

# Проекты VK WorkSpace

Восстановление из резервной копии

# Оглавление

---

Восстановление разделов docker-контейнеров	3
Восстановление баз данных PostgreSQL из дампов	3

Данный раздел описывает процедуру восстановления данных для **Проекты VK WorkSpace** в случае системного сбоя.

## ВНИМАНИЕ

Для восстановления работы нужно [установить "чистую" версию Проекты VK WorkSpace](#)

Перед восстановлением убедитесь, что система чистая и работает корректно

Восстановление данных из резервной состоит из двух шагов и выполняется в порядке, обратном созданию резервных копий.

1. Восстановление разделов docker-контейнеров.
2. Восстановление баз данных Postgres из ранее созданных дампов.

# Восстановление разделов docker-контейнеров

Для восстановления разделов контейнеров нужно иметь привилегии пользователя `root`, так как по умолчанию только `root` имеет доступ к этим разделам файловой структуры.

1. Создайте временную директорию и перейдите в нее. Затем скопируйте сюда архив, содержащий разделы контейнеров докер:

```
mkdir /tmp/volumes_backup
cd /tmp/volumes_backup
scp backup-server:/${ shell_product_name }/volumes_backup.tgz .
tar -xzf volumes_backup.tgz
${ shell_product_name }_minio
${ shell_product_name }_ssl-volume
${ shell_product_name }_trusted-certificates-volume
rm -f volumes_backup.tgz
```

2. Перенесите директории в папку с разделами `docker`. По умолчанию это `/var/lib/docker/volumes/`.

```
mv ${ shell_product_name }_minio \
    ${ shell_product_name }_ssl-volume \
    ${ shell_product_name }_trusted-certificates-volume \
    /var/lib/docker/volumes/
```

## ВНИМАНИЕ

Пользователь `root` больше не нужен, остальные операции можно делать из-под "обычного" пользователя

# Восстановление баз данных PostgreSQL из дампов

Данная процедура описывает баз данных PostgreSQL из предварительно созданных дампов. Процедура автоматизирована и для запуска нужно выполнить следующие действия:

1. Перейдите в директорию `scripts`, которая находится внутри директории с артефактами **Проекты VK Workspace**:

```
cd scripts
ls -la
-rwxrwx--- 1 user user 1275 Feb 02 11:22 db_backup.sh*
-rwxrwx--- 1 user user 1652 Feb 02 11:22 db_restore.sh*
```

2. Создайте здесь директорию  `dumps` , сохраните туда архив и распакуйте его:

```
mkdir dumps
ls -la
-rwxrwx--- 1 user user 1275 Feb 02 11:22 db_backup.sh*
-rwxrwx--- 1 user user 1652 Feb 02 11:22 db_restore.sh*
drwxr-xr-x 2 user user 4096 Feb 02 13:26 dumps/
scp backup-server:/{ shell_product_name }/{ shell_product_name }
_2025-02-02_04-00-00_backup.tgz ./dumps/
cd dumps
tar -xzf { shell_product_name }_2025-02-02_04-00-00_backup.tgz
```

3. Запустите скрипт  `db_restore.sh` , указав целевую директорию и общий суффикс, состоящий из временной метки и расширения, например, для файлов:

```
{ shell_product_name }_migrations_db_2025-02-02_04-00-00.bak
{ shell_product_name }_attachment_db_2025-02-02_04-00-00.bak
{ shell_product_name }db_2025-02-02_04-00-00.bak
```

Общим суффиксом будет  `2025-02-02_04-00-00.bak` . Таким образом, команда для восстановления будет выглядеть следующим образом:

```
./db_restore.sh ./dumps 2025-02-02_04-00-00.bak
```

#### ВНИМАНИЕ

Обращайте внимание на лог восстановления, при необходимости перенаправьте поток вывод в файл для дальнейшего анализа.

Пример успешного восстановления:


```
...
Recovering from backup: container: { shell_product_name }-database_service-1
{ shell_product_name }db { shell_product_name }db_02-02-2025_04_00.bak
UPDATE 1
pg_terminate_backend
-----
 t
 t
 t
 t
(4 rows)

DROP DATABASE
CREATE DATABASE
```

После успешного восстановления перезапустите  `{ shell_product_name }` :

```
docker compose -p { shell_product_name } restart
```

 Технический писатель: Белова Ирина

 6 мая 2026 г.