

Лабораторная работа №3

Тема: Nginx-прокси во внутренней сети с двумя бэкендами

Студент: Чуев Никита Сергеевич (группа p4250)

Преподаватель: Некрасов Евгений Андреевич

Дисциплина: Системное администрирование

Дата: 2025 год

Цели и задачи работы

- Развернуть во «серой» сети стенд из трёх узлов: `proxy01`, `app01`, `app02`.
- Настроить **nginx** как reverse-proxy на `proxy01` (порт 80).
- Поднять два HTTP-бэкенда на `app01` и `app02` (порт 8080), возвращающие строку с идентификатором бэкенда.
- Реализовать балансировку запросов между бэкендами и JSON-логирование в `/var/log/nginx/access.json` с полем `upstream`.
- Ограничить сетевой доступ через `ufw`: бэкенды доступны только

Топология и подготовка хостов

Во внутренней сети используются три узла:

Хост	Роль	DNS-имя	IP-адрес
proxy01	reverse-proxy (nginx)	proxy01.dc.local	10.100.0.1
app01	backend #1 (HTTP)	app01.dc.local	10.100.0.2

Бэкенды (каталог `app/`)

На `app01` и `app02` развёрнут один и тот же Python-бэкенд из файла `app/app.py`:

```
sudo apt -y update && sudo apt -y install python3  
sudo install -d -o root -g root /opt/simple-backend  
sudo cp app/app.py /opt/simple-backend/app.py
```

Логика ответа в `app/app.py`:

Запуск бэкендов и настройка nginx

Запуск бэкендов как экземпляры юнита:

```
# На app01  
sudo systemctl enable --now simple-backend@app01  
  
# На app02  
sudo systemctl enable --now simple-backend@app02
```

Проверка с proxy01 (вывод в `checks/backend.txt`):

Сетевая безопасность (каталог `firewall/`)

Цель: бэкенды доступны только с `proxy01`, наружу открыт только `proxy01:80`.

Настройка `ufw` на `app01` и `app02`:

```
sudo apt -y install ufw
sudo ufw default deny incoming
sudo ufw default allow outgoing
```

Проверки и failover (каталог `checks/`)

На `proxy01` используется скрипт `checks/run_all.sh`:

```
chmod +x checks/run_all.sh  
./checks/run_all.sh
```

Скрипт формирует:

- `checks/dns.txt` – вывод `host app01/app02/proxy01`.
- `checks/backend.txt` – ответы: `curl -s http://app01.dc.local:8080/` и `curl -s`

Итоги и вопросы

- Развёрнут стенд с `proxy01`, `app01`, `app02` во внутренней сети.
- Бэкенды из `app/app.py` запущены через `app/systemd/simple-backend@.service` как `simple-backend@app01` и `@app02`.
- Nginx (конфиг `proxy/nginx.conf.d/app.conf`) работает как reverse-proxy с балансировкой и JSON-логированием.
- Через `ufw` реализованы сетевые ограничения, все проверки и failover задокументированы в каталоге `checks/`.