You can override this, at the risk of breaking your Python installation or OS, by passing --break-system-packages.\n\nhint: See PEP 668 for the detailed specification.\n"}
```

```
PLAY RECAP *****
```

```
CentOS : ok=1 changed=0 unreachable=0 failed=1 skipped=0 rescued=0 ignored=0
server1 : ok=8 changed=2 unreachable=0 failed=1 skipped=0 rescued=0 ignored=0
server2 : ok=8 changed=2 unreachable=0 failed=1 skipped=0 rescued=0 ignored=0
```

```
atian@ATIAN-Workstation:~/ACT11$
```

Ln 6, Col 1 Spaces: 4 UTF-

GNU nano 7.2

install-docker.yml

```
- name: Install Docker on all nodes
  hosts: all
  become: yes
  tasks:
    - name: Update APT package cache
      apt:
        update_cache: yes

    - name: Ensure pip3 is installed
      apt:
        name: python3-pip
        state: present

    - name: Install Docker SDK for Python
      ansible.builtin.shell: python3 -m pip install docker --break-system-packages

    - name: Install prerequisite packages
      apt:
        name:
          - apt-transport-https
          - ca-certificates
          - curl
          - software-properties-common
        state: present

    - name: Add Docker's official GPG key
      ansible.builtin.apt_key:
        url: https://download.docker.com/linux/ubuntu/gpg
        state: present

    - name: Add Docker repository
      ansible.builtin.apt_repository:
        repo: deb [arch=amd64] https://download.docker.com/linux/ubuntu {{ ansible_distribution_re
```

```
GNU nano 7.2                                install-docker.yml
state: present

- name: Add Docker's official GPG key
  ansible.builtin.apt_key:
    url: https://download.docker.com/linux/ubuntu/gpg
    state: present

- name: Add Docker repository
  ansible.builtin.apt_repository:
    repo: deb [arch=amd64] https://download.docker.com/linux/ubuntu {{ ansible_distribution_re
    state: present

- name: Install Docker Engine
  apt:
    name: docker-ce
    state: present
    update_cache: yes

- name: Install Docker SDK for Python
  ansible.builtin.pip:
    name: docker
    state: present

- name: Ensure Docker service is started and enabled on boot
  ansible.builtin.service:
    name: docker
    state: started
    enabled: yes

- name: Add current user to docker group
  user:
    name: "{{ ansible_user }}"
    groups: docker
    append: yes
```

Github Link:

<https://github.com/zhriosk/ACT11#>

Reflections:

Answer the following:

1. What are the benefits of implementing containerizations?

- Containerization offers a streamlined way to develop, deploy, and manage applications by packaging them with all necessary dependencies into isolated units. This ensures consistent performance across different environments, simplifies deployment, and speeds up delivery cycles. Containers are lightweight compared to virtual machines, making them more efficient and scalable. They also enhance security through process isolation and support modular architectures like microservices, allowing teams to update or scale components independently.

Conclusions:

In this activity, I was able to do the Docker and