

Регламент резервного копирования и восстановления Логикор

В этой статье рассматривается процесс резервного копирования и восстановления компонентов системы Логикор.

Резервируемые компоненты

Резервному копированию подлежат только те компоненты, которые содержат состояние или критически важную конфигурацию.

Компоненты, которые подлежат резервному копированию (stateful-компоненты):

- **ClickHouse** — конфигурация (данные хранятся во внешнем слое);
- **ClickHouse Keeper** — координация кластера;
- **ETCD** — данные Patroni (Docker volume);
- **PostgreSQL** — данные Earth и Grafana (Docker volume);
- **S3 MinIO** — объекты (Docker volume);
- **Vault** — секреты и конфигурация (Docker volume);
- **Ansible** — конфигурация развёртывания (Git-репозиторий).

Компоненты, которые НЕ подлежат резервному копированию (stateless-службы):

- **Saturn (Collector)** — сбор логов;
- **Mercury (Frontend) + Earth (Backend)** — пользовательский интерфейс;
- **Grafana** — визуализация метрик;
- **Venus** — веб-интерфейс.

Данные службы разворачиваются заново с помощью Ansible.

Размещение компонентов

Компонент	Конфигурация	Данные	Где развернут
ClickHouse	/etc/clickhouse-server/	—	Host (отдельные VM)
ClickHouse Keeper	/etc/clickhouse-keeper/	/var/lib/clickhouse-keeper/ или /var/lib/clickhouse/coordination/	Host (отдельные VM)
ETCD	Vault, Ansible	Docker Volume (logiq_etcd-data)	Docker
PostgreSQL	Vault, Ansible	Docker Volume (logiq_patroni-data)	Docker
S3 MinIO	Vault, Ansible	Docker Volume (logiq_minio-data)	Docker
Vault	Vault, Ansible	Docker Volume (logiq_vault-data)	Docker

ClickHouse и ClickHouse Keeper развернуты на отдельных виртуальных машинах (внешний слой хранения), не через Docker. Остальные компоненты — в Docker.

Резервное копирование

ClickHouse

ClickHouse развернут на отдельных виртуальных машинах. Резервному копированию подлежит только конфигурация. Резервное копирование данных при необходимости выполняется отдельно через **clickhouse-backup**.

Резервное копирование **конфигурации**:

```
#!/bin/bash
BACKUP_DIR="/backup/clickhouse/$(date +%Y%m%d)"
mkdir -p "$BACKUP_DIR"
```

```
cp -r /etc/clickhouse-server/ "$BACKUP_DIR/config/"
```

Для резервного копирования **данных** используется инструмент **clickhouse-backup** (Altinity). Он предназначен для резервирования данных и размещения их во внешнем хранилище, например в S3. Конфигурация **/etc/clickhouse-backup/config.yml**:

```
clickhouse:
  username: "backup_user"
  password: "secure_password"
```

```
s3:
  access_key: "YOUR_KEY"
  secret_key: "YOUR_SECRET"
  bucket: "logiq-backups"
  endpoint: "https://s3.example.com"
  path: "clickhouse/"
```

```
general:
  remote_storage: "s3"
  backups_to_keep_local: 3
  backups_to_keep_remote: 14
```

Основные команды:

```
# Создать полную копию
clickhouse-backup create full_backup_$(date +%Y%m%d_%H%M%S)

# Отправить в S3
clickhouse-backup upload backup_name

# Или сразу создать и отправить
clickhouse-backup create_remote full_backup_$(date +%Y%m%d_%H%M%S)
```

Периодичность:

- **Конфигурация:** ежедневно;
- **Данные:** еженедельно (инкрементально).

ClickHouse Keeper

ClickHouse Keeper развернут на отдельных виртуальных машинах.

Резервное копирование:

```
#!/bin/bash
BACKUP_DIR="/backup/keeper/$(date +%Y%m%d)"
mkdir -p "$BACKUP_DIR"

# Конфигурация
cp -r /etc/clickhouse-keeper/ "$BACKUP_DIR/config/"

# Данные (встроенный Keeper)
cp -r /var/lib/clickhouse/coordination/ "$BACKUP_DIR/coordination/"
```

Для обеспечения консистентности данных рекомендуется выполнять резервное копирование одновременно на всех узлах кластера.

Периодичность: ежедневно.

ETCD (Patroni)

Компонент ETCD развернут в Docker, а его данные хранятся в Docker volume.

Резервное копирование:

```
#!/bin/bash
BACKUP_DIR="/backup/etcd/$(date +%Y%m%d)"
mkdir -p "$BACKUP_DIR"

# Остановить контейнер для консистентности
docker stop etcd

# Снапшот
ETCDCTL_API=3 docker run --rm \
  -v logiq_etcd-data:/etcd-data \
  -v "$BACKUP_DIR":/backup \
  quay.io/coreos/etcd:v3.5.0 \
  etcdctl snapshot save /backup/snapshot.db \
  --data-dir=/etcd-data

# Запустить обратно
docker start etcd
```

Периодичность: ежедневно.

PostgreSQL (Patroni)

PostgreSQL развернут в Docker вместе с Patroni (один compose\контейнер).

Резервное копирование:

```
#!/bin/bash
BACKUP_DIR="/backup/postgresql/$(date +%Y%m%d)"
mkdir -p "$BACKUP_DIR"

# Логический бэкап (без остановки)
docker exec patroni pg_dumpall -U postgres | gzip > "$BACKUP_DIR/full_backup.sql.gz"
docker exec patroni pg_dumpall --globals-only | gzip > "$BACKUP_DIR/globals.sql.gz"
```

S3 MinIO

MinIO развернут в Docker. Конфигурация компонента находится в Vault/Ansible, данные — в Docker volume.

Резервное копирование (ручное копирование):

```
#!/bin/bash
BACKUP_DIR="/backup/minio/$(date +%Y%m%d)"
mkdir -p "$BACKUP_DIR"

# Остановить контейнер
docker stop minio

# Скопировать volume
tar -czf "$BACKUP_DIR/minio-volume.tar.gz" \
  -C /var/lib/docker/volumes/logiq_minio-data/_data .

# Запустить обратно
docker start minio
```

Периодичность: при изменениях.

Vault

Vault развернут в Docker.

Резервное копирование:

```
#!/bin/bash
BACKUP_DIR="/backup/vault/$(date +%Y%m%d)"
mkdir -p "$BACKUP_DIR"

# Снапшот Raft
docker exec vault vault operator raft snapshot save "$BACKUP_DIR/vault_snapshot.db"
```

Периодичность: по регламенту Vault.

Ansible

Конфигурация Ansible (playbook, inventory, переменные, шаблоны) хранится в Git-репозитории.

Резервное копирование:

```
#!/bin/bash
BACKUP_DIR="/backup/ansible/$(date +%Y%m%d)"
mkdir -p "$BACKUP_DIR"

# Клонировать репозиторий (если есть доступ)
git clone --depth 1 /path/to/ansible-repo "$BACKUP_DIR/ansible-repo"

# Или скопировать локальную версию
# cp -r /path/to/ansible/ "$BACKUP_DIR/ansible-repo"
```

Периодичность: при изменениях в репозитории.

Убедитесь, что репозиторий Ansible доступен для восстановления. Рекомендуется использовать внешний Git (GitLab, GitHub) с доступом извне.

Восстановление

Порядок восстановления

Порядок восстановления компонентов выполняется в следующей последовательности:

1. ClickHouse Keeper — координация (host).
2. ClickHouse — хранилище (host).
3. ETCD — HA для PostgreSQL (Docker).
4. PostgreSQL — данные (Docker).
5. S3 MinIO — объектное хранилище (Docker).
6. Vault — секреты (Docker).
7. Stateless-службы с помощью Ansible (Saturn, Mercury, Earth, Venus, Grafana).

ClickHouse и Keeper развернуты на отдельных виртуальных машинах (внешний слой хранения). Сначала восстанавливаются они, затем Docker-компоненты, после чего выполняется повторное развертывание stateless-служб.

ClickHouse Keeper

Восстановление на том же сервере выполняется из сохраненной конфигурации и данных:

```
# Остановить Keeper
systemctl stop clickhouse-keeper

# Восстановить конфигурацию и данные
cp -r /backup/keeper/YYYYMMDD/config/* /etc/clickhouse-keeper/
cp -r /backup/keeper/YYYYMMDD/coordination/* /var/lib/clickhouse/coordination/

# Запустить
systemctl start clickhouse-keeper
```

Для замены узла в кластере необходимо:

1. Развернуть новый ClickHouse Keeper с тем же **server_id**.
2. Настроить **raft_configuration** в **keeper_server.xml**. После подключения узла Leader автоматически отправит снапшот.

ClickHouse

Для восстановления ClickHouse сначала возвращается конфигурация, после чего при необходимости восстанавливаются данные.

Восстановление конфигурации:

```
# Остановить ClickHouse
systemctl stop clickhouse-server

# Восстановить конфигурацию
cp -r /backup/clickhouse/YYYYMMDD/config/* /etc/clickhouse-server/

# Запустить
systemctl start clickhouse-server
```

Восстановление данных (через clickhouse-backup):

```
# Скачать из S3 (если нужно)
clickhouse-backup download backup_name

# Восстановить
clickhouse-backup restore backup_name
```

ETCD

Восстановление ETCD выполняется из снапшота:

```
# Остановить Patroni на всех узлах
docker stop patroni
docker stop etcd

# Очистить volume
docker run --rm -v logiq_etcd-data:/etcd-data alpine rm -rf /etcd-data/*

# Восстановить из снапшота
ETCDCTL_API=3 docker run --rm \
-v logiq_etcd-data:/etcd-data \
-v /backup/etcd/YYYYMMDD:/backup \
```

```
quay.io/coreos/etcd:v3.5.0 \
etcdctl snapshot restore /backup/snapshot.db \
--data-dir=/etcd/data
```

```
# Запустить
docker start etcd
docker start patroni
```

Восстановление необходимо выполнять при остановленном Patroni на всех узлах, иначе возможно состояние split-brain.

PostgreSQL

Восстановление PostgreSQL выполняется из дампа или из Docker volume.

Восстановление из дампа:

```
# Полное восстановление
cat /backup/postgresql/YYYYMMDD/full_backup.sql.gz | gunzip | docker exec -i patroni psql -U postgres
```

```
# Восстановление конкретной БД (Earth)
cat /backup/postgresql/YYYYMMDD/earth_backup.sql.gz | gunzip | docker exec -i patroni psql -U postgres -d earth
```

```
# Восстановление пользователей
cat /backup/postgresql/YYYYMMDD/globals.sql.gz | gunzip | docker exec -i patroni psql -U postgres
```

Восстановление из volume:

```
docker stop patroni
docker run --rm -v logiq_patroni-data:/pgdata alpine rm -rf /pgdata/*
tar -xzf /backup/postgresql/YYYYMMDD/postgres-volume.tar.gz \
-C /var/lib/docker/volumes/logiq_patroni-data/_data
docker start patroni
```

S3 MinIO

Восстановление MinIO из Docker volume:

```
docker stop minio
docker run --rm -v logiq_minio-data:/data alpine rm -rf /data/*
tar -xzf /backup/minio/YYYYMMDD/minio-volume.tar.gz \
-C /var/lib/docker/volumes/logiq_minio-data/_data
docker start minio
```

Vault

Восстановление:

```
# Остановить Vault
docker stop vault
```

```
# Восстановить снапшот
docker exec vault vault operator raft snapshot restore /backup/vault/YYYYMMDD/vault_snapshot.db
```

```
# Запустить
docker start vault
```

Stateless-службы

Saturn, Mercury, Earth, Venus, Grafana не имеют состояния. Они восстанавливаются развертыванием: **ansible-playbook AIO-docker-distributed-online.yml**.

Аварийные сценарии

Потеря узла ClickHouse Keeper

Некорректное поведение:

- ошибки координации в логах;
- таблицы в режиме **read-only**.

Решение:

1. Проверить кворум (минимум 2 из 3 узлов).
2. Если кворум есть — заменить узел:

- развернуть ClickHouse Keeper с тем же **server_id**;
 - настроить параметр **raft_configuration**. После подключения узла Leader автоматически отправит снапшот.
3. Если кворума нет — восстановить из резервной копии.

Потеря узла ETCD

Некорректное поведение:

- Patroni не выбирает лидера;
- ошибки подключения к ETCD.

Решение:

1. Проверить кворум (минимум 2 из 3 узлов).
2. Если кворум есть — добавить новый узел.
3. Если кворума нет — восстановить из снапшота (см. раздел [Восстановление ETDC](#)).

Полное восстановление кластера

Некорректное поведение:

Потеря всех узлов кластера.

Решение:

1. Восстановить ClickHouse Keeper (минимум 1 узел, host).
2. Восстановить ClickHouse (конфигурация, host).
3. Восстановить ETCD (снапшот, Docker).
4. Восстановить PostgreSQL (из дампа, Docker).
5. Восстановить S3 MinIO (volume, Docker).
6. Восстановить Vault (снапшот, Docker).
7. Развернуть stateless-службы с помощью Ansible (Saturn, Mercury, Earth, Venus, Grafana).

Внешний слой хранения (ClickHouse, Keeper, ETCD, MinIO) — это отдельные VM или Docker-контейнеры, на которых держится вся инфраструктура. Сначала восстанавливаем их, после чего разворачиваем службы с помощью Ansible.

Регламент проверок

Ежедневно

- Проверить успешность резервного копирования.
- Проверить свободное место в хранилище резервных копий.
- Проверить доступность всех узлов.

Еженедельно

- Тестовое восстановление ClickHouse (конфигурация).
- Тестовое восстановление ETCD (снапшот).
- Проверка целостности резервных копий (checksum).
- Проверка доступности репозитория Ansible.

Ежемесячно

- Полное тестовое восстановление на стенде.
- Проверка замены узла Keeper.
- Проверка замены узла ETCD.
- Тест восстановления PostgreSQL.

После обновлений

- Проверить совместимость резервных копий после обновления ClickHouse.
- Актуализировать документ (если изменились пути или команды).

Приложение

Скрипт полного резервного копирования:

```
#!/bin/bash
# backup-all.sh
```

```
DATE=$(date +%Y%m%d_%H%M%S)
BACKUP_ROOT="/backup"
```

```
echo "=== Начало резервного копирования Логикор ==="
```

```

# ClickHouse (host, конфигурация)
echo "[1/7] Резервное копирование ClickHouse..."
mkdir -p "$BACKUP_ROOT/clickhouse/$DATE"
cp -r /etc/clickhouse-server/ "$BACKUP_ROOT/clickhouse/$DATE/config/"

# ClickHouse Keeper (host)
echo "[2/7] Резервное копирование ClickHouse Keeper..."
mkdir -p "$BACKUP_ROOT/keeper/$DATE"
cp -r /etc/clickhouse-keeper/ "$BACKUP_ROOT/keeper/$DATE/config/"
cp -r /var/lib/clickhouse/coordination/ "$BACKUP_ROOT/keeper/$DATE/coordination/"

# ETCD (Docker)
echo "[3/7] Резервное копирование ETCD..."
mkdir -p "$BACKUP_ROOT/etcd/$DATE"
docker stop etcd
ETCDCTL_API=3 docker run --rm \
-v logiq_etcd-data:/etcd-data \
-v "$BACKUP_ROOT/etcd/$DATE":/backup \
quay.io/coreos/etcd:v3.5.0 \
etcdctl snapshot save /backup/snapshot.db \
--data-dir=/etcd-data
docker start etcd

# PostgreSQL (Docker, Patroni)
echo "[4/7] Резервное копирование PostgreSQL..."
mkdir -p "$BACKUP_ROOT/postgresql/$DATE"
docker exec patroni pg_dumpall -U postgres | gzip > "$BACKUP_ROOT/postgresql/$DATE/full_backup.sql.gz"
docker exec patroni pg_dumpall --globals-only | gzip > "$BACKUP_ROOT/postgresql/$DATE/globals.sql.gz"

# S3 MinIO (Docker)
echo "[5/7] Резервное копирование MinIO..."
mkdir -p "$BACKUP_ROOT/minio/$DATE"
docker stop minio
tar -czf "$BACKUP_ROOT/minio/$DATE/minio-volume.tar.gz" \
-C /var/lib/docker/volumes/logiq_minio-data/_data .
docker start minio

# Vault (Docker)
echo "[6/7] Резервное копирование Vault..."
mkdir -p "$BACKUP_ROOT/vault/$DATE"
docker exec vault vault operator raft snapshot save "$BACKUP_ROOT/vault/$DATE/vault_snapshot.db"

# Ansible (конфигурация)
echo "[7/7] Резервное копирование Ansible..."
mkdir -p "$BACKUP_ROOT/ansible/$DATE"
git clone --depth 1 /path/to/ansible-repo "$BACKUP_ROOT/ansible/$DATE/ansible-repo" 2>/dev/null || \
cp -r /path/to/ansible/ "$BACKUP_ROOT/ansible/$DATE/ansible-repo"

echo "=== Резервное копирование завершено ==="
echo ""
echo "Stateless-сервисы (Saturn, Mercury, Earth, Venus, Grafana) не подлежат резервному копированию."
echo "Восстанавливаются с помощью Ansible: ansible-playbook AIO-docker-distributed-online.yml"

```

Чек-лист восстановления

1. Оценить масштаб инцидента.
2. Определить затронутые компоненты.
3. Восстановить Keeper (если затронут).
4. Восстановить ClickHouse (если затронут).
5. Восстановить ETCD (если затронут).
6. Восстановить PostgreSQL (если затронут).
7. Восстановить S3 MinIO (если затронут).
8. Восстановить Vault (если затронут).
9. Развернуть stateless-службы с помощью Ansible.
10. Проверить работоспособность системы.
11. Задokumentировать инцидент.

Дополнительные материалы

1. ClickHouse Backup: <https://clickhouse.com/docs/operations/backup/overview>
2. ClickHouse Keeper: <https://clickhouse.com/docs/guides/sre/keeper/clickhouse-keeper>
3. clickhouse-backup: <https://github.com/Altinity/clickhouse-backup>
4. ETCD backup: <https://etcd.io/docs/latest/op-guide/recovery/>
5. PostgreSQL backup: <https://www.postgresql.org/docs/current/backup.html>

ID статьи: 1563

Последнее обновление: 29 апр., 2026

Обновлено от: Егоров В.

Ревизия: 12

База знаний Логикор -> Документация -> Регламент резервного копирования и восстановления Логикор

<https://docs.axel.pro/entry/1563/>