Отказоустойчивый веб-хостинг

Вариант №1 (Docker + Terraform)

Ремизов Кирилл Львович

2026-06-28

VKCLOUD-80 | Итоговая аттестация

Задача проекта

Цель: Развернуть отказоустойчивую инфраструктуру для веб-приложения в облаке VK Cloud.

Требования:

- Два веб-сервера с балансировщиком нагрузки.
- Автоматическое переключение трафика при отказе одного из серверов.
- Вся инфраструктура — как код (Terraform).
- Автоматический деплой через CI/CD (GitHub Actions).
- Наличие управляемой базы данных в архитектуре.
- Бастион для безопасного SSH-доступа.

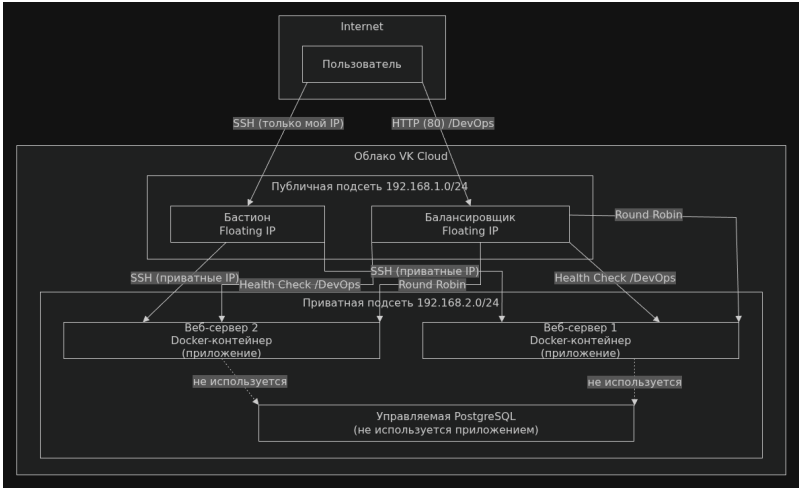
Что реализовано

- VPC с public/private подсетями (192.168.1.0/24, 192.168.2.0/24).
- Два веб-сервера с Docker-контейнерами (Flask-приложение).
- Балансировщик нагрузки Application L7 с health-check по ‘/DevOps‘.
- Бастион для SSH-доступа (доступ только с моего IP).
- Управляемая PostgreSQL (инстанс создан, база и пользователь готовы).
- Вся инфраструктура описана в Terraform, state хранится в S3.
- CI/CD через GitHub Actions (apply при пуше в main).
- Нагрузочное тестирование (‘ab -n 1000 -c 10‘) — 99.9% успешных.
- Отказоустойчивость продемонстрирована (остановка контейнера).

Что не реализовано

- **База данных** — создана, но *не используется* приложением.
  - Это сделано намеренно: по ТЗ БД должна присутствовать в структуре.
  - Подключение к БД проверено через ‘psql‘ (есть в видео).
  - В боевом окружении приложение можно легко интегрировать с БД.
- Health-check использует ‘/DevOps‘ (не выделенный ‘/health‘).
- Пароль БД захардкожен — в продакшене использовать переменные.
- Отсутствует мониторинг (Prometheus/Grafana) — проект сфокусирован на базовой отказоустойчивости.

Архитектура



- VPC: public (192.168.1.0/24) и private (192.168.2.0/24)
- Балансировщик + 2 веб-сервера (Docker, Flask)
- Бастион для SSH, управляемая PostgreSQL (задел)

Стек технологий

Компонент	Технология
IaC	Terraform (vkcs)
Контейнеры	Docker
Балансировка	VK Cloud LB (L7)
БД	PostgreSQL 14 (DBaaS)
CI/CD	GitHub Actions
Язык	Python + Flask + Gunicorn

Результаты тестирования

- Нагрузочный тест: `ab -n 1000 -c 10`
  - Успешных запросов: 999 (99.9%)
  - Среднее время: 130 мс
  - RPS: ~76.5
- Отказоустойчивость:
  - Остановка контейнера → трафик переключается на второй сервер
  - Восстановление через 10–15 секунд
- БД создана, подключение подтверждено через ‘psql‘

Доказательства

- Видеообзор (5 минут) — все ключевые этапы
- Логи: `ab`, `curl`, SSH-сессии
- Схема архитектуры и код

Все вышеперечисленное можно нати по ссылкам ниже:
`https://github.com/BlackCorbeau/final_assessment`
`https://git.rustprogersteam.ru/Black_Raven/MyResume`

Итоги

- Инфраструктура полностью описана кодом — воспроизводимость.
- Автоматизация через CI/CD.
- Отказоустойчивость работает — сайт доступен даже при сбое одного узла.
- БД присутствует как компонент, готова к интеграции.
- Проект соответствует всем критериям оценки.

Спасибо за внимание!