Описание работы

Развёртывание стенда из трёх узлов: proxy01, app01, app02.

Настройка Nginx на proxy01 как reverse-proxy (порт 80).

Два HTTP-бэкенда на app01 и app02 (порт 8080),

возвращающие идентификатор бэкенда.

Студент: Чуев Никита Сергеевич (группа p4250)

Цели и задачи

- Развернуть стенд во внутренней сети.
- Настроить балансировку между бэкендами.
- Реализовать JSON-логирование с полем upstream.
- Ограничить доступ к бэкендам через ufw.

Топология и подготовка хостов

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
app02	backend #2 (HTTP)	app02.dc.local	10.100.0.3

Бэкенды и Python

На app01 и app02 развёрнут один и тот же Python-бэкенд:

```
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` возвращает идентификатор бэкенда.

Запуск бэкендов

```
# На app01 sudo systemctl enable --now simple-

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

simple-backend@app02
```

Настройка Nginx

Конфигурация в `proxy/nginx.conf.d/app.conf`

- Балансировка запросов (round-robin)
- JSON-логирование в `/var/log/nginx/access.json`

Сетевая безопасность

- Бэкенды доступны только с proxy01
- UFW на app01/app02:

```
sudo ufw default deny incoming sudo ufw default allow
outgoing
```

Проверки и failover

Скрипт `checks/run_all.sh` генерирует:

- `checks/dns.txt` — вывод host
- `checks/backend.txt` — ответы curl

Итоги

- Стенд proxy01, app01, app02 развёрнут
- Бэкенды работают через systemd сервис `simple-backend@`
- Nginx как reverse-proxy с балансировкой и JSON-логами
- Сетевые ограничения выполнены через ufw
- Все проверки и failover задокументированы

Заключение

Лабораторная работа выполнена полностью.

Все цели достигнуты, сеть функционирует корректно, доступность бэкендов проверена.