СКАЙФОЛЛ ЛАБС

SkyV Discovery

Система сбора информации посредством дискаверинга

ИНСТРУКЦИЯ ПО УСТАНОВКЕ

экземпляра программного обеспечения

Версия 0.01

Москва, 2026 г.

Содержание

1. Назначение документа

Настоящий документ содержит порядок установки и первоначальной настройки экземпляра программного обеспечения SkyV Discovery (далее - «Система»). **SkyV Discovery предназначена для сбора и актуализации технической информации об устройствах, сетевых сервисах и параметрах инфраструктуры с применением агентов дискаверинга.**

Инструкция описывает требования к серверу, подготовку программной среды, развертывание контейнеров, настройку параметров безопасности, проверку готовности и первичные действия администратора после запуска Системы.

Важно. Значения, отмеченные в угловых скобках, например <FQDN_СИСТЕМЫ>, являются примерами. Перед выполнением команд замените их на значения, согласованные для конкретного экземпляра.

2. Состав развертываемого экземпляра

Экземпляр SkyV Discovery включает центральный hub (agent-hub), специализированные агенты и клиентские приложения. Компоненты могут разворачиваться в контейнерах (Docker Compose, Kubernetes и др.), на виртуальных машинах или физических серверах. Состав конкретной поставки определяется сценарием заказчика. Типовой состав:

Компонент	Назначение	Особенности
hub-postgres	СУБД PostgreSQL для хранения учетных записей, настроек агентов и служебных данных.	Данные сохраняются в именованном томе Docker hub_postgres_data.
kafka	Брокер сообщений Apache Kafka для обмена служебными сообщениями.	Доступен только внутри сети Docker в типовой конфигурации.
hub-api	Серверная часть SkyV Discovery на Go: API, авторизация, управление реестром агентов, выполнение миграций БД.	HTTP: 8080; gRPC: 9090.
web	Веб-интерфейс администратора и оператора.	Статический веб-клиент публикуется через внутренний Nginx.
gateway	Nginx-шлюз, через который предоставляются веб-интерфейс, API и служебные маршруты.	В типовой сборке публикуется на TCP 8090.
inventory	Перенаправление информации, собранной с клиентских приложений в шину данных. Хранение и управление конфигурациями клиентских приложений.	HTTP: 8081; gRPC: 9091.
snmp	Сканирование сетевой инфраструктуры по протоколу SNMP.	Требует базы данных OID.
nmap	Сканирование сетевой инфраструктуры с помощью утилиты NMAP.	Требует предустановленную утилиту NMAP.

После установки администратор создает учетные записи и настраивает агентов. В зависимости от сценария работы могут использоваться агенты SNMP, Nmap, mon и программные агенты сбора данных.

3. Системные и сетевые требования

3.1. Требования к серверу

Для развертывания одного экземпляра Системы рекомендуется выделить отдельный сервер или виртуальную машину. Минимальные параметры для начального экземпляра приведены в таблице. Для большого числа агентов, широких диапазонов сканирования и длительного хранения данных параметры должны быть увеличены по результатам оценки нагрузки.

Параметр	Минимальное значение	Рекомендации
Операционная система	64-разрядный Linux (Ubuntu 22.04+, Debian, Astra Linux и др.)	64-разрядная ОС, регулярная установка обновлений безопасности.
Процессор	1–2 vCPU для hub; 2–4 vCPU и более для inventory и других агентов	Увеличивайте при одновременной работе большого числа агентов и интенсивном использовании интерфейса.
Оперативная память	512 МБ–1 ГБ	Рекомендуется резерв свободной памяти для Docker, PostgreSQL и Kafka.
Дисковое пространство	50 ГБ сверх ОС	Объем зависит от срока хранения служебных данных, журналов и резервных копий.

3.2. Требования к программному обеспечению

- Docker Engine с поддержкой Docker Compose (плагин docker compose).
- Средства доступа к серверу: SSH-клиент и учетная запись с правами sudo.
- Современный браузер с включенным JavaScript для работы с веб-интерфейсом.
- Утилиты curl и tar для проверки работы и распаковки дистрибутива.

3.3. Сетевые требования и информационная безопасность

Направление	Порт / протокол	Назначение
Рабочие места пользователей -> gateway	TCP 8090 / HTTP в типовой поставке	Доступ к веб-интерфейсу и API через Nginx-шлюз. В промышленной среде рекомендуется использовать HTTPS на внешнем обратном прокси.
Администратор -> hub-api	TCP 8080 / HTTP	Служебная диагностика API. Не публикуйте порт во внешние сети без необходимости.
hub-api -> агенты	TCP 8080 и/или TCP 9090	Подключение к HTTP и gRPC интерфейсам зарегистрированных агентов. Конкретные значения задаются в карточке агента.
hub-api -> PostgreSQL и Kafka	Внутренняя сеть Docker	Обмен между контейнерами Системы. Внешнее опубликование портов не требуется.

Требование безопасности. Ограничьте доступ к TCP 8080 и TCP 9090 правилами межсетевого экрана. Не используйте значения по умолчанию для паролей СУБД, JWT_SECRET, SECRETS_ENCRYPTION_KEY и начальных учетных записей.

4. Подготовка сервера

4.1. Подготовка операционной системы

Выполняйте действия от имени пользователя, которому предоставлены права sudo. Перед установкой обновите состав пакетов и операционную систему:

```
sudo apt update
sudo apt -y upgrade
sudo apt install -y ca-certificates curl tar
```

Убедитесь, что на сервере настроены корректные DNS-серверы, синхронизация времени и сетевой доступ к адресам агентов, которые планируется подключать к SkyV Discovery.

4.2. Установка Docker Engine и Docker Compose

Установите Docker Engine и плагин Docker Compose в соответствии с регламентом вашей организации или инструкциями поставщика Docker. После установки проверьте доступность команд:

```
docker --version
docker compose version
```

Чтобы выполнять команды Docker без sudo, добавьте пользователя, выполняющего установку, в группу docker и повторно откройте SSH-сеанс:

```
sudo usermod -aG docker $USER
```

4.3. Размещение дистрибутива

Получите дистрибутив SkyV Discovery от ООО «Скайфолл Лабс» и разместите его в защищенном каталоге, например /opt/skyv-discovery. Имя архива в примере является условным:

```
sudo mkdir -p /opt/skyv-discovery
sudo tar -xf <skyv-discovery-distribution>.tar.gz -C /opt/skyv-discovery
sudo chown -R $USER:$USER /opt/skyv-discovery
```

Перейдите в каталог серверной части, содержащий файлы docker-compose.yml и .env.example. При сборке контейнеров из исходного кода необходимо сохранить структуру репозитория, используемую Dockerfile: каталоги common/common-lib-go, common/protobuf и itam/agent-hub/back должны быть доступны из общего корня контекста сборки.

5. Настройка дистрибутива

5.1. Настройка файла окружения

В каталоге серверной части создайте рабочий файл окружения на основе шаблона и ограничьте права на его чтение:

```
cd <каталог_серверной_части>
cp .env.example .env
chmod 600 .env
```

Откройте файл .env в текстовом редакторе и замените все примерные секреты и учетные данные. Состав основных параметров приведен в Приложении А.

Группа параметров	Действия перед запуском
PostgreSQL	Укажите уникальные значения HUB_POSTGRES_USER, HUB_POSTGRES_PASSWORD и HUB_POSTGRES_DB.
Авторизация	Задайте уникальный JWT_SECRET и значение JWT_ISSUER, соответствующее экземпляру.

Группа параметров	Действия перед запуском
Шифрование секретов	Задайте SECRETS_ENCRYPTION_KEY в формате Base64. Храните ключ в защищенном хранилище и не изменяйте его без согласованной процедуры миграции секретов.
Начальные учетные записи	Измените логины и задайте надежные пароли SEED_ADMIN_PASSWORD и SEED_VIEWER_PASSWORD. После первого входа замените или отключите неиспользуемые учетные записи.
Доступ из браузера	Укажите CORS_ORIGIN как URL, с которого пользователи будут открывать экземпляр Системы.
Интеграция с агентами	При использовании TLS задайте пути к доверенным центрам сертификации для HTTP и gRPC. Убедитесь, что соответствующие файлы доступны внутри контейнера hub-api.

5.2. Настройка доступа и TLS

По умолчанию Nginx-шлюз типовой сборки принимает HTTP-соединения на TCP 8090. Для производственного доступа рекомендуется разместить внешний обратный прокси с HTTPS или настроить TLS в соответствии с инфраструктурным стандартом организации.

- Опубликуйте для пользователей только согласованный внешний URL Системы.
- Ограничьте сетевой доступ к PostgreSQL, Kafka, hub-api и служебным интерфейсам контейнеров.
- При работе с агентами по TLS разместите файлы доверенных сертификатов в защищенном каталоге, смонтируйте его в hub-api в режиме только для чтения и укажите путь в переменных окружения.
- Не храните пароли, ключи и токены в документации, истории команд, незашифрованных резервных копиях и общедоступных каталогах.

6. Развертывание экземпляра

6.1. Запуск контейнеров

После подготовки файла .env проверьте конфигурацию Docker Compose и запустите контейнеры:

```
cd <каталог_серверной_части>
docker compose config
docker compose up -d --build
```

Команда запускает PostgreSQL, Kafka, серверную часть hub-api, веб-интерфейс и Nginx-шлюз. При первом запуске hub-api автоматически выполняет миграции структуры базы данных и создает начальные учетные записи, для которых в .env заданы пароли.

Проверьте состояние контейнеров:

```
docker compose ps
docker compose logs --tail=200 hub-api
```

6.2. Проверка работоспособности

Убедитесь, что все контейнеры имеют состояние running, а hub-api проходит проверку состояния. Выполните на сервере:

```
curl -f http://127.0.0.1:8090/health
```

При успешном запуске откройте в браузере адрес:

```
http://<FQDN_СИСТЕМЫ>:8090/
```

При использовании внешнего обратного прокси используйте согласованный HTTPS-адрес. Дополнительно интерфейс Swagger/API в типовой сборке доступен по пути /docs/.

Проверка	Ожидаемый результат	Действия при ошибке
docker compose ps	Контейнеры hub-postgres, kafka, hub-api, web и gateway запущены.	Просмотрите логи проблемного контейнера: docker compose logs <имя_сервиса>.
GET /health	HTTP-ответ без ошибки.	Проверьте логи hub-api, параметры DATABASE_URL, состояние hub-postgres и миграции.
Открытие веб-интерфейса	Появляется окно входа в SkyV Discovery.	Проверьте доступность TCP 8090, правила межсетевого экрана и настройки внешнего прокси.
Вход администратора	Успешная авторизация с учетной записью, заданной в SEED_ADMIN_LOGIN.	Проверьте, что пароль создан до первого запуска, затем убедитесь в отсутствии пробелов и неактуальных значений в .env.

7. Первичная настройка SkyV Discovery

После успешного запуска выполните первоначальную настройку в веб-интерфейсе под учетной записью администратора:

1. Измените пароль начальной учетной записи администратора и проверьте наличие только необходимых учетных записей.
2. Создайте пользователей и назначьте им роли в соответствии с задачами: администратор, оператор дискаверинга или наблюдатель.
3. Настройте сетевые диапазоны, исключения и профили учетных данных для разрешенных областей сбора.
4. Создайте и настройте агенты SNMP, Nmap и агентского сбора данных. Для каждого агента укажите адрес, транспорт, порты, параметры TLS и учетные данные, требуемые конкретным методом сбора.
5. Выполните проверку подключения. До подтверждения доступности не назначайте регулярное расписание запуска.
6. Запустите тестовый сбор вручную, проверьте последнюю активность и полученные результаты.
7. После успешной проверки настройте расписание запусков и правила уведомлений.

Важно. Перед запуском Nmap-сканирования убедитесь, что диапазон разрешен для сканирования правилами информационной безопасности и согласован с владельцами инфраструктуры.

8. Эксплуатационные операции

Операция	Команда	Комментарий
Просмотр статуса	docker compose ps	Показывает состояние контейнеров текущего экземпляра.
Просмотр журналов	docker compose logs --tail=200 hub-api	Для непрерывного вывода добавьте ключ -f.
Перезапуск	docker compose restart	Используйте после согласованного изменения параметров.
Остановка без	docker compose down	Именованные тома Docker сохраняются.

Операция	Команда	Комментарий
удаления данных		
Создание резервной копии БД	См. пример ниже.	Выполняйте перед обновлением или удалением данных.

Пример создания резервной копии PostgreSQL. Перед выполнением экспортируйте значения из .env в текущую сессию:

```
set -a
. ./.env
set +a
mkdir -p ./backups
docker compose exec -T hub-postgres pg_dump -U "$HUB_POSTGRES_USER" "$HUB_POSTGRES_DB" >
./backups/skyv-discovery-$(date +%F).sql
```

При обновлении экземпляра предварительно создайте резервную копию, остановите контейнеры, замените дистрибутив в соответствии с инструкцией к версии и запустите docker compose up -d --build. После обновления проверьте состояние контейнеров, вход в веб-интерфейс и доступность агентов.

9. Удаление экземпляра

Перед удалением обязательно создайте и проверьте резервную копию базы данных. Для остановки и удаления контейнеров без удаления данных используйте:

```
cd <каталог_серверной_части>
docker compose down
```

Полное удаление контейнеров и именованных томов Docker допустимо только после подтверждения, что данные и резервные копии больше не требуются:

```
docker compose down -v
```

После этого при необходимости удалите каталог распакованного дистрибутива:

```
sudo rm -rf /opt/skyv-discovery
```

Внимание. Команда docker compose down -v удаляет именованные тома Docker, включая данные PostgreSQL экземпляра SkyV Discovery. Операция необратима без заранее созданной резервной копии.

10. Контакты разработчика

Разработчик программного обеспечения: ООО «Скайфолл Лабс».

Канал связи	Контактные данные
Официальный сайт	skyflabs.ru
Техническая поддержка	support@skyflabs.ru
Телефон	+7 (495) 109-47-17

Приложение А. Параметры файла .env

Значения параметров должны быть определены до первого запуска экземпляра. Не переносите в рабочую среду значения, используемые в примерах разработки.

Переменная	Назначение	Требование
HUB_POSTGRES_USER	Имя пользователя PostgreSQL.	Используйте отдельную учетную запись экземпляра.
HUB_POSTGRES_PASSWORD	Пароль пользователя PostgreSQL.	Уникальный пароль, не хранить в открытом виде.
HUB_POSTGRES_DB	Имя базы данных SkyV Discovery.	Укажите согласованное имя БД.
DATABASE_URL / POSTGRES_DSN	Строка подключения hub-api к PostgreSQL.	В типовой Compose-конфигурации формируется с использованием HUB_POSTGRES_*; убедитесь в согласованности значений.
HTTP_ADDR	Адрес и порт HTTP-сервера hub-api.	По умолчанию используется порт 8080 внутри/в составе сборки.
CORS_ORIGIN	Разрешенный адрес веб-интерфейса для CORS.	Укажите фактический URL экземпляра.
JWT_SECRET	Секрет подписи токенов авторизации.	Используйте криптографически стойкое уникальное значение.
JWT_ISSUER	Идентификатор издателя токенов.	Укажите значение, уникальное для экземпляра.
SECRETS_ENCRYPTION_KEY	Ключ шифрования секретов агентов, закодированный Base64.	Храните в защищенном хранилище; не заменяйте без процедуры миграции.
SEED_ADMIN_LOGIN / SEED_ADMIN_PASSWORD	Начальная учетная запись администратора.	Задайте до первого запуска; после входа измените пароль.
SEED_VIEWER_LOGIN / SEED_VIEWER_PASSWORD	Начальная учетная запись наблюдателя.	Создавайте только при необходимости.
KAFKA_BROKERS	Адрес Kafka для hub-api.	В Compose-сборке: kafka:9092.
KAFKA_TOPIC_PREFIX	Префикс рабочих топиков Kafka.	Согласуйте при наличии общих брокеров.
GRPC_CLIENT_TLS_CA / HTTP_CLIENT_TLS_CA	Путь к корневому сертификату для TLS-подключений к агентам.	Файл должен быть доступен контейнеру hub-api в режиме только для чтения.
LOG_LEVEL / LOGGER_LEVEL	Уровень журналирования.	Для штатной эксплуатации используйте info или согласованное значение.

Приложение Б. Краткая проверка после установки

- ☐ Docker Engine и Docker Compose установлены; пользователь установки имеет доступ к Docker.
- ☐ Файл .env содержит уникальные значения всех паролей, ключей и начальных учетных записей.
- ☐ Контейнеры hub-postgres, kafka, hub-api, web и gateway запущены.
- ☐ Проверка GET /health выполняется без ошибки.
- ☐ Веб-интерфейс доступен по согласованному URL.
- ☐ Начальный администратор может войти в Систему; пароль начальной учетной записи изменен.
- ☐ Внешние правила межсетевого экрана ограничивают доступ к служебным портам.
- ☐ Создан тестовый агент; проверка подключения выполняется успешно.
- ☐ Резервное копирование PostgreSQL проверено до ввода экземпляра в эксплуатацию.