СВОЙ СЕРВЕР · ДЛЯ ИТ И БЕЗОПАСНОСТИ

Памятка по безопасности

Каждое утверждение ниже проверено 15 ноября 2026 на чистой машине Ubuntu 22.04 (2 ядра, 4 ГБ) с Layers 3.0.1, поставленным командой из письма с ключом. У каждого — команда, чтобы ваша служба проверила это на своём сервере. Чего мы не проверяли — так и написано.

Пароль панели администратора

С 15.11.2026 установщик сам ставит случайный пароль панели администратора (God Mode, /godmode) в /opt/layers/.env (ADMIN_PASSWORD, права 600). Серверы, поставленные раньше, пускают по admin / admin: ещё раз запустите команду установки из письма или команду ниже. Проверено на чистой машине: после этого admin / admin получает 401, и после перезапуска тоже; повторный запуск ничего не меняет.

```
cd /opt/layers && P=$(openssl rand -hex 16) && T=$(sudo docker compose exec -T layers wget -qO- --post-data "username=admin&password=admin&bearer=true" http://localhost:8080/data/auth/admin/login | sed -n 's/.*"token":"\([^"]*\)".*/\1/p') && [ -n "$T" ] && sudo docker compose exec -T layers wget -qO- --header "Authorization: Bearer $T" --header "Content-Type: application/json" --post-data "{\"current\":\"admin\",\"next\":\"$P\"}" http://localhost:8080/data/auth/admin/password >/dev/null && echo "ADMIN_PASSWORD=$P" | sudo tee -a .env >/dev/null && echo "Done: God Mode password is in /opt/layers/.env (ADMIN_PASSWORD)"
```

Проверка

```
cd /opt/layers && sudo docker compose exec -T layers wget -qO- --post-data "username=admin&password=admin" http://localhost:8080/data/auth/admin/login >/dev/null 2>&1 && echo "NOT CLOSED: admin/admin still works" || echo "Closed: admin/admin is refused"
```

Где лежат данные

- Всё — в /opt/layers на вашей машине: база PostgreSQL в data/db, файлы людей в data/layers/files, сертификаты HTTPS в data/caddy, ключ лицензии в license/, секреты сервера в .env.
- .env создаётся с правами 600 (только root); каталог базы — 700.
- Файлы людей пишутся с правами 644. С 15.11.2026 установщик закрывает их каталог (750) от других учёток сервера; на серверах, поставленных раньше, выполните команду ниже — сервер работает (проверено, здоров после перезапуска).

Проверка

```
sudo ls -la /opt/layers /opt/layers/data
sudo docker inspect layers-layers-1 -f '{{range .Mounts}}{{.Source}} → {{.Destination}}{{println}}{{end}}'
```

Закрыть файлы людей от других учёток сервера

```
sudo chmod 750 /opt/layers/data/layers && sudo docker compose -f /opt/layers/docker-compose.yml restart layer
s
```

Порты и куда ходит сервер

- Слушает: TCP 80 (переадресация на HTTPS и выпуск сертификата), TCP 443 (сайт и TURN поверх TLS для звонков), UDP 443, TCP 7881, UDP 7882, UDP 3478 (звонки). Больше ничего.
- Редакторы документов — a4 (.docx) и f1 (.xlsx) — отдельные контейнеры без своих портов: с ними список выше тот же (проверено 15.11.2026), браузер ходит к ним только через Caddy по /a4/ и /f1/. По настройкам в docker-compose.yml они спрашивают у Layers, кто человек, и берут и сохраняют файл через Layers — кто откроет файл, решает Диск; рабочие копии открытых документов — в data/editors.
- С 15.11.2026 панель администратора (/godmode и её API) отвечает только локальным сетям и самой машине; остальным — 403 (проверено на чистой установке). Другие адреса добавляются строкой GODMODE_ALLOW в .env или параметром установщика --godmode-allow.
- При установке машина скачивает с layers.md (install.sh, docker-compose.yml), с Docker Hub (образы), из архива Ubuntu (Docker, если его нет) и спрашивает api.ipify.org свой внешний адрес, чтобы предупредить о DNS.
- В работе (10 минут наблюдения с отрезанным выходом в интернет, после полного перезапуска): служба звонков спрашивала публичные STUN-серверы (stun.l.google.com, global.stun.twilio.com) о своём внешнем адресе, а Layers пытался достучаться до api.layers.md — ежечасная отметка ключа (следующий раздел). Больше ничего; браузер к внешним адресам не обращается.
- Закрытый контур: задайте NODE_IP (с 15.11.2026 установщик пишет его сам) и выключите отметку ключа. Проверено: после полного перезапуска без интернета, 6 минут — ни одной попытки выхода наружу, служба звонков работает, лицензия активна.
- С редакторами документов (проверено 15.11.2026: выход в интернет отрезан, полный перезапуск, 7 минут наблюдения): сервер поднялся за 104 с, лицензия активна, служба звонков работает с NODE_IP, оба редактора healthy. Редактор документов (a4) 7 раз пытался достучаться до support.sumoffice.com — службы производителя редактора — и без ответа продолжил работать; редактор таблиц (f1) наружу не ходил.

- Без NODE_IP и без интернета служба звонков не стартует (перезапускается с «could not resolve external IP»); остальной Layers работает.
- С настоящим доменом Caddy получает сертификат Let’s Encrypt через интернет; на стенде это не проверяли (там имя layers.localhost и свой сертификат).

Открытые порты

```
sudo ss -ltnup | grep docker-proxy
```

Как мы считали попытки выхода

```
SUB=$(sudo docker network inspect layers_default -f '{{(index .IPAM.Config 0).Subnet}}')
sudo iptables -I DOCKER-USER 1 -s $SUB -j DROP
sudo iptables -I DOCKER-USER 1 -s $SUB -j LOG --log-prefix "LAYERS-OUT "
for n in 10.0.0.0/8 172.16.0.0/12 192.168.0.0/16; do sudo iptables -I DOCKER-USER 1 -s $SUB -d $n -j RETURN;
done
sudo dmesg | grep LAYERS-OUT          # what the server tried to send out
```

Закрытый контур

```
cd /opt/layers
echo "NODE_IP=10.0.0.10" | sudo tee -a .env >/dev/null       # the server's address in your network
sudo tee docker-compose.override.yml >/dev/null <<'EOF'
services:
  layers:
    environment:
      LAYERS_LICENSE_SEEN_ENABLED: "false"
EOF
sudo docker compose up -d
```

Что уходит наружу; проверка ключа — офлайн?

- Ключ проверяется на самом сервере. Проверено: выход в интернет отрезан, полный перезапуск всех служб — лицензия LICENSED, режим full, никого не заблокировало.
- Если у сервера есть интернет, раз в час он отправляет на api.layers.md id ключа, id установки, id узла и версию — без людей, файлов и содержимого. В ответе может прийти продлённый ключ для этой установки; сервер проверяет его подпись и установку и ставит сам. Нет ответа — ничего не меняется.
- Отметка выключается строкой LAYERS_LICENSE_SEEN_ENABLED=false (см. «Закрытый контур» выше); продлённый ключ тогда кладётся файлом в /opt/layers/license.

Проверка

```
curl -s https://layers.example.com/layers/license/status | grep -o '"mode":"[A-Z]*"\|"regime":"[a-z-]*"'
# "mode":"LICENSED"
# "regime":"full"
```

Резервные копии и восстановление

- Всё состояние — база, файлы людей, .env и каталог ключа. Копия — две команды, сервер не останавливается.
- Восстановление проверено: база удалена, файлы стёрты, затем восстановлены из копии — те же 5 человек, 8 страниц, тот же файл с той же контрольной суммой, лицензия активна, владелец видит свои файлы. Сервер вернулся примерно за 1 мин 40 с.

Копия

```
cd /opt/layers
sudo docker compose exec -T db pg_dump -U layers -d layersdb -Fc -Z6 > ~/layersdb.dump
sudo tar czf ~/layers-files.tar.gz -C /opt/layers data/layers .env license
```

Восстановление

```
cd /opt/layers
sudo docker compose stop layers
sudo docker compose exec -T db psql -U layers -d postgres -c "drop database layersdb with (force)"
sudo docker compose exec -T db psql -U layers -d postgres -c "create database layersdb owner layers"
sudo docker compose exec -T db pg_restore -U layers -d layersdb --no-owner < ~/layersdb.dump
sudo tar xzf ~/layers-files.tar.gz -C /opt/layers data/layers
sudo docker compose up -d
```

Обновление и откат

- Версия — одна строка в .env (LAYERS_IMAGE). Обновление: копия базы, новая строка, docker compose up -d; изменения базы применяются сами. Перерыв — около 80 секунд.
- Откат проверен: 3.0.1 → 3.0.0 на той же базе поднялся и сохранил лицензию (инструмент базы предупреждает, что схема новее), затем обратно на 3.0.1. Надёжный откат — прежняя версия плюс копия, снятая перед обновлением (восстановление выше).

Обновление

```
cd /opt/layers
sudo docker compose exec -T db pg_dump -U layers -d layersdb -Fc -Z6 > ~/before-update.dump
sudo sed -i '/^LAYERS_IMAGE=/d' .env && echo "LAYERS_IMAGE=hissih/layers:3.0.1" | sudo tee -a .env >/dev/null
sudo docker compose up -d
sudo docker inspect layers-layers-1 -f '{{.Config.Image}} {{.State.Health.Status}}'
```

Откат

```
cd /opt/layers
sudo sed -i '/^LAYERS_IMAGE=/d' .env && echo "LAYERS_IMAGE=hissih/layers:3.0.0" | sudo tee -a .env >/dev/null
sudo docker compose up -d
```

Журналы и аудит

- Каждый запрос — строка в журнале сервера: метод, путь, ответ, время, id пользователя и браузер.
- Панель администратора ведёт журнал попыток входа (время, адрес, успех); в нём последние попытки с последнего перезапуска.
- Входы людей записываются в базу (auth_journal: время, способ, результат, адрес, браузер).
- Журналы контейнеров ротируются: docker-compose.yml держит не больше 5 файлов по 20 МБ на контейнер (драйвер json-file, max-size и max-file); пределы меняются там же, либо отправляйте журналы в свою систему. Узел, поставленный раньше, получает ротацию после нового docker-compose.yml и docker compose up -d.

Журнал сервера и драйвер журналов

```
cd /opt/layers
sudo docker compose logs --since 1h layers | grep "\[access\]"
sudo docker inspect layers-layers-1 -f '{{.HostConfig.LogConfig.Type}} {{json .HostConfig.LogConfig.Config}}'
# json-file {"max-file":"5","max-size":"20m"}
```

Входы в панель администратора

```
J=$(mktemp)
curl -s -c $J -o /dev/null https://layers.example.com/data/auth/admin/login --data "username=admin&password=
$(sudo sed -n 's/^ADMIN_PASSWORD=//p' /opt/layers/.env)"
curl -s -b $J https://layers.example.com/data/auth/admin/logins; rm -f $J
# [{"at":"2026-11-15T00:31:13Z","ip":"…","ok":true}, …]
```

Кто имеет доступ

- root на сервере имеет всё: .env, базу, файлы. Держите машину только за своими администраторами.
- Панель администратора (God Mode): одна служебная учётка admin со случайным паролем в /opt/layers/.env (ADMIN_PASSWORD); открыта только из локальных сетей, если в GODMODE_ALLOW не перечислены другие адреса. На серверах, поставленных до 15.11.2026, пароль admin и панель открыта отовсюду — смените пароль (команда вверху) и возьмите файл установки от 15.11.
- Внутри Layers людьми и доступом управляют владелец пространства и назначенные им администраторы.
- У команды Layers доступа к вашему серверу нет: средств удалённого доступа не ставится, а единственное обращение Layers наружу — отметка ключа, которую можно выключить.

Смена пароля God Mode

```
cd /opt/layers && P=$(openssl rand -hex 16) && T=$(sudo docker compose exec -T layers wget -qO- --post-data
"username=admin&password=admin&bearer=true" http://localhost:8080/data/auth/admin/login | sed -n 's/.*"toke
n":"\([^"]*\)".*/\1/p') && [ -n "$T" ] && sudo docker compose exec -T layers wget -qO- --header "Authorizatio
n: Bearer $T" --header "Content-Type: application/json" --post-data "{\"current\":\"admin\",\"next\":\"$P\"}"
http://localhost:8080/data/auth/admin/password >/dev/null && echo "ADMIN_PASSWORD=$P" | sudo tee -a .env >/de
v/null && echo "Done: God Mode password is in /opt/layers/.env (ADMIN_PASSWORD)"
```

Проверка

```
cd /opt/layers && sudo docker compose exec -T layers wget -qO- --post-data "username=admin&password=admin" ht
tp://localhost:8080/data/auth/admin/login >/dev/null 2>&1 && echo "NOT CLOSED: admin/admin still works" || ec
ho "Closed: admin/admin is refused"
```

Откуда открыта God Mode

```
curl -s -o /dev/null -w "%{http_code}\n" https://layers.example.com/godmode      # from a machine not in GODM
ODE_ALLOW
# 403
cd /opt/layers && sudo sed -i '/^GODMODE_ALLOW=/d' .env && echo "GODMODE_ALLOW=private_ranges 203.0.113.7" |
sudo tee -a .env >/dev/null && sudo docker compose up -d
```

Шифрование

- В пути: HTTPS с TLS 1.3 (Caddy); порт 80 отвечает 308 на HTTPS. Звонки из сетей, где открыт только 443, идут через TURN поверх TLS на 443.
- На диске: база и файлы людей хранятся как есть — .docx на диске тот же файл. Если политика требует шифрования на диске, разместите /opt/layers на зашифрованном томе (например, LUKS). Такую схему мы не проверяли.
- Пароли людей хранятся хэшами bcrypt; пароль God Mode после смены — тоже bcrypt; пароли базы и Redis и секреты сервера — случайные, в .env (600).

В пути

```
echo | openssl s_client -connect layers.example.com:443 -servername layers.example.com 2>/dev/null | grep -E
"Protocol|Cipher"
curl -s -o /dev/null -w "%{http_code} %{redirect_url}\n" http://layers.example.com/
# 308 https://layers.example.com/
```

На диске

```
sudo file /opt/layers/data/layers/files/* | head
sudo docker compose -f /opt/layers/docker-compose.yml exec -T db psql -U layers -d layersdb -Atc "select left
(password,4) from users limit 3"
# $2a$  (bcrypt)
```

Проверено 15.11.2026, Layers 3.0.1

Страница подготовлена с помощью ИИ по журналам наших проверок.