

2. Установка сервера

Эталонная установка — Linux с systemd, ядро и консоль в одном процессе (`gits-server run`). Раздельная установка ядра и консоли описана в §9. Все команды выполняются от `root`, если не указано иное.

Раскладка файлов:

Путь	Владелец, права	Содержимое
<code>/usr/local/bin/gits-server</code> , <code>gits-console</code> , <code>gitsctl</code>	<code>root</code> , 0755	исполняемые файлы
<code>/etc/gits/server.yaml</code>	<code>root:gits</code> , 0640	конфигурация
<code>/etc/gits/gits.env</code>	<code>root</code> , 0600	необязательно: <code>GITS_DB_DSN</code> с паролем БД
<code>/etc/gits/console.crt</code> , <code>console.key</code>	<code>root:gits</code> , 0644 / 0640	сертификат HTTPS консоли
<code>/var/lib/gits/ca/</code>	<code>gits</code> , 0700	УЦ СКР: <code>ca.crt</code> , <code>ca.key</code> (0600)
<code>/var/lib/gits/keys/</code>	<code>gits</code> , 0700	ключи подписи: <code>commands.key</code> , <code>config.key</code> , <code>update.key</code> , <code>deploy.key</code> (0600)
<code>/var/lib/gits/server/</code>	<code>gits</code> , 0700	сертификат и ключ шлюза агентов

Ядро отказывается запускаться, если файлы ключей доступны группе или остальным пользователям.

1. Подготовка узла

```
# Системная учётная запись и каталоги
useradd --system --home-dir /var/lib/gits --shell /usr/sbin/nologin gits
install -d -m 0750 -o root -g gits /etc/gits
install -d -m 0700 -o gits -g gits /var/lib/gits

# Исполняемые файлы из релиза
install -m 0755 gits-server_<ver>_linux_amd64 /usr/local/bin/gits-server
install -m 0755 gitsctl_<ver>_linux_amd64 /usr/local/bin/gitsctl
gits-server version
```

Синхронизация времени обязательна (01 §1.4). Проверьте её командой `timedatectl` (`System clock synchronized: yes`).

2. PostgreSQL

Нужны база данных и роль-владелец. Расширения не требуются, схему создают миграции СКР.

```
# Debian/Astra: apt install postgresql; RHEL: dnf install postgresql-server && postgresql-setup --initdb
sudo -u postgres psql -v ON_ERROR_STOP=1 -c "CREATE ROLE gits LOGIN PASSWORD 'длинный-случайный-пароль'"
sudo -u postgres createdb -O gits gits
```

Astra Linux SE с мандатным контролем доступа. PostgreSQL из состава Astra отклоняет подключение пользователя без мандатной метки (`error obtaining MAC configuration for user "gits"`). Роли БД `gits`

должна соответствовать учётная запись ОС с меткой:

```
pdpl-user -l 0:0 gits
```

PostgreSQL на отдельном узле. Разрешите подключение в `pg_hba.conf` только с адресов узлов СКР (`hostssl gits gits 10.0.0.10/32 scram-sha-256`) и укажите в DSN `sslmode=require` или `verify-full`.

3. Конфигурация

```
install -m 0640 -o root -g gits server.yaml.example /etc/gits/server.yaml
```

Пример с комментариями ко всем параметрам — `deploy/server.yaml.example` в `gits-deploy_<ver>.zip`. Обязательно задайте:

Параметр	Значение
<code>db_dsn</code>	<code>postgres://gits:<пароль>@127.0.0.1:5432/gits?sslmode=disable</code> (или <code>GITS_DB_DSN</code> в <code>/etc/gits/gits.env</code>)
<code>data_dir</code>	<code>/var/lib/gits</code>
<code>gateway.listen</code>	<code>0.0.0.0:8443</code> . По умолчанию <code>127.0.0.1:8443</code> , и тогда удалённые агенты не подключатся
<code>gateway.agent_endpoints</code>	адреса, по которым агенты подключаются к ядру или ретрансляторам, например <code>gits.bank.local:8443</code>
<code>api.listen</code> , <code>api.tls_cert</code> , <code>api.tls_key</code>	<code>0.0.0.0:443</code> и сертификат консоли (§6)

Пароль БД можно не хранить в `server.yaml`:

```
install -m 0600 /dev/null /etc/gits/gits.env
echo 'GITS_DB_DSN=postgres://gits:<пароль>@127.0.0.1:5432/gits?sslmode=disable' > /etc/gits/gits.env
```

Утилита `gitsctl` работает с теми же файлами и должна запускаться **от имени gits**, иначе ключи получат чужого владельца. Ниже используется сокращение:

```
gctl() { sudo -u gits env GITS_CONFIG=/etc/gits/server.yaml $(cat /etc/gits/gits.env 2>/dev/null) gitsctl "$@"; }
```

4. Схема БД

```
sudo -u gits env GITS_CONFIG=/etc/gits/server.yaml gits-server migrate
# migrations applied
```

Ядро (`gits-server run` / `gits-core`) также применяет миграции при каждом запуске. Консоль (`gits-console`) миграции не применяет и ждёт появления схемы до 2 минут.

5. УЦ, сертификат шлюза и ключи

```
gctl init-ca -name "GITS Root CA <организация>"
gctl issue-server-cert -dns gits.bank.local,gits -ip 10.0.0.10
gctl gen-signing-keys
```

- `init-ca` создаёт корневой сертификат СКР на 10 лет (`/var/lib/gits/ca/ca.crt`) и его ключ. Этим УЦ подписываются сертификаты шлюза (2 года), агентов и ретрансляторов (1 год). Системное хранилище доверенных сертификатов в этой цепочке не используется: агенты доверяют только корню СКР.
- `issue-server-cert` — сертификат шлюза агентов. В `-dns` и `-ip` перечислите **все** имена и адреса, по которым агенты обращаются к ядру, включая адрес из `gateway.agent_endpoints`. По умолчанию подставляются `localhost` и `127.0.0.1`, и удалённые агенты такой сертификат не примут. Если набор адресов изменился, выпустите сертификат заново и перезапустите сервер.
- `gen-signing-keys` создаёт ключи Ed25519 для подписи команд, конфигураций и пакетов обновлений. Ключ `deploy.key` для push-установки ядро создаст при первом запуске.

Проверка:

```
ls -l /var/lib/gits/ca /var/lib/gits/keys /var/lib/gits/server # ключи — 0600, владелец gits
openssl x509 -in /var/lib/gits/server/server.crt -noout -ext subjectAltName
```

6. Сертификат консоли

Консоль открывается в браузере, поэтому её сертификат выпускает **корпоративный УЦ**, которому доверяют APM администраторов, а не УЦ СКР. Имя в сертификате — адрес консоли (`gits.bank.local`).

```
install -m 0644 -o root -g gits gits-console.crt /etc/gits/console.crt # сертификат + промежуточные
install -m 0640 -o root -g gits gits-console.key /etc/gits/console.key
```

Когда задан `api.tls_cert`, cookie сессии получает флаг `Secure`. Обратный прокси перед консолью возможен (`api.listen: 127.0.0.1:8080`, `secure_cookie: true`), но тогда в журнале действий администраторов вместо адреса клиента будет записан адрес прокси: заголовок `X-Forwarded-For` не используется.

7. Первый администратор и перечень мер

```
GITS_PASSWORD='не короче 16 символов' gctl create-user -login admin -roles system_admin
# user created: admin (...) roles=[system_admin]

install -m 0644 tech_measures_level1.csv /tmp/measures.csv # файл должен быть доступен пользователю git
s
gctl load-measures -csv /tmp/measures.csv
# 276 measures loaded
```

Роли: `system_admin`, `checks_admin`, `analyst`, `grizi_lead`, `grizi_dispatcher`, `grizi_secretary`, `auditor`.
Администратор системы не видит событий и свидетельств (разделение полномочий), для этого нужны

отдельные пользователи с ролями `auditor` или `analyst`. Раскатку обновлений агентов утверждает второй пользователь с ролью `system_admin`.

8. Запуск службы

```
install -m 0644 deploy/systemd/gits-server.service /etc/systemd/system/  
systemctl daemon-reload  
systemctl enable --now gits-server  
journalctl -u gits-server -n 20 -o cat
```

В журнале должны появиться строки:

```
level=INFO msg="api listening" addr=0.0.0.0:443 tls=true  
level=INFO msg="gateway listening" component=gateway addr=[::]:8443
```

Юнит запускает сервер от `gits` с `ProtectSystem=strict`. Запись разрешена только в `/var/lib/gits`, а привязка к порту 443 без `root` выполняется через `CAP_NET_BIND_SERVICE`.

Проверка консоли: откройте `https://gits.bank.local/` и войдите как `admin`. Проверка из командной строки:

```
curl -s -o /dev/null -w '%{http_code}\n' https://gits.bank.local/ # 200
```

Межсетевой экран узла: откройте входящие TCP 443 (из сегмента администраторов) и TCP 8443 (от хостов и ретрансляторов); при `syslog.enabled` — UDP 5514.

9. Раздельная установка ядра и консоли

Используется, когда консоль вынесена на отдельный узел или ядер несколько.

Узел	Файлы	Юнит
ядро	<code>gits-core</code> , <code>gitsctl</code> , <code>/etc/gits/server.yaml</code> , весь <code>/var/lib/gits</code>	<code>deploy/systemd/gits-core.service</code>
консоль	<code>gits-console</code> , <code>/etc/gits/server.yaml</code> (секции <code>db_dsn</code> и <code>api</code>), сертификат консоли	<code>deploy/systemd/gits-console.service</code>

- Ядро и консоль связаны только через БД: задания передаются через таблицу `core_jobs`, сигналы — через `LISTEN/NOTIFY PostgreSQL`.
- Ключей и УЦ на узле консоли нет. Команды `gitsctl`, которым нужны ключи (`init-ca`, `issue-server-cert`, `gen-signing-keys`, `publish-agent`), выполняются на узле ядра.
- Несколько ядер: у всех одинаковые `/var/lib/gits` (УЦ и ключи) и одна БД. Фоновые задания выполняет ядро-лидер, выбранное через `advisory lock`. Агенты получают адреса всех ядер в `gateway.agent_endpoints`. Пороговые правила считаются отдельно в каждом ядре.
- Не включайте `gits-server.service` вместе с `gits-core` / `gits-console` на одном узле.

10. Первичная настройка после запуска

1. **Пакеты агента.** Установочные комплекты консоли и обновления собираются только из подписанных пакетов репозитория СКР. Опубликуйте нужные платформы на узле ядра:

```
bash gctl publish-agent -file gits-agent_<ver>_windows_amd64.exe -version <ver> -os windows -arch amd64 -format exe gctl publish-agent -file gits-agent_<ver>_linux_amd64 -version <ver> -os linux -arch amd64 -format bin gctl publish-agent -file gits-agent_<ver>-1_amd64.deb -version <ver> -os linux -arch amd64 -format deb gctl publish-agent -file gits-agent-<ver>-1.x86_64.rpm -version <ver> -os linux -arch amd64 -format rpm
```

Форматы `exe` и `bin` нужны для установочных комплектов и обновления агентов, установленных командой `gits-agent install`. Форматы `deb` и `rpm` нужны для обновления агентов, установленных из пакетов. Загрузить пакет можно и в консоли, раздел «Обновления агентов». 2. **Информационные системы.** В разделе «Информационные системы» создайте ИС и укажите для каждой «Адреса подключения агентов» (ядро или ретранслятор сегмента). Выдайте пользователям доступ к ИС. То же через CLI: `gctl create-is -name ...`, `gctl grant-is -login ... -is ...`. 3. **Уведомления, LDAP, syslog, SIEM** — секции `notify`, `acc`, `syslog`, `siem` в `server.yaml` и перезапуск службы.

11. Резервное копирование и восстановление

Что	Как	Частота
БД	<code>pg_dump -Fc gits > gits-\$(date +%F).dump</code> или непрерывная архивация WAL (RPO 15 мин, design/02 §7)	ежедневно / непрерывно
УЦ и ключи	<code>tar -C /var/lib/gits -czf gits-keys.tgz ca key s server</code>	после <code>init-ca</code> , <code>gen-signing-keys</code> , <code>issue -server-cert</code> и появления <code>deploy.key</code>
Конфигурация	<code>/etc/gits</code>	после изменений

Архив с УЦ и ключами храните отдельно от копий БД, зашифрованным и с ограниченным доступом. Без него восстановленная из копии система не сможет продлевать сертификаты агентов и подписывать им команды, конфигурацию и обновления. Агенты доверяют ключам, которые получили при регистрации.

Восстановление: развернуть узел по §1–3, восстановить `/etc/gits` и `/var/lib/gits` с сохранением владельца `gits` и прав 0600 на ключи, создать пустую БД (§2), затем `pg_restore -d gits gits-<дата>.dump` и запустить службу.

12. Обновление сервера

1. Прочитайте раздел версии в `CHANGELOG.md`: там перечислены миграции и особые шаги.
2. Сделайте резервную копию БД и `/var/lib/gits` (§11).
3. Замените исполняемые файлы и перезапустите службу: `systemctl restart gits-server`. Миграции применяются при старте. При раздельной установке сначала обновите и запустите ядро, затем консоль.
4. Опубликуйте пакеты агента новой версии (§10) и запустите раскатку в разделе «Обновления агентов».

13. Сервер на Windows

Сборки `gits-server_<ver>_windows_amd64.exe` и `gitsctl_<ver>_windows_amd64.exe` работают с теми же `server.yaml` и командами. Режим службы Windows у сервера нет: запуск при загрузке настраивается планировщиком заданий или внешней обёрткой службы, от имени выделенной учётной записи. Проверка прав на файлы ключей на Windows не выполняется, поэтому ограничьте доступ к `data_dir` списками доступа (SYSTEM, Администраторы, учётная запись службы). Эталонной и проверенной остаётся установка на Linux.