Санкт-Петербургское государственное бюджетное
профессиональное образовательное учреждение
«Радиотехнический колледж»

ОТЧЕТ

по практической работе №10

на тему: «Применение Ansible»

по учебной практике УП.03 Эксплуатация объектов сетевой инфраструктуры

специальность 09.02.06 Сетевое и системное администрирование

Выполнил: студент группы С1-21
Авраменко Кирилл Владимирович

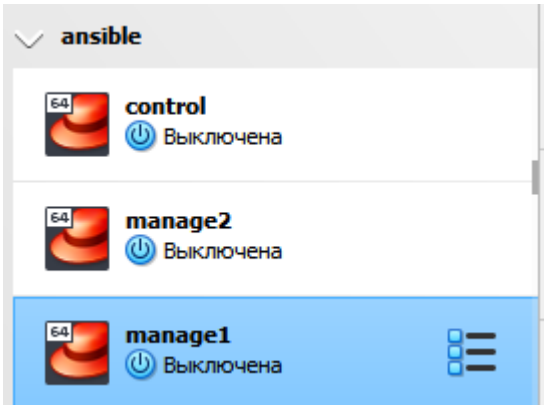
Проверил: преподаватель
Дубровин Виталий Александрович

Санкт-Петербург
2025

Цель работы: Отработка практических навыков

Осваиваемые компетенции: ПК 3.1, ПК 3.2, ПК 3.3, ПК 3.4, ПК 3.5, ПК 3.6.

Задание.



1. Произвести установку и настройку стека LAMP с помощью Ansible

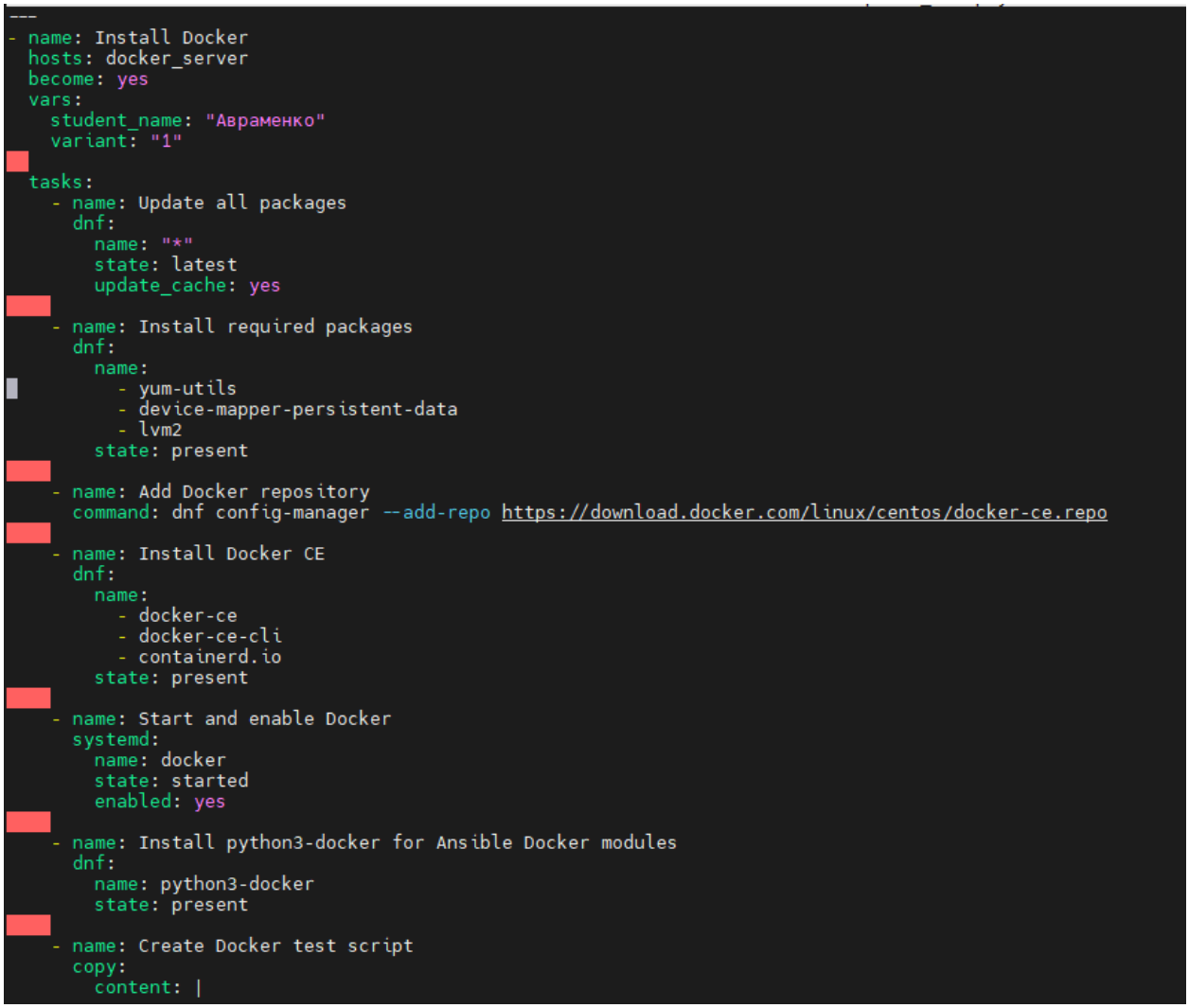
```yaml
---
- name: Install LAMP Stack
  hosts: all
  become: yes
  vars:
    student_name: "Авраменко"
    variant: "1"
    mysql_root_password: "secure_password"

  tasks:
    - name: Update all packages
      dnf:
        name: "*"
        state: latest
        update_cache: yes

    - name: Install Apache
      dnf:
        name: httpd
        state: present

    - name: Start and enable Apache
      systemd:
        name: httpd
        state: started
        enabled: yes

    - name: Install MariaDB
      dnf:
        name:
          - mariadb
          - mariadb-server
        state: present

    - name: Start and enable MariaDB
      systemd:
        name: mariadb
        state: started
        enabled: yes

    - name: Install PHP and required modules
      dnf:
        name:
          - php
          - php-mysqlnd
          - php-json
          - php-gd
          - php-mbstring
          - php-xml
          - php-zip
        state: present
```

- name: Install LAMP Stack
  hosts: lamp_server
  become: yes
  vars:

```
  student_name: "Авраменко"
  variant: "1"
  mysql_root_password: "secure_password"

tasks:
  - name: Update all packages
    dnf:
      name: "*"
      state: latest
      update_cache: yes

  - name: Install Apache
    dnf:
      name: httpd
      state: present

  - name: Start and enable Apache
    systemd:
      name: httpd
      state: started
      enabled: yes

  - name: Install MariaDB
    dnf:
      name:
        - mariadb
        - mariadb-server
      state: present

  - name: Start and enable MariaDB
    systemd:
      name: mariadb
      state: started
      enabled: yes

  - name: Install PHP and required modules
    dnf:
      name:
```

```
      - php
      - php-mysqlnd
      - php-json
      - php-gd
      - php-mbstring
      - php-xml
      - php-zip
    state: present

- name: Restart Apache
  systemd:
    name: httpd
    state: restarted

- name: Create info.php file
  copy:
    content: |
      <?php
      // Created by {{ student_name }}, Variant {{ variant }}
      phpinfo();
      ?>
    dest: /var/www/html/info.php
    owner: apache
    group: apache
    mode: '0644'

- name: Set firewall rules
  firewalld:
    service: "{{ item }}"
    permanent: yes
    state: enabled
  loop:
    - http
    - https

- name: Reload firewall
  command: firewall-cmd --reload
```

2. Произвести установку и настройку docker с помощью Ansible



```
---
- name: Install Docker
  hosts: docker_server
  become: yes
  vars:
    student_name: "Авраменко"
    variant: "1"

  tasks:
    - name: Update all packages
      dnf:
        name: "*"
        state: latest
        update_cache: yes

    - name: Install required packages
      dnf:
        name:
          - yum-utils
          - device-mapper-persistent-data
          - lvm2
        state: present

    - name: Add Docker repository
      command: dnf config-manager --add-repo https://download.docker.com/linux/centos/docker-ce.repo

    - name: Install Docker CE
      dnf:
        name:
          - docker-ce
          - docker-ce-cli
          - containerd.io
        state: present

    - name: Start and enable Docker
      systemd:
        name: docker
        state: started
        enabled: yes

    - name: Install python3-docker for Ansible Docker modules
      dnf:
        name: python3-docker
        state: present

    - name: Create Docker test script
      copy:
        content: |
```

- name: Install Docker
  hosts: docker_server
  become: yes
  vars:
    student_name: "Авраменко"
    variant: "1"

  tasks:
    - name: Update all packages
      dnf:
        name: "*"
        state: latest
        update_cache: yes

```
  - name: Install required packages
    dnf:
      name:
        - yum-utils
        - device-mapper-persistent-data
        - lvm2
      state: present

  - name: Add Docker repository
    command: dnf config-manager --add-repo https://download.docker.com/linux/centos/docker-ce.repo

  - name: Install Docker CE
    dnf:
      name:
        - docker-ce
        - docker-ce-cli
        - containerd.io
      state: present

  - name: Start and enable Docker
    systemd:
      name: docker
      state: started
      enabled: yes

  - name: Install python3-docker for Ansible Docker modules
    dnf:
      name: python3-docker
      state: present

  - name: Create Docker test script
    copy:
      content: |
```

```
#!/bin/bash
          # Created by {{ student_name }}, Variant {{ variant }}
          docker run --rm hello-world
        dest: /root/test-docker.sh
        owner: root
        group: root
        mode: '0755'

    - name: Run test container
      command: /root/test-docker.sh
      register: docker_test

    - name: Show test result
      debug:
        var: docker_test.stdout_lines
```

3. Произвести установку и настройку стека LAMP + WordPress с помощью Ansible

```
---
- name: Install LAMP and WordPress
  hosts: wordpress_server
  become: yes
  vars:
    student_name: "Авраменко"
    variant: "1"
    mysql_root_password: "secure_password"
    wordpress_db_name: "wordpress"
    wordpress_db_user: "wpuser"
    wordpress_db_password: "wppassword"

  tasks:
    - name: Update all packages
      dnf:
        name: "*"
        state: latest
        update_cache: yes

    - name: Install Apache
      dnf:
        name: httpd
        state: present

    - name: Start and enable Apache
      systemd:
        name: httpd
        state: started
        enabled: yes

    - name: Install MariaDB
      dnf:
        name:
          - mariadb
          - mariadb-server
        state: present

    - name: Start and enable MariaDB
      systemd:
        name: mariadb
        state: started
        enabled: yes

    - name: Install PHP and required modules
      dnf:
        name:
          - php
          - php-mysqlnd
          - php-json
          - php-gd
```

```
        - php-mbstring
        - php-xml
        - php-zip
      state: present

  - name: Restart Apache
    systemd:
      name: httpd
      state: restarted

  - name: Create WordPress database
    mysql_db:
      name: "{{ wordpress_db_name }}"
      state: present
      login_unix_socket: /var/lib/mysql/mysql.sock
vars:
 ansible_python_interpreter: /usr/bin/python3
      ansible_module_defaults:
        mysqld_db:
          connect_params:
          db_driver: pymsql
  - name: Create WordPress database user
    mysql_user:
      name: "{{ wordpress_db_user }}"
      password: "{{ wordpress_db_password }}"
      priv: "{{ wordpress_db_name }}.*:ALL"
      host: localhost
      state: present
      login_unix_socket: /var/lib/mysql/mysql.sock
    vars:
      ansible_python_interpreter: /usr/bin/python3
      ansible_module_defaults:
        mysql_user:
          connect_params:
          db_driver: pymsql

  - name: Create WordPress database user
    mysql_user:
      name: "{{ wordpress_db_user }}"
      password: "{{ wordpress_db_password }}"
      priv: "{{ wordpress_db_name }}.*:ALL"
      host: localhost
      state: present
      login_unix_socket: /var/lib/mysql/mysql.sock

  - name: Download WordPress
    get_url:
      url: https://wordpress.org/latest.tar.gz
      dest: /tmp/wordpress.tar.gz

  - name: Create WordPress directory
    file:
```

```
      path: /var/www/html/wordpress
      state: directory
      owner: apache
      group: apache

  - name: Extract WordPress
    unarchive:
      src: /tmp/wordpress.tar.gz
      dest: /tmp/
      remote_src: yes

  - name: Move WordPress files
    shell: cp -r /tmp/wordpress/* /var/www/html/wordpress/

  - name: Set ownership
    file:
      path: /var/www/html/wordpress
      owner: apache
      group: apache
      recurse: yes

  - name: Create wp-config
    template:
      src: wp-config.php.j2
      dest: /var/www/html/wordpress/wp-config.php
      owner: apache
      group: apache

  - name: Set firewall rules
    firewalld:
      service: "{{ item }}"
      permanent: yes
      state: enabled
    loop:
      - http
      - https

- name: Install Python MySQL module
    dnf:
 name:
       - python3-PyMySQL
      state: present
```

```yaml
---
- name: Install LAMP and WordPress
  hosts: wordpress_server
  become: yes
  vars:
    student_name: "Авраменко"
    variant: "1"
    mysql_root_password: "secure_password"
    wordpress_db_name: "wordpress"
    wordpress_db_user: "wpuser"
    wordpress_db_password: "wppassword"

  tasks:
    - name: Update all packages
      dnf:
        name: "*"
        state: latest
        update_cache: yes

    - name: Install Apache
      dnf:
        name: httpd
        state: present

    - name: Start and enable Apache
      systemd:
        name: httpd
        state: started
        enabled: yes

    - name: Install MariaDB
      dnf:
        name:
          - mariadb
          - mariadb-server
        state: present

    - name: Start and enable MariaDB
      systemd:
        name: mariadb
        state: started
        enabled: yes

    - name: Install PHP and required modules
      dnf:
        name:
          - php
          - php-mysqlnd
          - php-json
          - php-gd
          - php-mbstring
          - php-xml
```

Вывод: Отработал практические навыки по ansible