

Домашняя инфраструктура.

Обзор

Домашняя сеть объединяет три физические рабочие станции, ноутбук и два удалённых VPS. Вместе они образуют тестовый контур для разработки и одновременно выполняют роль медиacentра и сетевого хранилища.

Узлы

ws-001 – основная рабочая станция (Ubuntu 26.04.1 LTS).

Intel Core i3-2120 / 8 GB / SSD 120 GB (system), HDD 250 GB (documents).

Назначение: работа, учеба и разработка проектов.

Установлены: Python, VSCode, Chromium, Remmina, LibreOffice, Midnight Commander.

ws-002 – вторая рабочая станция (Windows 10 Pro).

Intel Core i3-540 / 4 GB / HDD 160 GB (system), HDD 500 GB (documents), 3 x HDD по 1 TB (видеоархив).

Назначение: медиacentр (вторым монитором подключен телевизор), задачи, требующие Windows.

Установлены: Delphi 12, MobaXterm, Far Manager, MSOffice, pgAdmin4, Photoshop, FineReader. К ней же подключены принтер и сканер.

nb-001 – ноутбук (Mint 22.3).

Intel Celeron N4100 / 4 GB / SSD 120 GB.

Назначение: мобильное устройство для работы, разработки.

Установлены: Python, VSCode, Chromium, Remmina, LibreOffice, Midnight Commander.

srv-001 – компьютер в роли сервера (Ubuntu 24.04.4 LTS).

Intel Pentium Dual-Core E2220 / 6 GB / три HDD и один SSD.

Каждый диск выделен под свою задачу:

- HDD 250 GB – разбит на два логических тома:
 - 60 GB (system) – операционная система;
 - 190 GB (data) – серверные приложения: PostgreSQL, Redis, Gitea, Docker.
- HDD 500 GB (storage) – сетевое хранилище; через Samba подключено ко всем устройствам сети (компьютеры, ноутбук, смартфон).
- SSD 120 GB (VM) – полностью отдан под виртуализацию.
- HDD 120 GB (backup) – резервное копирование базы данных, скриптов и виртуальных машин.

Этот компьютер является центром сетевой инфраструктуры.

srv-002 – виртуальный сервер для развертывания (Debian 11.11).

Виртуальная машина на базе srv-001 (QEMU, 1.5 GB RAM).

Служит целевым сервером для тестового развертывания приложений.

Установлены: Docker, Docker Compose, Nginx.

Особенность: процессор хоста не поддерживает аппаратную виртуализацию (VT-x), поэтому VM работает в режиме TCG — медленнее, но для тестового стенда достаточно.

vps-001 – арендованный на fvds.ru (Ubuntu 24.04.4 LTS).

Выполняет роль домашнего проху и обеспечивает доступ к внешним ресурсам.

Из-за ограничений провайдера (DPI) прямой WireGuard-туннель не работает, поэтому туннель поднят через SSH SOCKS-прокси.

vps-002 – арендованный на fvds.ru (Ubuntu 24.04.4 LTS).

Выполняет роль production: на данном VPS развёрнут сайт-визитка, доступный по домену ts-dev27.ru.

CI/CD

Для проверки работоспособности проектов и выявления недочетов при разработке был развернут тестовый стенд полного цикла с собственным Git-сервером Gitea (в Docker на srv-001):

push → *Gitea Actions* → *ci-templates/test.yml* → *ci-templates/deploy.yml* → *srv-002*

Чтобы избежать дублирования кода и ошибок при развертывании, был создан отдельный репозиторий – *ci-templates*, в котором описана общая логика (*test.yml* и *deploy.yml*) для разрабатываемых проектов.

Для упрощения разработки также реализована двухуровневая система секретов:

- первый уровень – владелец (*SSH_KEY*, *SSH_USER*, *SSH_HOST*, *DB_USER*, *DB_PASSWORD*, *EMAIL_HOST_USER*, *EMAIL_HOST_PASSWORD*);
- второй уровень – репозиторий.

Чтобы снизить нагрузку на канал провайдера и на *srv-002*, был создан отдельный образ для job – *gitea-runner-image* (Python, git, openssh-client, PostgreSQL), в котором стартуют задачи. Сам контейнер *gitea-runner* использует образ *gitea/act_runner:latest* и содержит только git.

Сети и адресация

Поскольку в инфраструктуре используются две сети – локальная и VPN (WireGuard), то для разделения по ролям и масштабируемости диапазон IP-адресов был разбит следующим образом:

1. роутер и серверы – *XX.XX.XX.1* – *XX.XX.XX.10* и *YY.YY.YY.1* – *YY.YY.YY.10*;
2. рабочие станции – *XX.XX.XX.11* – *XX.XX.XX.20* и *YY.YY.YY.11* – *YY.YY.YY.20*;
3. ноутбуки – *XX.XX.XX.21* – *XX.XX.XX.30* и *YY.YY.YY.21* – *YY.YY.YY.30*.

Питание

Для бесперебойной работы и защиты от потери данных по USB подключен ИБП Ippon Back Basic 650. Это позволило настроить автоматическое выключение сервера при пропадании питания от сети через *upsmon*. Виртуальный сервер *srv-002* при этом выключается вместе с хостом (*on_poweroff=destroy*) и поднимается снова при его старте (*autostart*).

Текущее состояние

Инфраструктура активно развивается и расширяется. При этом основной функционал – тестовый стенд – уже полностью настроен и работает.