Лабораторная работа

Тема: Автоматизация развёртывания HA-стенда в «серой» сети без внешнего интернета

Стек: `Nginx` + `keepalived (VRRP VIP)` + `ZooKeeper (3)` + `ClickHouse 3×3` + `nftables` + `Ansible`

Выполнили студенты: Чуев Никита, Ковалев Никита,
Выдумкин Илья

Группа: p4250

1. Цель и требования

Цель: развернуть отказоустойчивый стенд ClickHouse (3 шарда × 3 реплики) с единым входом через VIP и автоматическими проверками.

- **Офлайн:** без внешнего интернета, артефакты и шаблоны — в репозитории.
- **Идемпотентность:** повторный прогон Ansible → минимум изменений.
- **VRRP:** VIP мигрирует между двумя proxy при отказе.
- **Nginx LB:** балансировка по 9 CH-нодам + JSON access log

2. Топология стенда

Сеть: `10.10.0.0/24` • **VIP (VRRP):** `10.10.0.100` • домен: `dc.local`

Группа	Хосты	DNS-имя	IP
proxy	proxy01 / proxy02	`proxy01.dc.local`, `proxy02.dc.local`	`10.10.0.11`, `10.10.0.12`
vip	VRRP	`vip.dc.local`	`10.10.0.100`

3. Репозиторий и порядок запуска

Структура:

```
ansible/
  inventories/lab/hosts.ini
  inventories/lab/group_vars/*.yml
  roles/
    role_common
    role_dns
    role_firewall
```

Оркестрация (site.yml):

- `common` → базовые пакеты, ssh-
- `dns` → dnsmasq/hosts + resolv.c
- `fw` → nftables deny-by-default
- `zk` → ZooKeeper 3 ноды
- `ch` → ClickHouse 3×3 (rolling: `s`
- `nginx` → upstream на 9 нод
- `keepalived` → VRRP VIP
- `checks` → приемочные тесты

4. Офлайн-режим и базовая подготовка

- **Офлайн ClickHouse:** пакеты кладутся в `artifacts/clickhouse/deb/*.deb`, роль устанавливает их локально.
- **Bootstrap:** установка `python3`, `sudo`, диагностических утилит, настройка timezone/locale.
- **SSH-hardening:** `PermitRootLogin no`, `PasswordAuthentication no`, `AllowUsers admin`.
- **Безопасные проверки:** включение `fstrim.timer` только при наличии unit-файла.

5. DNS и firewall в серой сети

DNS (dnsmasq):

- локальная зона `dc.local`
- записи `host-record` генерируются из inventory
- `/etc/resolv.conf` направлен на внутренний DNS
- guard для cloud-init при правке `/etc/hosts`

Firewall (nftables): политика `drop`, разрешаем только нужное

- proxy: `80/tcp` + VRRP (proto 112) между proxy
- zk: `2181/2888/3888` из lab-сети
- ch: `8123/9000` из lab-сети
- логирование drop с rate-limit

6. ZooKeeper: кластер из 3 нод

Роль: `role_zookeeper` • **Порты:** `2181/2888/3888`

- установка `zookeeperd` на `zk01/zk02/zk03`
- генерация `zoo.cfg` с `server.1/2/3` и FQDN
- создание `myid` на каждой ноде (1/2/3 из inventory)
- systemd: enable + start
- healthcheck: `ruok → imok`

```
echo ruok | nc -w 2 zk01.dc.local 2181   # ожидаем: imok
```

7. ClickHouse: кластер 3×3 (3 шарда × 3 реплики)

- **Установка офлайн:** все `.deb` копируются на узел и ставятся локально через `apt`.
- **Интеграция с ZK:** генерируется `zookeeper.xml` со списком нод ZK.
- **Топология кластера:** `remote_servers.xml` описывает `cluster_3x3` (3×3).
- **Доступ:** `users.d/` создаёт HTTP-пользователя с паролем.
- **Демо DDL:** создаются таблицы `ReplicatedMergeTree` +

8. Nginx LB + VRRP VIP (keepalived)

Nginx (proxy01/proxy02):

- upstream на 9 нод CH
- `max_fails=2`, `fail_timeout=5s`, `keepalive 64`
- `proxy_next_upstream` для 502/503/504 + timeout
- JSON access.log: `ts`

Keepalived (VRRP):

- proxy01 = MASTER (priority 150)
- proxy02 = BACKUP (priority 100)
- VIP: `10.10.0.100/24`
- GARP для ускорения переключения
- notify-скрипты логируют смену роли

```
# лог смены роли

tail -n 50 /var/log/keepalived-vip.log
```

9. Проверки, отказоустойчивость и итог

make test запускает `role_checks` и сохраняет отчёты в `ansible/checks/`:

- `vip-failover.txt` — остановка keepalived на держателе VIP → проверка миграции
- `proxy-lb.txt` — 20 запросов через VIP (`SELECT hostName()`)
- `proxy-fail-node.txt` — остановка одной CH-ноды → запросы через VIP без 502/504 → возврат ноды
- `zk-quorum.txt` — `ruok/imok` по всем ZK
- `ch-health.txt` — `system.clusters`, `system.replicas`