Cloudmaster — резервное копирование

- Зачем делать резервное копирование?
- Рекомендации
  - Правила резервирования:
  - Объекты резервирования:
  - Частота резервирования:
  - Место хранения резервных копий:
  - Срок хранения:
  - Проверка восстановления:
- Элементы конфигурации, подлежащие резервированию
  - Конфигурации компонент Cloudmaster
  - Конфигурации Python-модулей
- Исполняемые файлы, подлежащие резервированию
  - Исполняемые компоненты Cloudmaster
  - Сервисы операционной системы для Cloudmaster
  - Исполняемый код Python-модулей
- Данные, подлежащие резервированию
- Рекомендуемые к применению технические меры
  - Резервное копирование
  - Восстановление

Зачем делать резервное копирование?

Резервное копирование необходимо для сохранности данных, конфигурационных файлов и исполняемого кода для восстановления системы в случае сбоев, потери данных или ошибок конфигурации.

Рекомендации

Правила резервирования:

- **три** резервные копии
- **две** из которых хранятся в различных средах (местах, носителях)
- **одна** копия из которых гарантированно размещается во внешней системе хранения

Объекты резервирования:

- вся виртуальная машина (снепшоты и полный бэкап)
- дампы баз данных (см. ниже) из сервиса PostgreSQL

Частота резервирования:

• Полное резервное копирование: раз в сутки (ночью, например, в 02:00).

• Инкрементное резервное копирование: каждые 12 часов.

Место хранения резервных копий:

• Локальное хранилище (не применимо для резервирования всей ВМ): /backups/ на отдельном (от всех прочих) диске или разделе. Локальное хранилище рекомендуется использовать только, как оперативное временное хранилище.

• Внешнее хранилище: облачное хранилище или выделенный сервер для бэкапов (например, с доступом по протоколам S3-объектов или FTP).

Срок хранения:

• Ежедневные копии: хранить 7 дней.

• Еженедельные копии: хранить 4 недели.

• Ежемесячные копии: хранить 6 месяцев.

Проверка восстановления:

• Каждые две недели тестировать процесс восстановления на отдельном тестовом сервере

• Еженедельно проверять целостность резервных копий

Элементы конфигурации, подлежащие резервированию

Конфигурации компонент Cloudmaster

Отдельные файлы:

- /opt/cloudmaster/services/auth/application.properties
- /opt/cloudmaster/services/broker/application.properties
- /opt/cloudmaster/temporal/temporal-server.yaml

Каталог */opt/cloudmaster/nginx/* со всем содержимым (в т.ч. SSL-сертификаты в подкаталоге /opt/cloudmaster/nginx/ssl)

Конфигурации Python-модулей

Точный перечень конфигурационных файлов для Python-модулей уточняется с разработчиками Python-модулей.

Рекомендуется резервировать полностью весь каталог */opt/cloudmaster/services/activity* со всем содержимым (содержит, как исполняемые .py-файлы, так и необходимые конфигурационные файлы)

Исполняемые файлы, подлежащие резервированию

Исполняемые компоненты Cloudmaster

Отдельные файлы:

- /opt/cloudmaster/services/auth/cm-idm.jar
- /opt/cloudmaster/services/broker/cm-broker.jar
- /opt/cloudmaster/temporal/bin/temporal
- /opt/cloudmaster/temporal/bin/temporal-server

Каталог */opt/cloudmaster/www* со всем содержимым - содержит статичные файлы для SPA пользовательского интерфейса (административный и клиентский порталы)

Сервисы операционной системы для Cloudmaster

Отдельные systemd-файлы из каталога */usr/lib/systemd/system/*:

- /usr/lib/systemd/system/cloudmaster-auth.service
- /usr/lib/systemd/system/cloudmaster-broker.service
- /usr/lib/systemd/system/temporal.service
- /usr/lib/systemd/system/cloudmaster-activity.service

Исполняемый код Python-модулей

Точный перечень необходимого для резервирования исполняемого кода для Python-модулей уточняется с разработчиками Python-модулей.

Рекомендуется резервировать полностью весь каталог */opt/cloudmaster/services/activity* со всем содержимым (содержит, как исполняемые .py-файлы, так и необходимые конфигурационные файлы)

Данные, подлежащие резервированию

В базе данных, размещаемой в отдельном сервисе на основе PostgreSQL, резервированию подлежат дампы баз данных:

- cloudmaster (точное имя базы данных необходимо уточнить т.к. при начальной установке может быть задано произвольное имя)
- temporal
- temporal_visibility

Рекомендуемые к применению технические меры

Резервное копирование

1. Автоматизация через скрипты:
Создать bash-скрипт для автоматического выполнения резервного копирования всех элементов конфигурации, исполняемых компонент/кода и сервисов

2. Использование специализированных инструментов резервного копирования:

- rsync для синхронизации с удаленным сервером.
- rclone для синхронизации с облачными хранилищами (например S3).
- pg_dump для снятия дампа БД PostgreSQL.

```
#!/bin/bash

# Настройки
BACKUP_DIR="/var/backups/cloudmaster"  # Локальный каталог для хранения
резервных копий
REMOTE_SERVER="user@remote-server:/path/to/backup"  # Удаленный сервер для
rsync
S3_BUCKET="yandex-s3-bucket-name"  # Название S3 бакета Яндекс Облака
PG_USER="postgres"  # Пользователь PostgreSQL
PG_DB="your_database_name"  # Имя базы данных
DATE=$(date +"%Y-%m-%d_%H-%M-%S")  # Текущая дата и время
ARCHIVE_NAME="backup_$DATE.tar.gz"  # Имя архива

# Создание каталога для локальных резервных копий (если не существует)
mkdir -p "$BACKUP_DIR"

# Шаг 1: Резервное копирование каталогов и файлов
echo "Создание архива с резервной копией..."
tar -czf "$BACKUP_DIR/$ARCHIVE_NAME"
    /opt/cloudmaster
    /usr/lib/systemd/system/cloudmaster-auth.service
    /usr/lib/systemd/system/cloudmaster-broker.service
```

```
    /usr/lib/systemd/system/temporal.service
    /usr/lib/systemd/system/cloudmaster-activity.service

if [ $? -ne 0 ]; then
    echo "Ошибка создания архива!"
    exit 1
fi
echo "Архив создан: $BACKUP_DIR/$ARCHIVE_NAME"

# Шаг 2: Резервное копирование базы данных PostgreSQL
echo "Создание дампа базы данных PostgreSQL..."
PG_DUMP_FILE="$BACKUP_DIR/postgres_backup_$DATE.sql"
pg_dump -U "$PG_USER" -F c "$PG_DB" > "$PG_DUMP_FILE"

if [ $? -ne 0 ]; then
    echo "Ошибка создания дампа базы данных!"
    exit 1
fi
echo "Дамп базы данных создан: $PG_DUMP_FILE"

# Шаг 3: Синхронизация с удаленным сервером через rsync
echo "Синхронизация с удаленным сервером..."
rsync -avz "$BACKUP_DIR/" "$REMOTE_SERVER"

if [ $? -ne 0 ]; then
    echo "Ошибка синхронизации с удаленным сервером!"
    exit 1
fi
echo "Синхронизация с удаленным сервером завершена."

# Шаг 4: Синхронизация с облачным хранилищем Яндекс Облака через rclone
echo "Синхронизация с облачным хранилищем Яндекс Облака..."
rclone copy "$BACKUP_DIR/" "s3:$S3_BUCKET"

if [ $? -ne 0 ]; then
    echo "Ошибка синхронизации с облачным хранилищем!"
    exit 1
fi
echo "Синхронизация с облачным хранилищем завершена."

# Очистка старых резервных копий (опционально)
# echo "Очистка старых резервных копий..."
# find "$BACKUP_DIR" -type f -mtime +30 -exec rm {} ;
# echo "Очистка завершена."

echo "Резервное копирование успешно завершено."
```

Пояснения:

1. Резервное копирование каталогов и файлов:
   С помощью команды tar создается архив с указанными каталогами и файлами.

2. Резервное копирование базы данных PostgreSQL:
   Используется pg_dump для создания дампа базы данных в формате SQL.

3. Синхронизация с удаленным сервером:
Команда rsync используется для передачи резервной копии на удаленный сервер.

4. Синхронизация с облачным хранилищем Яндекс Облака:
rclone должен быть предварительно настроен для работы с S3 бакетом Яндекс Облака. Для этого выполните команду rclone config и настройте подключение.

5. Очистка старых резервных копий (опционально):
В скрипте есть закомментированный блок для удаления резервных копий старше 30 дней.

Предварительная настройка:

• Убедитесь, что у вас установлены rsync, pg_dump, rclone и они настроены.

• Настройте доступ к удаленному серверу через SSH (например, с использованием ключа).

• Настройте rclone для работы с Яндекс Облаком (используя S3 API).

Запустите скрипт вручную или добавьте его в cron для автоматического выполнения.

3. Мониторинг и уведомления:
Настроить уведомления об успешности или ошибках выполнения задач резервирования через email или мессенджеры (например, Telegram).

Восстановление

1. Восстановление компонент Cloudmaster:

• Разархивировать резервные копии

• Перезапустить восстановленные сервисы

2. Восстановление NGINX:

• Разархивировать конфигурацию и сертификаты

• Проверить конфигурацию

• Перезапустить NGINX

3. Восстановление баз данных:

• Загрузить дампы в сервер для базы данных temporal, temporal_visibility, cloudmaster (точное имя базы данных необходимо уточнить т.к. при начальной установке может быть задано произвольное имя)

4. Восстановление системных файлов:

• Разархивировать сервисы systemd в каталог */usr/lib/systemd/system/*