



O T U S

Python QA Engineer

otus.ru

Меня хорошо видно & слышно?



Защита проекта

Тема: Проект по тестированию VPN



Седреева Галина

Должность Инженер

Компания "ООО С-Терра Си Эс Пи"

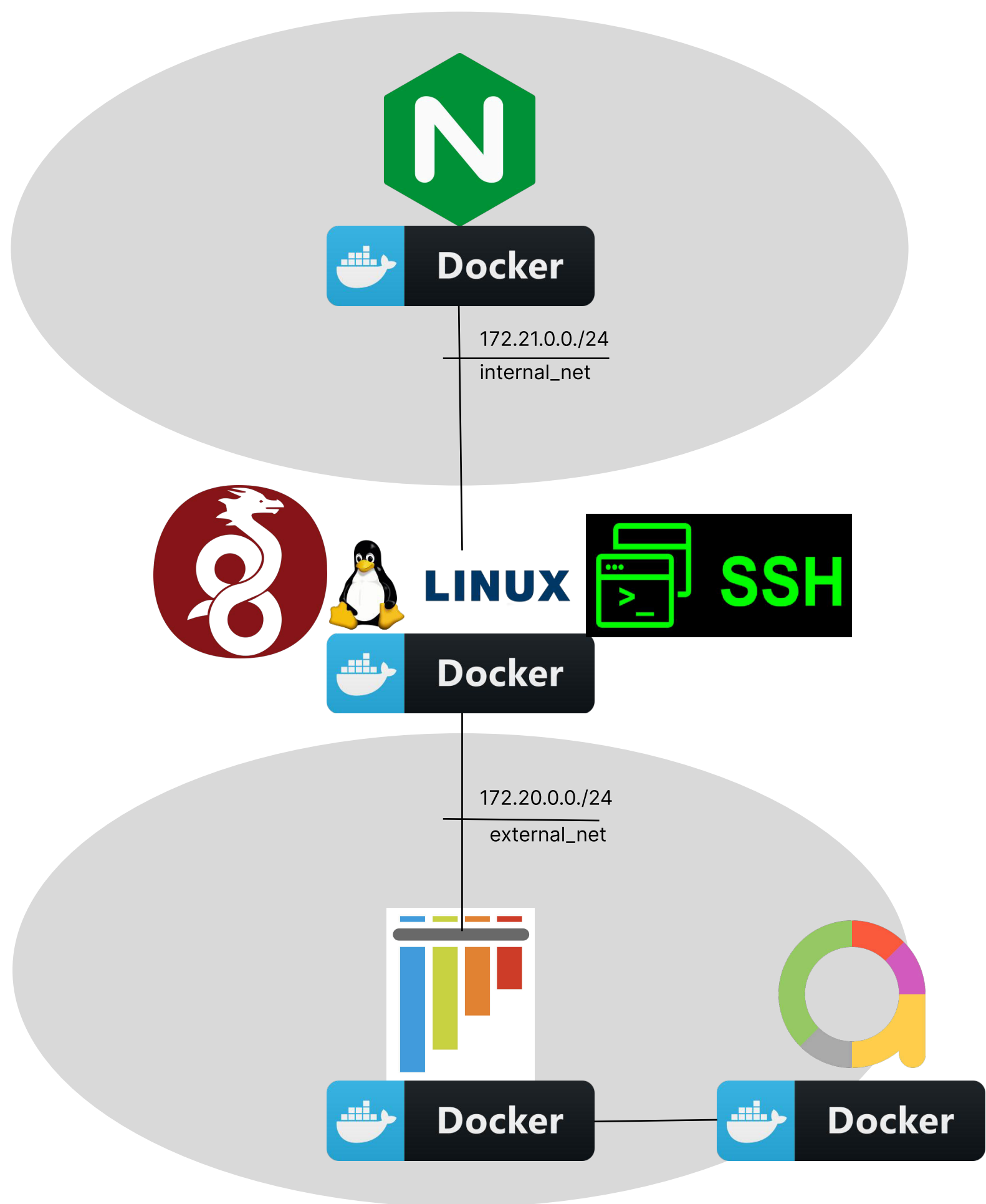
Цели проекта

1. Выполнять автоматическое тестирование с отчетностью в Allure.

2. Этапы выполнения проекта:

1. собрать стенд из docker-контейнеров и сетей между ними
 2. настроить в них VPN, SSH и nginx
 3. написать тесты
 4. настроить отчетность в Allure
-

3. Настройка серверов и клиентов происходила автоматически, безучастия инженера, в процессе развёртывания стенда, одной командой.



Используемые технологии

- **Docker Compose** — для разворачивания стенда.
- **WireGuard** — для организации VPN.
- **Nginx** — в роли тестового сервиса в защищённой сети.
- **Python / Pytest** — для написания автотестов.
- **Paramiko** — для удалённого выполнения команд на сервере.
- **Allure** — для генерации HTML-отчётов по результатам тестирования

Контейнер с клиентом

```

41  > Run Service
42  client:
43    image: linuxserver/wireguard:latest
44    container_name: client
45    environment:
46      - PUID=1000
47      - PGID=1000
48      - TZ=Europe/Moscow
49      - VPN_PROTOCOL=wireguard
50    cap_add:
51      - NET_ADMIN
52      - SYS_MODULE
53    volumes:
54      - ./wireguard/config:/config
55      - /lib/modules:/lib/modules
56      - ./tests:/tests
57      - ./allure-results:/allure-results
58    sysctls:
59      - net.ipv4.conf.all.src_valid_mark=1
60      - net.ipv4.ip_forward=1
61    networks:
62      external_net:
63        ipv4_address: 172.20.0.11
64    command:
65      sh -c "apk add --no-cache python3 py3-pip openssh-client curl wireguard-tools &&
66      pip3 install --break-system-packages pytest allure-pytest paramiko requests &&
67      mkdir -p /tests &&
68      cd /tests &&
69      pytest test_vpn.py --alluredir=/allure-results &&
70      allure generate ./allure-results -o ./allure-report --clean
71      tail -f /dev/null"
72    restart: unless-stopped

```

Контейнер с сервером

```
2 wireguard:
3   image: linuxserver/wireguard:latest
4   container_name: wireguard
5   cap_add:
6     - NET_ADMIN
7     - SYS_MODULE
8   environment:
9     - PUID=1000
10    - PGID=1000
11    - TZ=Europe/Moscow
12    - SERVERURL=wireguard
13    - SERVERPORT=51820
14    - PEERS=1
15    - PEERDNS=auto
16    - INTERNAL_SUBNET=10.13.13.0/24
17    - ALLOWEDIPS=10.13.13.0/24,172.21.0.0/24
18    - LOG_CONFS=true
19   volumes:
20     - ./wireguard/server:/config
21     - /lib/modules:/lib/modules:ro
22   ports:
23     - "51820:51820/udp"
24     - "2222:22"
25   sysctls:
26     - net.ipv4.conf.all.src_valid_mark=1
27     - net.ipv4.ip_forward=1
28   networks:
29     external_net:
30       ipv4_address: 172.20.0.10
31     internal_net:
32       ipv4_address: 172.21.0.10
33   command:
34     sh -c "apk add --no-cache openssh && \
35     ssh-keygen -A && \
36     mkdir -p /root/.ssh && \
37     sed -i 's/#PermitRootLogin prohibit-password/PermitRootLogin yes/' /etc/ssh/sshd_config && \
38     echo 'root:123' | chpasswd && \
39     /usr/sbin/sshd -D"
```


Фикстура

```
7 @pytest.fixture(scope="function")
8 def set_vpn_connection(request):
9     """Фикстура для установки нужного состояния VPN перед тестом."""
10    mode = request.param
11    wg_config_path = "/etc/wireguard/wg0.conf"
12
13    # Убедимся, что конфиг существует
14    if not os.path.exists(wg_config_path):
15        pytest.fail(f"Конфиг WireGuard не найден по пути {wg_config_path}")
16
17    def is_wireguard_active():
18        result = subprocess.run(["ip", "link", "show", "wg0"],
19                                stdout=subprocess.PIPE,
20                                stderr=subprocess.PIPE)
21        return result.returncode == 0
22
23    # Очищаем текущее состояние
24    if is_wireguard_active():
25        subprocess.run(["wg-quick", "down", "wg0"], check=False)
26
27    # Поднимаем соединение с сервером
28    if mode == "enabled":
29        subprocess.run(["chmod", "600", "/config/wg_confs/wg0.conf", wg_config_path], check=True)
30        subprocess.run(["wg-quick", "up", wg_config_path], check=True)
31        time.sleep(2) # даём время на установку соединения
32
33    yield mode
34
35    # Чистка после теста
36
37    if is_wireguard_active():
38        subprocess.run(["wg-quick", "down", "wg0"], check=False)
```

Примеры тестов

```
24 @allure.description("Тест 1: Проверяем, что без VPN web-сервер недоступен")
25 @pytest.mark.parametrize("set_vpn_connection", ["disabled"], indirect=True)
26 @allure.title("Проверка недоступности веб-сервера без VPN")
27 @allure.feature("Проверка доступности")
28 def test_webserver_unreachable_without_vpn(set_vpn_connection):
29
30     with allure.step("Проверяем, что веб-сервер недоступен"):
31         with pytest.raises(requests.exceptions.RequestException) as exc_info:
32             requests.get(WEB_SERVER_IP, timeout=R_TIMEOUT)
33
34         exception = exc_info.value
35         assert isinstance(exception, (
36             requests.exceptions.ConnectionError,
37             requests.exceptions.Timeout,
38             requests.exceptions.RetryError
39         )), f"Ожидалась ошибка подключения, получено: {type(exception).__name__}"
40
41
42 @allure.description("Тест 2: Запускаем VPN и проверяем, что клиент подключён")
43 @pytest.mark.parametrize("set_vpn_connection", ["enabled"], indirect=True)
44 @allure.title("Проверка подключения клиента через wg show")
45 @allure.feature("Соединение [ ] VPN")
46 def test_vpn_client_connection(set_vpn_connection):
47
48     with allure.step("Запрашиваем информацию [ ] состоянии WireGuard (wg show)"):
49         result = run_cmd("wg show")
50
51     with allure.step("Проверяем успешное выполнение команды 'wg show'"):
52         assert result.returncode == 0, f"Команда 'wg show' завершилась [ ] ошибкой. STDERR: {result.stderr}"
53
54     with allure.step("Проверяем наличие интерфейса wg0 и peer в выводе"):
55         assert WG_INTERFACE in result.stdout, "Интерфейс wg0 не найден в выводе 'wg show'"
56         assert "peer:" in result.stdout, "Поле 'peer:' отсутствует в выводе 'wg show'"
57
```

