

**ООО «БАЗИС»**

**СИСТЕМА УПРАВЛЕНИЯ ОБЛАЧНОЙ ПЛАТФОРМОЙ «ТИОНИКС»  
ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ  
РУКОВОДСТВО ПО ЭКСПЛУАТАЦИИ  
RU.NPФЛ.00005-01.97.01**

**ЛИСТОВ 18**

2021

## Оглавление

<b>Оглавление .....</b>	<b>2</b>
<b>1. Введение .....</b>	<b>4</b>
1.1. Назначение руководства .....	4
1.2. Перечень эксплуатационных документов .....	4
<b>2. Общие указания .....</b>	<b>5</b>
2.1. Резервное копирование БД.....	5
2.1.1. Инструкция по внеплановому резервному копированию БД.....	7
2.1.2. Инструкция по восстановлению резервной копии БД .....	7
2.1.3. Логирование резервного копирования .....	8
2.1.4. Порядок снятия резервного копирования.....	8
2.2. Порядок диагностики, локализации и устранения проблем .....	8
2.2.1. Диагностика, локализация и устранения проблем с доступностью серверов.....	8
2.2.2. Диагностика, локализация и устранения проблем загрузки процессора .....	10
2.2.3. Диагностика, локализация и устранения проблем СУБД.....	10
2.2.4. Процедура проверки функционирования ПО после восстановления резервной копии и перевода ПО в штатный режим функционирования .....	13
2.3. Оптимизация и настройки производительности.....	13
2.3.1. Оптимизация производительности СУБД.....	13
2.4. Порядок вывода ПО из эксплуатации .....	14
<b>3. Требования к аппаратным средствам .....</b>	<b>15</b>
<b>4. Описание функций и структура.....</b>	<b>16</b>

## **АННОТАЦИЯ**

Настоящий документ предназначен для технического администратора ПО и содержит инструкции по выполнению работ, необходимых для эксплуатации ПО.

## **1. ВВЕДЕНИЕ**

### **1.1. Назначение руководства**

Настоящее руководство по техническому обслуживанию содержит инструкции по выполнению следующих работ:

- сопровождение и обслуживание ПО;
- диагностику, локализацию и устранение проблем;
- порядок вывода ПО из эксплуатации.

### **1.2. Перечень эксплуатационных документов**

Дополнительно к настоящему документу технические администраторы должны использовать следующие документы: «Инструкция по установке ПО Система управления облачной платформой Тионикс»

## 2. ОБЩИЕ УКАЗАНИЯ

### 2.1. Резервное копирование БД

Для настройки процедуры резервного копирования на сервере БД выполните следующие команды:

1) создайте структуры директорий для резервных копий БД, используя команду:

```
$ mkdir /pgbackup/db/{backup,log,scripts}
```

2) создайте группу для резервного копирования, если ранее не создавалась, используя команду:

```
$ sudo groupadd backup-database
```

3) создайте пользователя для резервного копирования файлового хранилища, используя команду:

```
$ sudo useradd -s /bin/bash -g backup -d /pgbackup/ -m backup-database
```

4) задайте пароль для пользователя backup-storage, используя команду:

```
$ sudo passwd backup-database
```

5) задайте права на директорию, используя команду:

```
$ sudo chown -R backup-database:backup /pgbackup/
```

6) создайте скрипт для снятия резервной копии БД /pgbackup/db/scripts/pgbackup\_db.sh со следующим содержанием:

```
#!/bin/bash
DBASE="SUOP" #название базы данных передается во время вызова скрипта
DB_OWNER="backup" #пользователь, под которым происходит подключение
DB_PASS="backup" #пароль
DB_HOST="192.168.0.1" #адрес сервера СУБД
```

```
BACKUP_DIR="/pgbackup/db/backup" #путь к хранилищу резервных
копий
LOG_DIR="/pgbackup/db/log" # путь до каталога с логами
KEEP_BACKUP="120" #количество хранимых копий (4 месяца)
THREADS=3 #количество потоков
BIN='/usr/bin/pg_dump' #путь к каталогу с актуальной версией
pg_dump (PostgreSQL) 12.1
DATE=$(date +%F-%k-%M)
find $BACKUP_DIR -name "*.backup" -mtime +${KEEP_BACKUP} -delete
> ${LOG_DIR}/${DATE}.log 2>&1
```

```
PGPASSWORD=${DB_PASS} ${BIN}/pg_dump -U ${DB_OWNER} -h ${DB_HOST}
-Fd -j${THREADS} -v --blobs $DBASE --file
${BACKUP_DIR}/${DATE}_${SUOP}.pgsql.backup >>
${LOG_DIR}/${DATE}.log 2>&1
```

```
DUMP_SIZE=$(du -sh ${BACKUP_DIR}/${DATE}_${SUOP}.pgsql.backup |
awk '{print $1} ')
COMPLETED=$(ls -lt ${BACKUP_DIR}/ | awk 'NR==2{print $6,$7,$8} ')
cat > /pgbackup/db/log/dump.log<< EOF
```

**Название БД:**

`$SUOP`

**Размер резервной копии:**

`$DUMP_SIZE`

**Снятие завершено:**

`$COMPLETED"`

**Лог бэкапа:**

`${LOG_DIR}/${DATE}.log`

для обеспечения возможности запускать скрипт назначьте права доступа на файл (скрипт)

`/pgbackup/db/scripts/pgbackup_db.sh` следующей командой:

```
$ sudo chmod +x /pgbackup/db/scripts/pgbackup_db.sh
```

настройте запуск скрипта резервного копирования БД по расписанию, используя команду:

```
$ sudo crontab -u backup-database -e
```

вставьте строки:

```
0 22 * * * /pgbackup/db/scripts/pgbackup_db.sh
```

### 2.1.1. Инструкция по внеплановому резервному копированию БД

Выполнить команду для снятия бэкапа БД:

```
$ sudo -u backup-database /pgbackup/scripts/pgbackup_db.sh
```

### 2.1.2. Инструкция по восстановлению резервной копии БД

Необходимо на сервере БД создать БД, в которую будет происходить восстановление бекапа:

```
$ sudo -u postgres createdb -O user suop
```

Где

- suop– название базы данных;
- user — имя пользователя под которым будет происходить восстановление БД.

Восстановить бекап БД мониторинга, данное действие производится с VM резервного копирования:

```
$ pg_restore -U user -h ip_address -Fd -v -j3 -d dir_for_dump  
>>/pgbackup/restore.log 2>&1
```

Где

- user — имя пользователя под которым производится восстановление БД;
- ip\_address — ip-адрес сервера СУБД;
- dir\_for\_dump — директория резервной копии;
- /pgbackup/restore.log лог восстановления БД;

В случае проверки данного файла лога обнаружены ошибки вида:

- Failed;
- Error;
- Fatal;
- Warning.

Необходимо зафиксировать возникшую проблему, передав данный лог для обработки.

### **2.1.3. Логирование резервного копирования**

#### **2.1.3.1. Логирование резервного копирования БД**

Логирование резервного копирования БД мониторинга осуществляется на ВМ резервного копирования в файл /pgbackup/\${SUOP}/log/\${DATE}.log.

Где

- \${DATE} – дата снятия бекапа в формате ГОД-МЕСЯЦ-ДЕНЬ-ЧАС-МИНУТА;
- \${SUOP} — имя БД, для которой производилось резервное копирование.

В случае проверки данного файла лога обнаружены ошибки вида:

- Failed;
- Error;
- Fatal;
- Warning.

Необходимо зафиксировать возникшую проблему, передав данный лог для обработки.

#### **2.1.4. Порядок снятия резервного копирования**

Очередность резервного копирования:

Основная БД.

### **2.2. Порядок диагностики, локализации и устранения проблем**

#### **2.2.1. Диагностика, локализация и устранения проблем с доступностью серверов**

В случае диагностирования недоступности сервера ПО в течении более 5 минут необходимо зафиксировать возникшую проблему, после чего выполнить перезагрузку виртуальной машины.

В случае, если перезагрузка виртуальной машины невозможна, либо после перезагрузки диагностируется недоступность сервера, необходимо войти в консоль администрирования виртуализации и выяснить причины недоступности виртуальной машины.

Диагностика, локализация и устранения проблем дискового пространства

Могут быть диагностированы следующие проблемы дискового пространства ПО:

- закончилось свободное место на диске;



- превышение лимитов на открытие файлов;
- иные ошибки.

### **2.2.1.1. Закончилось свободное место на диске**

При обнаружении ошибки вида:

```
No space left on device
```

На сервере, у которого закончилось свободное место на диске, для диагностирования проблемных папок выполните команду:

```
df -h
```

Команда выдает перечень папок с указанием занимаемого места на диске в подобном виде:

```
Filesystem Size Used Avail Use% Mounted on  
/dev/mapper/vg_boportalem-lv_root 230G 219G 16M 100% /
```

Для решения проблемы необходимо зафиксировать возникшую проблему, после чего выбрать следующие варианты решения проблемы:

- очистка ненужных логов ПО, установленного на сервере;
- увеличить раздел, на котором закончилось место.

### **2.2.1.2. Превышение лимитов на открытие файлов**

При обнаружении проблемы вида:

```
Too many open files
```

Необходимо выполнить следующие действия:

- Зафиксировать возникшую проблему
- На сервере, на котором возникла данная проблема в `/etc/sysctl.conf` добавьте строку (значение параметра должно выбираться исходя из текущих настроек ПО):

```
fs.file-max = 65536
```

Выполните команду:

```
# sysctl -p
```

В файл /etc/security/limits.conf добавьте:

```
* soft nofile 32768
* hard nofile 32768
```

Значения должны быть ниже fs.file-max.

Выполните перезапуск Серверов, при работе которых возникала проблема.

## **2.2.2. Диагностика, локализация и устранения проблем загрузки процессора**

При длительной (более 1 часа) высокой загрузки процессора (более 95%) необходимо выполнить следующие действия:

- зафиксировать возникшую проблему;
- провести анализ перечня процессов на предмет выявления проблемного процесса;
- в зависимости от процесса, вызывающего критическую загрузку процессора выполнить либо завершение процесса, либо детальный анализ логов для выявления причин возникновения проблемы с последующим формированием предложений по устранению проблемы.

## **2.2.3. Диагностика, локализация и устранения проблем СУБД**

### **2.2.3.1. Автоматическая «очистка мусора» базы данных**

В ходе функционирования ПО в СУБД может накапливаться так называемый «мусор» – системные данные, оставшиеся от незавершенных по различным причинам операций, которые могут влиять на производительность ПО.

Для удаления этих данных («очистки мусора») необходимо еженедельно или при снижении производительности СУБД выполнять автоматическую «очистку мусора» встроенными средствами СУБД PostgreSQL.

Для этого необходимо на сервере баз данных выполнить следующую команду:

```
# vacuumdb -h host -p port -U user -d SUOP -f -v -z
```

- host - адрес/имя хоста;
- port - порт, на котором работает СУБД;
- user - имя пользователя;

– SUOP - имя БД.

Для автоматизации данного процесса возможно использование cron. Для настройки периодического выполнения команды необходимо:

Открыть на редактирование файл заданий:

```
cron $ crontab -e -u postgres
```

Вставить правило периодического выполнения в следующем формате (пример):

```
* * * * * vacuumdb -h host -p port -U user -d suop -f -v -z
```

Где

```
* * * * *
```

```
-----
```

```
|||||
```

```
|||||----- день недели (0 – 7) (Воскресенье =0 или =7);
```

```
|||----- месяц (1 – 12);
```

```
||----- день (1 – 31);
```

```
|----- час (0 – 23);
```

```
----- минута (0 – 59).
```

### 2.2.3.2. Мониторинг блокировок СУБД

В ходе функционирования ПО на СУБД могут возникать критические блокировки данных, существенно влияющие на производительность ПО.

В случае снижения производительности ПО, а также периодически (еженедельно) необходимо осуществлять мониторинг блокировок СУБД следующим образом:

Подключитесь к СУБД:

```
# psql -h host -p port -U user -d database
```

- host – указать адрес/имя хоста, к которому производится подключение;
- port – порт, на котором работает СУБД;
- user – имя пользователя;
- suop – имя БД, к которой производится подключение.

**Просмотр блокировок в СУБД в текущий момент:**

```
=# SELECT blocked_locks.pid AS
blocked_pid,blocked_activity.username AS
blocked_user,blocking_locks.pid AS
blocking_pid,blocking_activity.username AS
blocking_user,blocked_activity.query AS blocked_statement,
blocking_activity.query AS current_statement_in_blocking_process
FROM pg_catalog.pg_locks blocked_locks JOIN
pg_catalog.pg_stat_activity blocked_activity ON
blocked_activity.pid = blocked_locks.pid JOIN pg_catalog.pg_locks
blocking_locks ON blocking_locks.locktype =
blocked_locks.locktype AND blocking_locks.DATABASE IS NOT
DISTINCT FROM blocked_locks.DATABASE AND blocking_locks.relation
IS NOT DISTINCT FROM blocked_locks.relation AND
blocking_locks.page IS NOT DISTINCT FROM blocked_locks.page AND
blocking_locks.tuple IS NOT DISTINCT FROM blocked_locks.tuple AND
blocking_locks.virtualxid IS NOT DISTINCT FROM
blocked_locks.virtualxid AND blocking_locks.transactionid IS NOT
DISTINCT FROM blocked_locks.transactionid AND
blocking_locks.classid IS NOT DISTINCT FROM blocked_locks.classid
AND blocking_locks.objid IS NOT DISTINCT FROM blocked_locks.objid
AND blocking_locks.objsubid IS NOT DISTINCT FROM
blocked_locks.objsubid AND blocking_locks.pid !=
blocked_locks.pidJOIN pg_catalog.pg_stat_activity
blocking_activity ON blocking_activity.pid = blocking_locks.pid
WHERE NOT blocked_locks.GRANTED
```

**В случае обнаружения критических блокировок необходимо обратиться к разработчикам**

**ПО.**

## **2.2.4. Процедура проверки функционирования ПО после восстановления резервной копии и перевода ПО в штатный режим функционирования**

Для проверки функционирования ПО необходимо:

- Открыть главную страницу ПО;
- Осуществить вход в личный кабинет пользователя;
- При наличии доступа, перейти в административную консоль.

## **2.3. Оптимизация и настройки производительности**

### **2.3.1. Оптимизация производительности СУБД**

В ходе эксплуатации СУБД PostgreSQL с ростом объемов данных, а также в результате интенсивных операций удаления и обновления данных – может наблюдаться снижение производительности, замедление выполнения запросов, рост использования процессора и повышенная утилизация подсистемы дискового ввода-вывода. Универсального решения, по повышению производительности не существует, однако есть набор методов, сочетание которых позволит избежать негативных эффектов:

- увеличение объема оперативной памяти под разделяемый буфер (параметр конфигурации `shared_buffers`). Рекомендуемый объем – 40% от объема оперативной памяти, но не более 8 GB. Приводит к снижению интенсивности дискового ввода-вывода;
- выставить значение `effective_cache_size` в 50% от объема оперативной памяти. Это позволит оптимизатору запросов более точно определить объем файлового кэша и выбрать оптимальный план выполнения запроса;
- увеличение точности собираемой статистической информации. Увеличьте параметр `default_statistics_target` до значения 100 – 1000. Это должно положительно сказаться на построении оптимальных планов выполнения запросов;
- в случае если наблюдается снижение производительности запросов, производящих объемные выборки данных, сортировки и группировку данных, возможно постепенное увеличение параметра `work_mem`, с постоянным контролем над поведением СУБД. Более точное значение позволит определить анализ плана выполнения запроса, до и после изменения параметра;

- произвести операцию очистки старых версий записей.

#### **2.4. Порядок вывода ПО из эксплуатации**

После окончания срока эксплуатации ПО (5 лет с момента ввода в эксплуатацию, если иное не определено дополнительно) выполните следующие процедуры:

- сформируйте копию базы данных ПО;
- загрузите сформированные копии баз данных ПО, документации ПО на носители информации для хранения;
- передайте сформированный согласно п. 2 пакет должностному лицу, ответственному за дальнейшее хранение информации. Должностное лицо, ответственное за дальнейшее хранение информации должно определяться Приказом по организации, являющейся оператором ПО. Требования к техническим средствам.

### 3. ТРЕБОВАНИЯ К АППАРАТНЫМ СРЕДСТВАМ

К аппаратному и программному обеспечению, которые используются для функционирования ПО «Система управления облачной платформой «ТИОНИКС», предъявляются требования, изложенные в таблице 3.

Таблица 3 – Минимальные требования к программному и аппаратному обеспечению

Операционная система	Astra Linux Special Edition CentOS 7.8 Red Hat Enterprise Linux 7.8 Ubuntu 16.04.5 - 16.04.6 Debian 9.11
Процессор Оперативная память	В соответствии с требованиями ОС, установленной в среде виртуализации
Жесткий диск (минимальное свободное пространство)	1 ГБ
Сетевой адаптер	Ethernet 1 Гбит/сек
Браузер	Яндекс.Браузер 20.7.2 Спутник 4.1 Chromium-gost 84 Google Chrome 84 Mozilla Firefox 79 Microsoft Edge 83 Chromium 81 Safari 13.0
Монитор	Диагональ не менее 17", разрешение не менее 1280x1024 (4:3) или 1440x900 (16:9);
Видеоадаптер	Без ограничений
Периферийное оборудование	Клавиатура, манипулятор типа мышь

#### **4. ОПИСАНИЕ ФУНКЦИЙ И СТРУКТУРА**

Структура и описание функций программных средств ПО и структура технических средств ПО приводится в документе «ПО «Система управления облачной платформой «ТИОНИКС». Руководство администратора. RU.НРФЛ.00005-01.95.01».



**ПЕРЕЧЕНЬ СОКРАЩЕНИЙ**

<b>Сокращение</b>	<b>Расшифровка</b>
БД	База данных
ВМ	Виртуальная машина
ОС	Операционная система
ПО	Программное обеспечение «Система управления облачной платформой «ТИОНИКС»
СУБД	Система управления базами данных
ЦОД	Дата-центр, или центр обработки данных — это специализированное здание для размещения серверного и сетевого оборудования и подключения абонентов к каналам сети Интернет

