



МойОфис Частное Облако 3

В ВАРИАНТЕ ИСПОЛНЕНИЯ ГОСТ

Руководство по резервному копированию

ООО «НОВЫЕ ОБЛАЧНЫЕ ТЕХНОЛОГИИ»

**ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ
«МОЙОФИС ЧАСТНОЕ ОБЛАКО 3»
В ВАРИАНТЕ ИСПОЛНЕНИЯ ГОСТ
3.1G**

РУКОВОДСТВО ПО РЕЗЕРВНОМУ КОПИРОВАНИЮ

Версия 1

На 18 листах

Дата публикации: 12.09.2024

**Москва
2024**

МойОфис

Все упомянутые в этом документе названия продуктов, логотипы, торговые марки и товарные знаки принадлежат их владельцам.

Товарные знаки «МойОфис» и «MyOffice» принадлежат ООО «НОВЫЕ ОБЛАЧНЫЕ ТЕХНОЛОГИИ».

Ни при каких обстоятельствах нельзя истолковывать любое содержимое настоящего документа как прямое или косвенное предоставление лицензии или права на использование товарных знаков, логотипов или знаков обслуживания, приведенных в нем. Любое несанкционированное использование этих товарных знаков, логотипов или знаков обслуживания без письменного разрешения их правообладателя строго запрещено.

СОДЕРЖАНИЕ

1 Общие сведения	6
2 Резервное копирование служб	7
2.1 ArangoDB	8
2.1.1 Резервное копирование данных	8
2.1.2 Восстановление данных	10
2.2 PostgreSQL	12
2.2.1 Резервное копирование данных	12
2.2.2 Восстановление данных	14
2.3 Файлы Системы хранения данных	16
2.3.1 Резервное копирование данных	16
2.3.1.1 Резервирование при использовании хранилища FS	16
2.3.1.2 Резервирование при использовании хранилища MinIO	17
2.3.2 Восстановление данных	18
2.4 Elasticsearch	18
2.5 Redis	18

ПЕРЕЧЕНЬ СОКРАЩЕНИЙ, ТЕРМИНОВ И ОПРЕДЕЛЕНИЙ

В настоящем документе применяют следующие сокращения с соответствующими расшифровками (см. Таблицу 1).

Таблица 1 — Сокращения и расшифровки

Сокращение, термин	Расшифровка и определение
Ansible	Система управления конфигурациями, используемая для автоматизации настройки и развертывания программного обеспечения
Inventory file	Файл Ansible с перечислением ролей и их IP-адресов
MD5-хеш (hash)	Контрольная сумма, предназначенная для проверки целостности файла
PGS	Pythagoras, система хранения данных в составе «МойОфис Частное Облако 3» в варианте исполнения ГОСТ
БД	База данных
Вендор (vendor)	Поставщик брендированного продукта
ЕСИА	Единая система идентификации и аутентификации
Кластер (cluster)	Объединенная группа серверов
ОС	Операционная система
Персистентность	Свойство структуры данных, сохраняющее свои состояния и доступ к этим состояниям
ПО	Программное обеспечение
Тенант (tenant)	Логический объект, включающий в себя совокупность вычислительных ресурсов, пользователей и репозиторий
Хост (Host)	Устройство, предоставляющее сервисы формата «клиент-сервер»

1 ОБЩИЕ СВЕДЕНИЯ

«МойОфис Частное Облако 3» в варианте исполнения ГОСТ — комплекс безопасных веб-сервисов и приложений для организации хранения, доступа и совместной работы с файлами и документами внутри компании, использующих отечественные средства криптографической защиты информации. Взаимодействие всех клиентских приложений с серверными системами осуществляется по сетевым каналам, защищенным с помощью протокола TLS с использованием отечественной криптографии.

В состав продукта входят:

- Система хранения данных для безопасного хранения корпоративных файлов и обеспечения возможностей авторизации, аутентификации и разграничения прав доступа пользователей;
- Система редактирования и совместной работы для индивидуального и совместного редактирования презентаций, текстовых и табличных документов;
- Административная панель системы хранения для управления пользователями, группами, общими папками, доменами и тенантами.

В состав продукта входят следующие приложения для работы в веб-браузерах и на мобильных устройствах:

- «МойОфис Документы» — веб-приложение для организации структурированного хранения файлов, выполнения операций с файлами и папками, настройки совместного доступа;
- «МойОфис Текст» — веб-редактор для быстрого и удобного создания и форматирования текстовых документов любой сложности;
- «МойОфис Таблица» — веб-редактор для создания электронных таблиц, ведения расчетов, анализа данных и просмотра сводных отчетов;
- «МойОфис Презентация» — веб-редактор для создания, оформления и демонстрации презентаций;
- «МойОфис Документы» для мобильных платформ — приложение для просмотра и редактирования текстовых документов, электронных таблиц и презентаций, просмотра PDF-файлов, а также доступа к облачным хранилищам на смартфонах и планшетах с ОС Android, iOS и iPadOS.

Подробное описание возможностей продукта приведено в документе «"МойОфис Частное Облако 3" в варианте исполнения ГОСТ. Функциональные возможности».

2 РЕЗЕРВНОЕ КОПИРОВАНИЕ СЛУЖБ

«Горячий» режим резервного копирования позволяет выполнять резервирование без остановки работы служб Системы хранения данных. Рекомендуется ограничить доступ пользователей к базам данных и сервисам приложения перед резервированием или выбрать период наименьшей загрузки.

Резервное копирование необходимо выполнять последовательно для всех служб из-за взаимосвязи баз данных.

2.1 ArangoDB

База данных ArangoDB содержит метаданные файлов и права доступа к ним.

2.1.1 Резервное копирование данных

Для резервного копирования сервера ArangoDB предлагается использовать следующий сценарий:

```
#!/bin/bash
# Backup ArangoDB database into a daily file.
#
set -e
BACKUP_DIR="<>BACKUP_DIR">"
CURRENT_DATETIME="$(date +%Y_%m_%d_%H%M)"
KEEP_BACKUPS=<RETENTION_TIME>
ARANGODB_IMAGE="pythagoras/pgs-arangodb"
ARANGODB_VERSION="<>VERSION">"
ARANGODB_HOST="<>HOST">"
ARANGODB_PORT="<>PORT">"
ARANGODB_PASS="<>PASSWORD">"
RC=0

docker run \
    --name arangodb_backup \
    --network pgs-network \
    --rm \
    -v "${BACKUP_DIR}:${BACKUP_DIR}" \
    pgs-private-registry:5001/${ARANGODB_IMAGE}:${ARANGODB_VERSION} \
    arangodump \
    --all-databases true \
    --compress-output false \
    --include-system-collections true \
    --output-directory \
    "${BACKUP_DIR}/arangodb_$(date +%Y_%m_%d_%H%M)" \
    --server.authentication true \
    --server.endpoint "tcp://${ARANGODB_HOST}:${ARANGODB_PORT}" \
    --server.password ${ARANGODB_PASS} \
    --server.username 'root' || RC=1

# Do tar.gz of backup folder
cd "${BACKUP_DIR}" || RC=1
tar -czf "arangodb_${CURRENT_DATETIME}.tar.gz" \
"arangodb_${CURRENT_DATETIME}" || RC=1
rm -rf "${BACKUP_DIR}/arangodb_${CURRENT_DATETIME}" || RC=1

# Delete archives older then KEEP_BACKUPS value
find "${BACKUP_DIR:?}"/ -mtime +"${KEEP_BACKUPS}" \
    -name "*.tar.gz" -exec rm -f {} \; || RC=1

if [ ${RC} = 0 ]
then
    echo "Successfully backup"
else
    echo "Backup Failed"
fi
```

Для выполнения резервного копирования необходимо присвоить значения переменным, перечисленным в таблице 2. Остальные настройки следует оставить без изменений.

Таблица 2 — Обязательные переменные для резервного копирования ArangoDB

Наименование переменной	Описание	Пример
<BACKUP_DIR>	Путь к месту хранения копий	"/opt/Pythagoras/backups/arangodb"
<RETENTION_TIME>	Глубина хранения (вводится без кавычек)	целое число, например "7"
<VERSION>	Версия компонента в релизе	"3.8.8"
<HOST>	Хост подключения	"coordinator" — при кластерной установке "arangodb" — при установке без отказоустойчивости
<PORT>	Порт подключения	"8529"
<PASSWORD>	Пароль, значение переменной ARANGODB_PASSWORD из файла inventory установки Системы хранения данных	

2.1.2 Восстановление данных

Для восстановления данных ArangoDB предлагается использовать сценарий, представленный ниже:

```
#!/bin/bash
#
# Restore ArangoDB database into a daily file.
#
set -e

BACKUP_DIR="<>
BACKUP_NAME="<>
ARANGODB_IMAGE="pythagoras/pgs-arangodb"
ARANGODB_VERSION="<>
ARANGODB_PASS="<>
ARANGODB_HOST="<>
ARANGODB_PORT="<>
RC=0

# Do tar.gz of backup folder
cd "${BACKUP_DIR}" || RC=1
tar -xf "${BACKUP_NAME}.tar.gz" -C "${BACKUP_DIR}" || RC=1

docker run \
  --name arangodb_restore \
  --network pgs-network \
  --rm \
  -v "${BACKUP_DIR}:${BACKUP_DIR}" \
  pgs-private-registry:5001/${ARANGODB_IMAGE}:${ARANGODB_VERSION} \
  arangorestore \
  --all-databases true \
  --create-database true \
  --input-directory "${BACKUP_DIR}/${BACKUP_NAME}" \
  --server.authentication true \
  --server.endpoint "tcp://${ARANGODB_HOST}:${ARANGODB_PORT}" \
  --server.password "${ARANGODB_PASS}" \
  --server.username 'root' || RC=1
if [ ${RC} = 0 ]
then
  echo "Backup was restored successfully "
else
  echo "Restoring backup was failed"
fi
```

Для восстановления данных необходимо присвоить значения переменным, перечисленным в таблице 3. Остальные настройки следует оставить без изменений.

Таблица 3 — Обязательные переменные для резервного копирования ArangoDB

Наименование переменной	Описание	Пример
<BACKUP_DIR>	Путь к месту хранения копий	"/opt/Pythagoras/backups/arangodb"
<RETENTION_TIME>	Глубина хранения (вводится без кавычек)	целое число, например "7"
<VERSION>	Версия компонента	"3.8.8"
<HOST>	Сервер подключения	"coordinator" — при кластерной установке "arangodb" — при установке без отказоустойчивости
<PORT>	Порт подключения	"8529"
<PASSWORD>	Пароль, значение переменной ARANGODB_PASSWORD из файла inventory установки Системы хранения данных	

2.2 PostgreSQL

Служба Postgres (PostgreSQL) используется в Системе хранения данных для хранения данных Keycloak — Single Sign-On сервиса, входящего в состав системы. В PostgreSQL хранятся пользовательские данные, информация о тенантах и права доступа.

2.2.1 Резервное копирование данных

Для резервного копирования сервера PostgreSQL предлагается использовать следующий сценарий:

```
#!/bin/bash
#
# Backup a Postgresql database into a daily file.
#
set -e
BACKUP_DIR="<<BACKUP_DIR>"
POSTGRES_IMAGE="pythagoras/pgs-postgres"
POSTGRES_VERSION="<<VERSION>"
POSTGRES_HOST="<<HOST>"
POSTGRES_PORT="<<PORT>"
DATABASE=keycloak
DATABASE_USER_PASS="<<PASSWORD>"
DATABASE_USER=keycloak
CURRENT_DATETIME="$(date +%Y_%m_%d_%H%M)"
KEEP_BACKUPS=<RETENTION_TIME>
FILE=postgres_${CURRENT_DATETIME}
OUTPUT_FILE=${BACKUP_DIR}/${FILE}
RC=0

docker run \
  --name postgres_backup \
  --network pgs-network \
  --env PGPASSWORD=${DATABASE_USER_PASS} \
  --rm \
  -v "${BACKUP_DIR}:${BACKUP_DIR}" \
  pgs-private-registry:5001/${POSTGRES_IMAGE}:${POSTGRES_VERSION} \
  pg_dump -Fc \
  -h ${POSTGRES_HOST} \
  -p ${POSTGRES_PORT} \
  -U ${DATABASE_USER} \
  ${DATABASE} \
  -f ${OUTPUT_FILE}.gz \
  -v || RC=1

# Delete archives older then KEEP_BACKUPS value
find "${BACKUP_DIR}"/ -mtime +"${KEEP_BACKUPS}" -name "*.gz" -exec rm -f {} \; ||
RC=1

if [ ${RC} = 0 ]
then
  echo "Successfully backup"
else
  echo "Backup Failed"
fi
```

Для выполнения резервного копирования необходимо присвоить значения переменным, перечисленным в таблице 4. Остальные настройки следует оставить без изменений.

Таблица 4 — Обязательные переменные для резервного копирования PostgreSQL

Наименование переменной	Описание	Пример
<BACKUP_DIR>	Путь к месту хранения копий	"/opt/Pythagoras/backups/postgres"
<RETENTION_TIME>	Глубина хранения (вводится без кавычек)	целое число, например "7"
<VERSION>	Версия компонента в релизе	"12.18"
<HOST>	Хост подключения	"haproxy" — при кластерной установке "postgres" — при установке без отказоустойчивости
<PORT>	Порт подключения	"5000" — при кластерной установке "5432" — при установке без отказоустойчивости
<PASSWORD>	Пароль пользователя Keycloak, переменная KEYCLOAK_POSTGRES_PASSWORD из файла inventory установки Системы хранения данных	

2.2.2 Восстановление данных

Для восстановления данных PostgreSQL предлагается использовать сценарий, представленный ниже:

```
#!/bin/bash
#
# Restore a Postgresql database.
#
set -e
BACKUP_DIR="<>BACKUP_DIR>"
DATABASE=keycloak
POSTGRES_IMAGE="pythagoras/pgs-postgres"
POSTGRES_HOST="<>HOST>"
POSTGRES_PORT="<>PORT>"
POSTGRES_VERSION="<>VERSION>"
USER=keycloak
DATABASE_USER_PASS="<>PASSWORD>"
CURRENT_DATETIME="$(date +%Y_%m_%d_%H%M)"

FILE_NAME="<>NAME>"

docker run \
  --name postgres_backup \
  --network pgs-network \
  --env PGPASSWORD=${DATABASE_USER_PASS} \
  --rm \
  -v "${BACKUP_DIR}:${BACKUP_DIR}" \
  pgs-private-registry:5001/${POSTGRES_IMAGE}:${POSTGRES_VERSION} \
  pg_restore -v --clean \
  -h ${POSTGRES_HOST} \
  -p ${POSTGRES_PORT} \
  -U ${USER} \
  -d ${DATABASE} \
  ${BACKUP_DIR}/${FILE_NAME}
```

Для восстановления данных необходимо присвоить значения переменным, перечисленным в таблице 5. Остальные настройки следует оставить без изменений.

Таблица 5 — Обязательные переменные для резервного копирования PostgreSQL

Наименование переменной	Описание	Пример
<BACKUP_DIR>	Путь к месту хранения копий	"/opt/Pythagoras/backups/postgres"
<HOST>	Сервер подключения	"haproxy" — при кластерной установке "postgres" — при установке без отказоустойчивости
<PORT>	Порт подключения	"5000" — при кластерной установке "5432" — при установке без отказоустойчивости
<VERSION>	Версия компонента в релизе	"12.18"
<PASSWORD>	Пароль пользователя Keycloak, переменная KEYCLOAK_POSTGRES_PASSWORD из файла inventory установки Системы хранения данных	
<NAME>	Полное имя файла резервной копии	"postgres_2023_03_26_1435.gz"

2.3 Файлы Системы хранения данных

2.3.1 Резервное копирование данных

2.3.1.1 Резервирование при использовании хранилища FS

При выборе хранилища типа FS директория хранения файлов на серверах с ролью `storage` будет располагаться по пути `/media/storage`.

Перед началом резервирования необходимо убедиться, что путь, предназначенный для резервного копирования, создан и доступен.

Для копирования файлов на сторонний сервер необходимо выполнить следующую команду:

```
rsync -vrah /media/storage/ remote_host: <BACKUP> --delete --inplace \  
--whole-file --progress -B 131072
```

где: `<BACKUP>` — путь к месту хранения копий (например: `/backup_folder`).

Допускается использование средств виртуальной инфраструктуры:

- механизмов полного копирования виртуальных машин;
- приложений, предоставляющих средства резервного копирования для элементов виртуальных инфраструктур (например: Veeam Backup).

2.3.1.2 Резервирование при использовании хранилища MinIO

При использовании в качестве хранилища S3 сервиса MinIO, находящегося в составе дистрибутива, следует использовать сценарий, представленный ниже:

```
#!/bin/bash
set -e
BACKUP_DIR="<BACKUP_DIR>"
MINIO_IMAGE="<MINIO_IMAGE_NAME>"
MINIO_VERSION="<VERSION>"
MINIO_URL="<MINIO_URL>"
MINIO_ACCESS_KEY="<ACCESS_KEY>"
MINIO_SECRET_KEY="<SECRET_KEY>"
MINIO_ALIAS_NAME="<MINIO_ALIAS_NAME>"
MINIO_BUCKET="<MINIO_BUCKET>"
RC=0
docker run --net=host -it --entrypoint=bash -v ${BACKUP_DIR}:${BACKUP_DIR} \
pgs-private-registry:5001/${MINIO_IMAGE}:${MINIO_VERSION} -c \
"mc alias set ${MINIO_ALIAS_NAME} ${MINIO_URL} ${MINIO_ACCESS_KEY} \
${MINIO_SECRET_KEY};"
mc mirror --remove --preserve ${MINIO_ALIAS_NAME}/${MINIO_BUCKET} $BACKUP_DIR/" \
|| RC=1
if [ ${RC} = 0 ]
then
echo "Successfully backup"
else
echo "Backup Failed"
fi
```



В каталоге **<BACKUP_DIR> перед копированием будут удалены все данные.**

Для выполнения резервного копирования необходимо присвоить значения переменным, перечисленным в таблице 6. Остальные настройки следует оставить без изменений.

Таблица 6 — Обязательные переменные для резервного копирования ArangoDB

Наименование переменной	Описание	Пример
<BACKUP_DIR>	Путь к месту хранения копий	"/opt/Pythagoras/backup"
<MINIO_IMAGE>	Наименование образа компонента в релизе (вводится без кавычек)	"minio/minio"
<VERSION>	Версия компонента в релизе	"RELEASE.2023-12-13T23-28-55Z"
<ACCESS_KEY>	Minio access key	minio
<SECRET_KEY>	Minio secret key	minio123
<MINIO_ALIAS_NAME>	Имя alias для текущего S3 сервиса (8 символов)	myminio
<MINIO_BUCKET>	Имя bucket, резервная копия которого выполняется	pgs-files

2.3.2 Восстановление данных

Данные могут быть восстановлены с помощью копирования файлов из резервной копии в директорию `/media/storage`.

2.4 Elasticsearch

ElasticSearch — служба, используемая Системой хранения данных для поиска по файлам и содержанию документов. Служба не блокирует работу системы, и в случае полной потери поискового индекса все файлы будут проиндексированы при следующем обращении.

При необходимости обязательного сохранения данных поиска (при развертывании стенда с отказоустойчивостью данного компонента) предполагается использование стороннего решения, рекомендованного вендором ПО.

2.5 Redis

Резервное копирование БД Redis в Системе хранения данных не является обязательным. Сервис используется для хранения неперсистентных данных (кеша).

После восстановления данных необходимо выполнить инициализацию ключей с помощью команды:

```
docker exec $(docker ps -qf name=aristoteles) python3 \  
/opt/Pythagoras/Aristoteles/initializers/RedisInit.py
```