

DevOps Assignment - 2

Ansible Playbook Exercise

Set Up:

- Installing ansible.
- Creating a remote server using AWS EC2 Instance.
Chosen platform: Red Hat Enterprise Linux with High Availability.
- Connected to the server using ssh client.

Step 1: Configuring Git login

Using username and security token.

- `https://user:token@github.com/path`

Step 2: Creating Ansible vault to Store the Git username and token

Creating a vault and setting a vault password and creating a file called `secrets.yml` that stores the github username and security token.

The `secrets.yml` file will be encrypted.

```
Sep 21 22:44
bhavya@bhavya-Inspiron-5570: ~/Desktop/Assignment2
bhavya@bhavya-Inspiron-5570:~/Desktop/Assignment2$ ansible-vault create secrets.yml
New Vault password:
Confirm New Vault password:
bhavya@bhavya-Inspiron-5570:~/Desktop/Assignment2$ cat secrets.yml
$ANSIBLE_VAULT;1.1;AES256
3235333530396666646463396366343232666639306362306661373262663265646264663464633
0
3838623762616435386166343163336539326231306636330a39666635623639656536363433343
5
3633386462363164303334333662633037326333613434356265313630313135396666336635663
6
6663656631383463340a36656232393132323963343861383933333264633466306637363136666
4
3231333231343733336662343135646332633132323139356438356262306532373965376463306
3
3635653432333131323935316161353336333831656464393863383965613231346233393763353
1
6536326263386638653361356262626462363038646436643561613562353737393331336336653
7
65633334366632306533
bhavya@bhavya-Inspiron-5570:~/Desktop/Assignment2$ echo "ENCRYPTED"
ENCRYPTED
```

We can decrypt the file using the following commands:

```
Sep 21 22:48
bhavya@bhavya-Inspiron-5570: ~/Desktop/Assignment2
bhavya@bhavya-Inspiron-5570:~/Desktop/Assignment2$ ansible-vault decrypt secrets.yml
Vault password:
Decryption successful
bhavya@bhavya-Inspiron-5570:~/Desktop/Assignment2$ cat secrets.yml
gituser: Bhavya-Tripathi
gitpass: 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
bhavya@bhavya-Inspiron-5570:~/Desktop/Assignment2$ echo "DECRYPTED"
DECRYPTED
bhavya@bhavya-Inspiron-5570:~/Desktop/Assignment2$
```

Step 3: The Ansible Git Example Playbook

Here we have created a nodejs app and uploaded it in a private github repository.

Then we create the Ansible playbook. (gitexample.yml)

The github username and token we created in the secrets.yml file.

```

- name: Install and Launch the Simple NodeJS Application
  hosts: nodeserver
  vars_files:
    - secrets.yml
  vars:
    - destdir: /apps/SampleNodeApp
  tasks:

    - name : install Node and NPM
      become: yes
      register: ymrepo
      yum:
        name: nodejs
        state: latest

    - name : validate the nodejs installation
      debug: msg="Installation of node is Successfull"
      when: ymrepo is changed

    - name: Version of Node and NPM
      shell:
        "npm -v && node -v"
      register: versioninfo

    - name: Version Info
      debug:
        msg: "Version info {{ versioninfo.stdout_lines }}"
      when: versioninfo is changed

    - name: Download the NodeJS code from the GitRepo
      become: yes
      git:
        repo: 'https://{{gituser}}:{{gitpass}}@github.com/Bhavya-Tripathi/DevOpsAssignment2.git'
        dest: "{{ destdir }}"

    - name: Change the ownership of the directory
      become: yes
      file:
        path: "{{destdir}}"
        owner: "ec2-user"
      register: chgrpout

    - name: Install Dependencies with NPM install command
      shell:
        "npm install"
      args:
        chdir: "{{ destdir }}"
      register: npminstlout

    - name: Debug npm install command
      debug: msg='{{npminstlout.stdout_lines}}'

    - name: Start the App
      async: 10
      poll: 0
      shell:
        "(node index.js > nodesrv.log 2>&1 &)"
      args:
        chdir: "{{ destdir }}"
      register: appstart

    - name: Validating the port is open
      tags: nodevalidate
      wait_for:
        host: "localhost"
        port: 3002
        delay: 10
        timeout: 30
        state: started
        msg: "NodeJS server is not running"

```

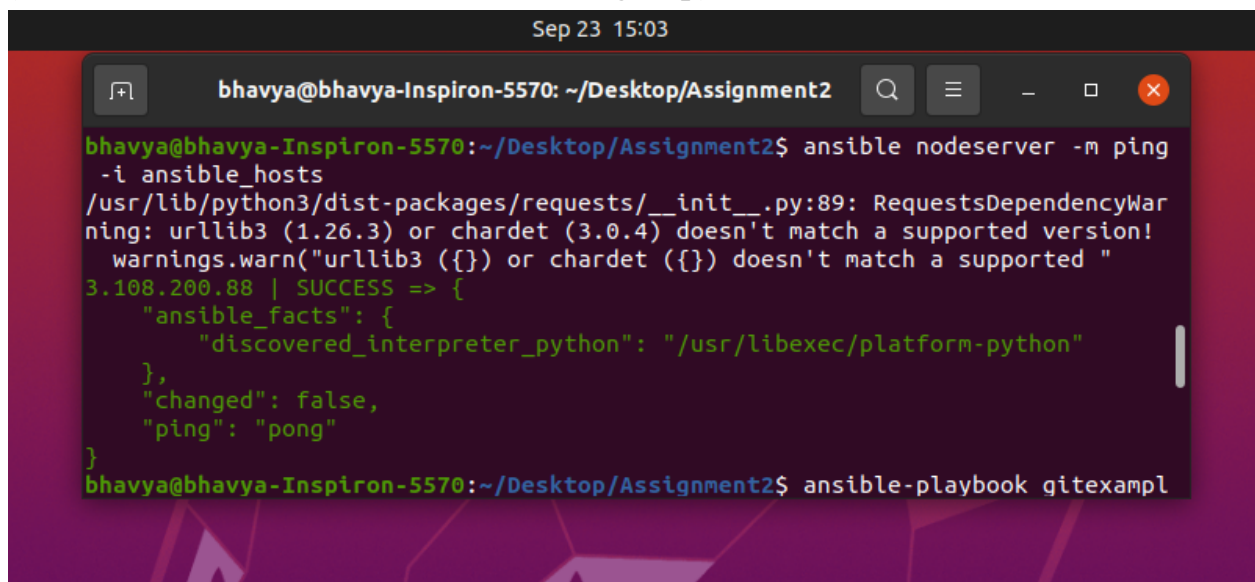
Step 4: Launch the Playbook with Ansible Git

Now we launch the playbook using the `ansible-playbook` command

```
ansible-playbook gitexample.yml --ask-vault-pass
```

Method:

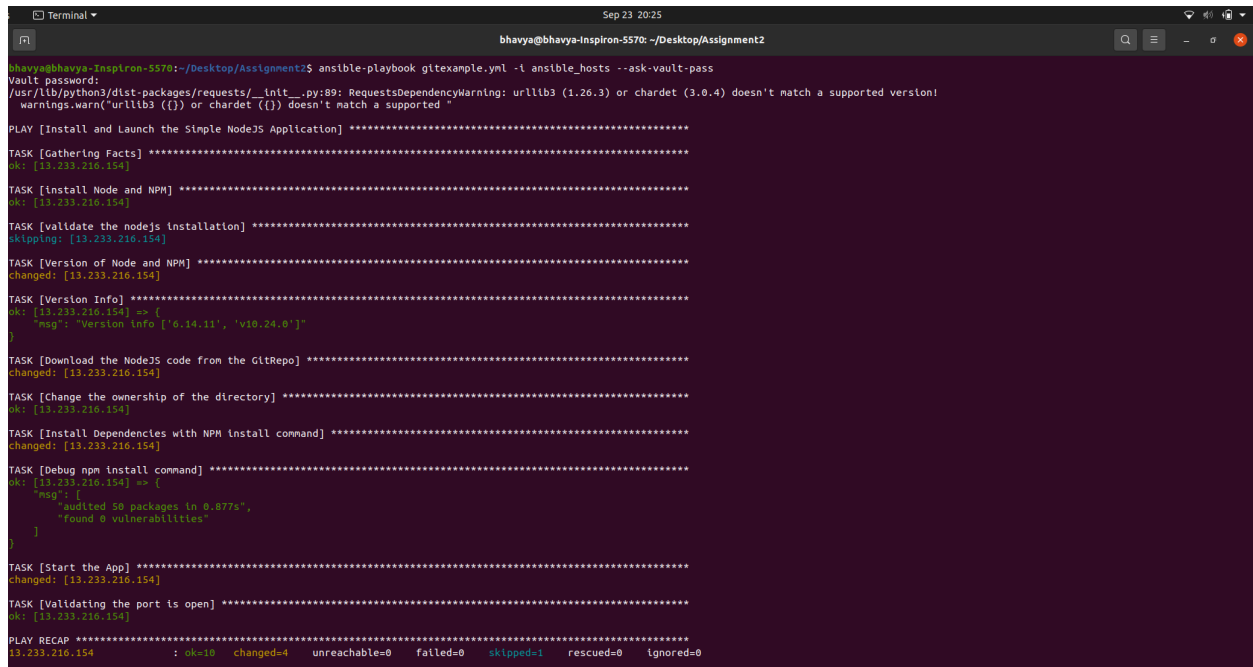
First we check if the hostgroup is reachable using the following command:
(here, `nodeserver` is the name of our hostgroup)

A terminal window titled "bhavya@bhavya-Inspiron-5570: ~/Desktop/Assignment2" with a timestamp "Sep 23 15:03". The terminal shows the command `ansible nodeserver -m ping -i ansible_hosts` being executed. The output includes a warning about urllib3 and chardet versions, followed by a successful ping result for 3.108.200.88. The JSON output shows "ansible_facts" with "discovered_interpreter_python" and "ping" as "pong". The prompt then changes to `ansible-playbook gitexampl`.

```
Sep 23 15:03
bhavya@bhavya-Inspiron-5570: ~/Desktop/Assignment2
bhavya@bhavya-Inspiron-5570:~/Desktop/Assignment2$ ansible nodeserver -m ping
-i ansible_hosts
/usr/lib/python3/dist-packages/requests/__init__.py:89: RequestsDependencyWar
ning: urllib3 (1.26.3) or chardet (3.0.4) doesn't match a supported version!
  warnings.warn("urllib3 ({}), or chardet ({}), doesn't match a supported "
3.108.200.88 | SUCCESS => {
  "ansible_facts": {
    "discovered_interpreter_python": "/usr/libexec/platform-python"
  },
  "changed": false,
  "ping": "pong"
}
bhavya@bhavya-Inspiron-5570:~/Desktop/Assignment2$ ansible-playbook gitexampl
```

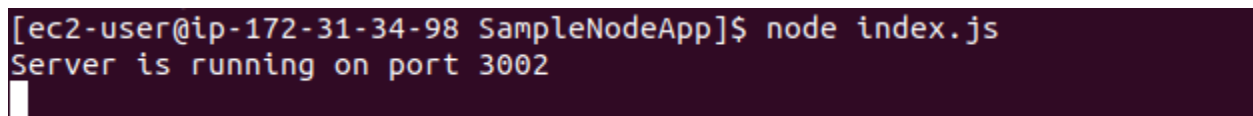
`Ansible_hosts` is an inventory file that contains the name of the host group, public IP address of the host server etc.

Now we launch the playbook:

A terminal window titled "Terminal" with a subtitle "bhavya@bhavya-Inspiron-5570: ~/Desktop/Assignment2". The terminal shows the command `ansible-playbook gitexample.yml -i ansible_hosts --ask-vault-pass` being executed. It prompts for a vault password, which is entered as `/usr/lib/python3/dist-packages/requests/_init_.py:89: RequestsDependencyWarning: urllib3 (1.26.3) or chardet (3.0.4) doesn't match a supported version!`. The output shows the playbook "Install and Launch the Simple NodeJS Application" running successfully. Tasks include: "Gathering Facts", "Install Node and NPM", "validate the nodejs installation" (skipped), "Version of Node and NPM" (changed), "Version Info" (ok), "Download the NodeJS code from the GitRepo" (changed), "Change the ownership of the directory" (ok), "Install Dependencies with NPM (install command)" (changed), "Debug npm install command" (ok), "Start the App" (changed), and "Validating the port is open" (ok). The final output is `PLAY RECAP` showing 10 ok, 4 changed, 0 unreachable, 0 failed, 1 skipped, 0 rescued, and 0 ignored.

The playbook ran successfully.

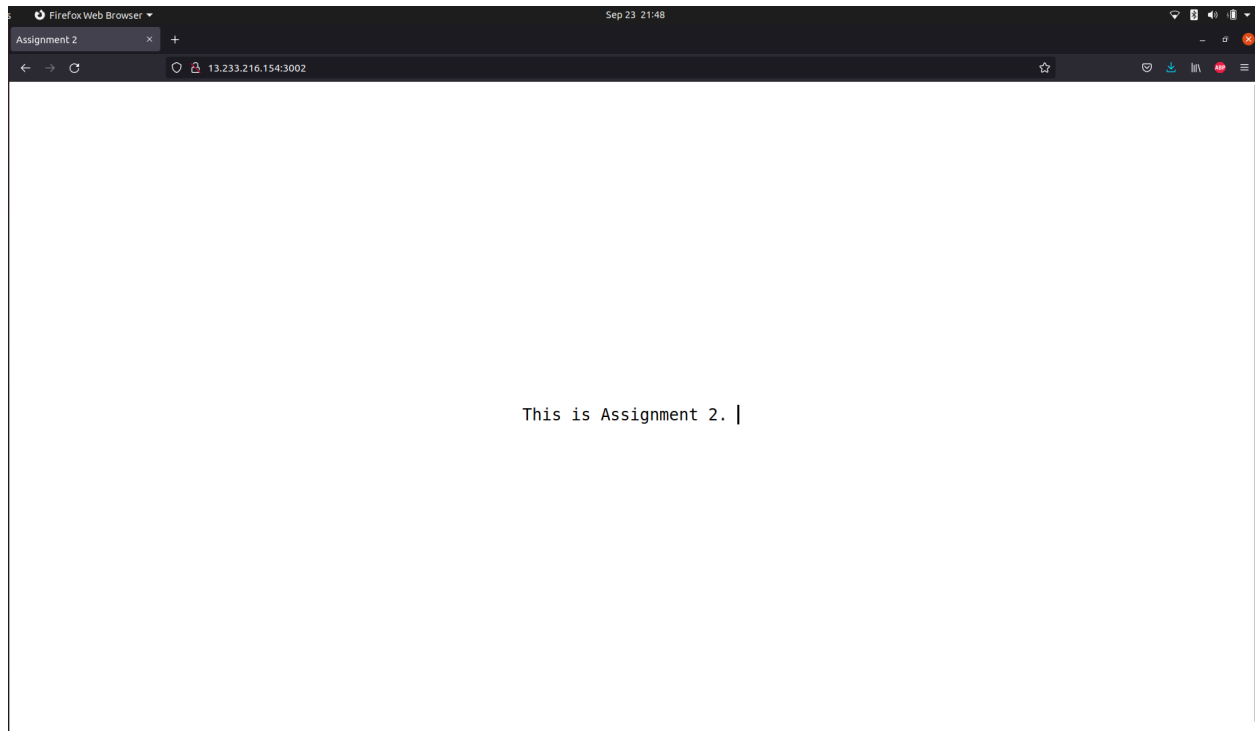
This means our github repository has been cloned to our server in the `/apps/SampleNodeApp` directory.

A terminal window showing the command `node index.js` being executed in the `SampleNodeApp` directory. The output is `Server is running on port 3002`.

The server is running successfully!

Step 5: Validate the Deployment

The remote server here is 13.233.216.154, now we can access the URL via <http://13.233.216.154:3002>.



Our Node Website has been deployed successfully!