СДИ «Базис» 14.0

Руководство по установке

Содержание

1. Примечания по процедуре установки

В руководстве описывается начальная установка системы «СДИ Базис» версии 14.0 и выше. Для миграции данных из существующей Oracle-версии Базис необходимо обратиться к документу «Руководство по миграции данных из «СДИ Базис» версии 13.5+ (Oracle) в «СДИ Базис» версии 14 (Postgres)».

2. Содержимое дистрибутива поставки

Дистрибутив прикладного решения **СДИ Базис 14.0** поставляется в виде **ZIP** архивов с компонентами сервера приложений и базы данных Базис.

Архив с компонентами стандартного приложения Базис:

basis-standard-[VERSION].zip

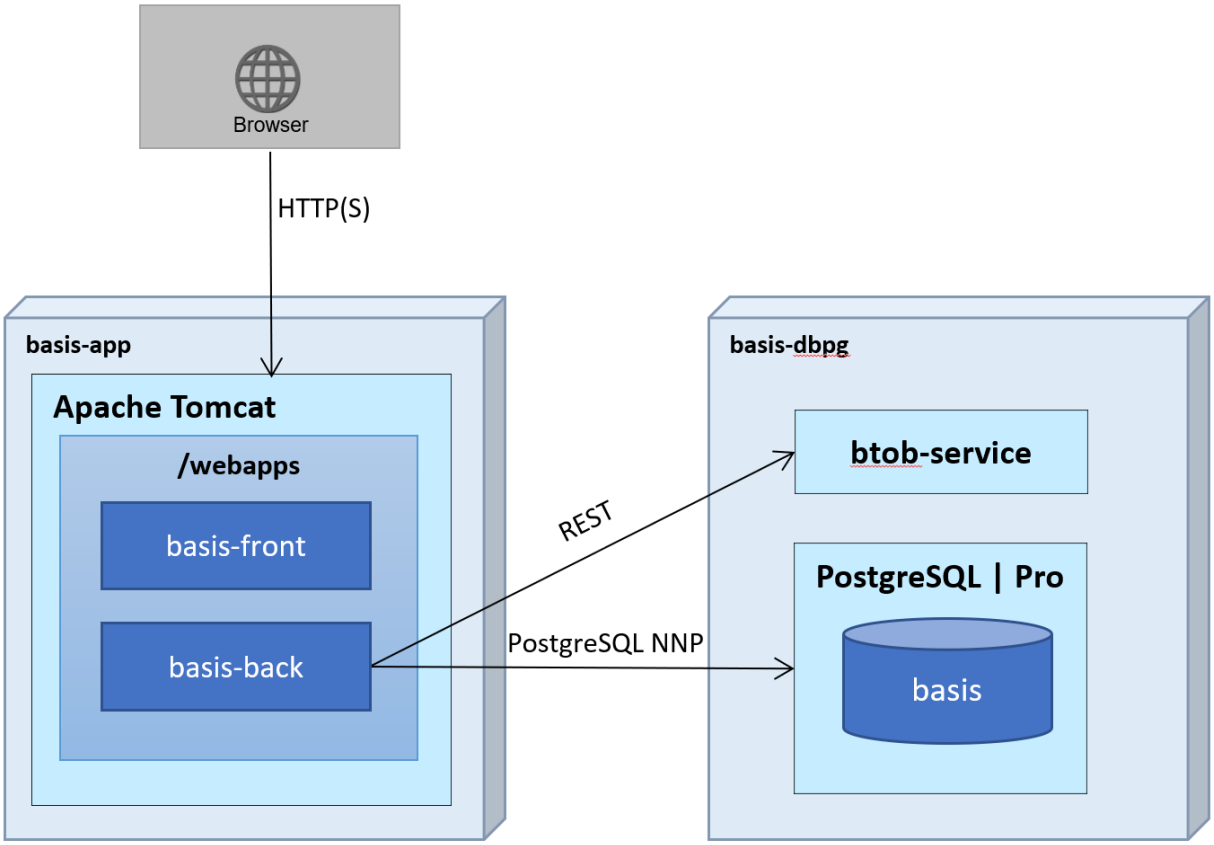
Архив с компонентами БД Базис:

basis-standard-dbpg-[VERSION].zip

Состав архива **basis-standard-dbpg**:

Имя папки	Описание
pg-backup	Содержит начальный дамп БД Базис PostgreSQL, а также утилиты для бэкапа и восстановления БД PostgreSQL.
pg-stuff	Содержит набор утилит и конфигурационных файлов для настройки сервисов БД Базис, а также для миграции данных из БД Oracle

3. Основные компоненты и топология контура



3.1. Перечень компонентов

Компонент	Назначение
basis-app	Физический узел или логический раздел, выделенный для хостинга сервера приложений Tomcat СДИ Базис.
basis-dbpg	Физический узел или логический раздел, выделенный для хостинга инстанции СУБД PostgreSQL.
btob-service	Сервис динамической трансляции и кэширования данных
PostgreSQL или PostgresPro	База данных Базис СУБД PostgreSQL (PostgresPro). Используется для хранения элементов конфигурации и объектов учета.

4. Установка PostgreSQL на Linux

В данном разделе изложена пошаговая процедура установки **PostgreSQL 16** с необходимыми расширениями на ОС **CentOS** 7 или 8. Раздел описывает типовой набор административных конфигураций, необходимых для установки СУБД PostgreSQL на чистую операционную систему, и предназначен в качестве подсказки Администратору Linux, не претендуя на полноту и покрытие иных конфигураций Linux, требуемых в том или ином промышленном окружении.

Процедура установка на другие ОС семейства RedHat отличается незначительно. Postgres Pro может быть установлен только на enterprise версии Linux, например **RHEL** 8 или 9 (полный список поддерживаемых систем приведен в документе «СДИ «Базис» 14.0. Системные требования»). Существенные отличия в установке и конфигурировании СУБД **Postgres Pro** далее по тексту выделены в примечания.

За информацией об установке на другие версии ОС, а также за различной справочной информацией следует обращаться к документации на PostgreSQL или Postgres Pro.

4.1. Подготовка к установке

Используются следующие компоненты и версии продуктов:

- CentOS 8
- PostgreSQL 16.0
- Orafce 4.6
- pg_variables 1.2.1
- pg_dbms_job 1.5

В распоряжении администратора на его рабочей станции минимально должны иметься следующие инструменты, необходимые для выполнения процедуры инсталляции:

- **SSH/SFTP клиент:** для удаленного подключения к консоли и файловой системе Linux-сервера.

4.2. Роли и полномочия администраторов развертывания

Роль	Описание полномочий
Linux Admin	Данная роль должна иметь полный административный доступ к узлам **basis-app**, **basis-db**, позволяющий выполнить развертывание и первоначальную настройку OS на данных узлах. В процессе конфигурации создает и предоставляет необходимые полномочия для **Tomcat Admin**.
Tomcat Admin	Данная роль имеет полномочия на узле **basis-app**, позволяющие выполнять развертывание сервера приложений **Apache Tomcat**. Необходимые полномочия на узлах получает от **Linux Admin**. Также получает необходимые учетные данные для настройки подключения к Postgres Database от **Postgres DBA**.
Postgres DBA	Данная роль должна иметь административные полномочия на развертывание PostgreSQL на узле **basis-db**. По завершении конфигурации роль обязана предоставить необходимые учетные данные для подключения к созданной БД command для роли **Tomcat Admin**.

4.3. Развертывание и настройка ОС Linux

Роль: Linux Admin

В данной инструкции не описывается процедура базовой установки **64-битной** версии операционной системы **CentOS** на целевые узлы **basis-app, basis-dbpg**. За соответствующей информацией следует обращаться к документации на операционную систему.

Для старта как минимум потребуется установка базового набора пакетов **«Infrastructure Server»**.

4.3.1. Установка дополнительных пакетов ОС Linux

Для сборки из исходных кодов PostgreSQL, установки и корректной работы всех компонентов платформы требуется установить дополнительные системные пакеты **CentOS** (**gcc**, **automake**, **python** и др). Для установки требуемых пакетов следует воспользоваться утилитой **yum** из командной строки **root**:

```
$ yum groupinstall "Development Tools" -y

$ yum install libtool-ltdl-devel libicu-devel bzip2-devel boost python3-devel systemd-devel zlib zlib-devel readline-devel libxml2 libxml2-devel -y

$ yum install java-11-openjdk-devel -y
```

Выберите директорию для java 11 по умолчанию с помощью утилиты **update-alternatives**:

```
$ update-alternatives --config java
```

Пакеты необходимо установить со всеми зависимостями, которые автоматически определяет **yum**.

4.3.2. Проверка настроек Firewall

Необходимо проконтролировать настройки firewall, не заблокирован ли доступ извне к портам на узлах **basis-app, basis-dbpg**. А также нет ли каких-либо ограничений при взаимодействии между внутренними узлами контура, если используется распределенная установка.

Порт	Сервис
TCP *:111	rpc
TCP *:22	ssh
TCP *:5432	postgres
TCP *:9090	btob-service

Обычно доступ к узлам контура извне контролируется сетевыми администраторами на отдельном аппаратном firewall. На самом узле сервис **firewalld** рекомендуется отключить, чтобы не блокировать взаимодействие между компонентами **basis-app, basis-dbpg** в случае распределенной установки компонентов среды:

```
# systemctl stop firewalld
# systemctl disable firewalld
```

4.3.3. Создание пользователей OS Linux

Установка и запуск СУБД Postgres выполняется от непривилегированного пользователя OS. Требуется создать такого пользователя с именем **postgres** и первичной группой **postgres**:

```
# useradd -U -m postgres
# passwd postgres
```

4.3.4. Предоставление полномочий root-пользователя

Так как процедуру настройки и запуск/остановку прикладных приложений будет выполнять роль **Postgres DBA**, используя непривилегированного пользователя **postgres**, необходимо дать этому пользователю права на запуск скриптов управления от пользователя **root** при помощи утилиты **sudo**. Для этого нужно добавить в файл **/etc/sudoers** следующую строку:

```
postgres ALL=(root) NOPASSWD: ALL
```

4.3.5. Настройка лимитов на ресурсы

Для созданного пользователя требуется изменить лимиты на ресурсы:

- Количество процессов (**nproc**) = 4096
- Максимальный размер файла (**fsize**) = unlimited
- Максимальное количество дескрипторов (**nofile**) = 65536

Для этого нужно создать новый файл **/etc/security/limits.d/90-postgres.conf** со следующим содержанием:

```
postgres        soft    nproc     4096
postgres        hard    nproc     4096
postgres        soft    fsize     unlimited
postgres        hard    fsize     unlimited
postgres        soft    nofile    65536
postgres        hard    nofile    65536
```

4.3.6. Настройка локального resolver

При развертывании среды СДИ «Базис» рекомендуется на каждом из узлов среды в локальный hosts-файл ресолвера прописать well-known синонимы узлов с конкретными для данной среды IP-адресами. И затем всегда использовать эти синонимы при настройке

прикладных компонентов и сервисов СДИ «Базис». Данный подход позволит в последующем, без изменения сложной конфигурации многочисленных прикладных компонентов, прозрачно клонировать или переносить среду на другое железо или в другое окружение, изменяя только IP-адреса в hosts-файле на актуальные.

Отредактируйте файл **/etc/hosts** и добавьте туда следующие записи:

```
<ip-адрес узла basis-app> basis-app
<ip-адрес узла basis-db>  basis-dbpg
```

4.4. Установка СУБД PostgreSQL

Роль: Postgres DBA

В разделе описана пошаговая процедура установки стандартной версии PostgreSQL из исходного кода. Все действия выполняются на узле **basis-dbpg** от непривилегированного пользователя **postgres**.

ПРИМЕЧАНИЕ

Версия Postgres Pro может быть установлена только из официального бинарного дистрибутива по инструкции.

Скачайте архив postgresql-16.x.tar.bz2 (например, postgresql-16.0.tar.bz2) с исходным кодом СУБД PostgreSQL с официального сайта в домашнюю директорию /home/postgres и распакуйте его:

```
$ tar xf postgresql-16.0.tar.bz2
$ cd postgresql-16.0
```

Подготовьте конфигурацию с необходимыми опциями:

```
$ ./configure --with-icu --with-python --enable-nls='de ru' --with-systemd
--with-libxml
```

Скомпилируйте и установите PostgreSQL:

```
$ make all
$ sudo make install
```

Назначьте пользователя **postgres** владельцем каталога **/usr/local/pgsql**:

```
$ sudo chown -R postgres /usr/local/pgsql
```

Добавьте в переменную окружения PATH пользователя postgres путь к каталогу /usr/local/pgsql/bin:

```
$ export PATH=$PATH:/usr/local/pgsql/bin
```

// Установка PostgreSQL на Linux

Инициализируйте БД:

```
$ initdb -D /usr/local/pgsql/data
```

Создайте следующий файл описания службы **/etc/systemd/system/postgresql.service**:

```
[Unit]
Description=PostgreSQL database server
Documentation=man:postgres(1)
After=syslog.target
After=network.target

[Service]
Type=notify

User=postgres
Group=postgres

# Location of database directory
Environment=PGDATA=/usr/local/pgsql/data

# Disable OOM kill
OOMScoreAdjust=-1000
Environment=PG_OOM_ADJUST_FILE=/proc/self/oom_score_adj
Environment=PG_OOM_ADJUST_VALUE=0

ExecStart=/usr/local/pgsql/bin/postgres -D ${PGDATA}
ExecReload=/bin/kill -HUP $MAINPID
KillMode=mixed
KillSignal=SIGINT

# Do not set any timeout value, so that systemd will not kill postgres
# during crash recovery.
TimeoutSec=0

# 0 is the same as infinity, but "infinity" needs systemd 229
TimeoutStartSec=0

TimeoutStopSec=1h

[Install]
WantedBy=multi-user.target
```

Установите сервис **PostgreSQL** в автозапуск:

```
$ sudo systemctl daemon-reload
$ sudo systemctl enable postgresql.service
```

Стартуйте сервис **PostgreSQL**:

```
$ sudo systemctl start postgresql
```

Проверьте соединение:

```
$ psql
     (16.0)
```

Для выхода из CLI postgresql наберите \q.

Создайте пароль для админа **postgres**:

```
$ psql -c "alter user postgres with password <PASSWORD>"
```

ВНИМАНИЕ

Далее в тексте документа, в примерах конфигурационных файлов (command_config.properties, application.properties), а также в скрипте создания роли basis (recreate_pgdb.sql) в качестве значения пароля пользователя basis используется текст **password**. Во время установки необходимо заменить этот **password** на настоящий пароль.

Также в тексте документа при создании других аккаунтов с паролем используется лексема **<PASSWORD>**, которую также нужно заменить на настоящий пароль соответствующего аккаунта.

Если firewall включен, добавьте разрешение для сервиса **postgresql**:

```
$ sudo firewall-cmd --add-service=postgresql --permanent
$ sudo firewall-cmd -reload
```

4.4.1. Разрешение удаленного доступа

Для разрешения доступа к сервису Postgres с удаленных узлов добавьте в файл **/usr/local/pgsql/data/postgresql.conf** следующую запись:

```
 listen_addresses = '*'
```

Добавьте в конфигурацию **/usr/local/pgsql/data/pg_hba.conf** разрешение принимать соединения с удаленных узлов подсети:

```
# ACCEPT FROM TRUSTED SUBNET
host all all <Подсеть СДИ Базис>/24 md5
```

Перезапустите сервис **postgresql**:

```
$ sudo systemctl restart postgresql
```

4.4.2. Установка расширений PostgreSQL

Установите расширение **orafce**:

```
$ cd ~/postgresql-16.0/contrib
$ git clone https://github.com/orafce/orafce.git
$ cd orafce
$ make USE_PGXS=1
$ make install
$ make installcheck
```

Установите расширение **pg_variables**:

```
cd ~/postgresql-16.0/contrib
$ git clone https://github.com/postgrespro/pg_variables.git
$ cd pg_variables
$ make USE_PGXS=1
$ sudo make install
$ make installcheck
```

Установите расширение **pg_dbms_job**:

```
cd ~/postgresql-16.0/contrib
$ git clone https://github.com/MigOpsRepos/pg_dbms_job.git
$ cd pg_dbms_job
$ sudo make install
$ make installcheck
```

4.5. Конфигурирование PostgreSQL

Основные настройки PostgreSQL находятся в двух файлах: **postgresql.conf** и **pg_hba.conf**. В первом хранятся настройки самой базы данных, а во втором – настройки доступа к ней.

К обязательным настройкам Базис относятся следующие из **/usr/local/pgsql/data/postgresql.conf**:

```
listen_addresses = '*' # или указать host basis-app
shared_buffers = 128MB
search_path = 'public,btob,oracle,pg_catalog,basis'
timezone = 'Europe/Moscow'
log_timezone = 'Europe/Moscow'
include 'basis.global.conf'
```

Остальные настройки остаются на усмотрение администратора PostgreSQL и могут отличаться в различных версиях БД.

Пример настроек содержится в дистрибутиве в каталоге **pg-stuff/config** и может быть использован в качестве начальной конфигурации. Для этого скопируйте конфигурационный файлы с предустановленными настройками из дистрибутива в каталог

/usr/local/pgsql/data

```
$ cp ~/basis-distr/pg-stuff/config/postgresql.conf /usr/local/pgsql/data/
$ cp ~/basis-distr/pg-stuff/config/pg_hba.conf /usr/local/pgsql/data/
$ cp ~/basis-distr/pg-stuff/config/basis.global.conf /usr/local/pgsql/data/
```

Перезапустите сервис **PostgreSQL**:

```
$ systemctl restart postgresql
```

5. Создание прикладной базы данных Базис

Роль: Postgres DBA

Все действия выполняются на узле **basis-dbpg** от непривилегированного пользователя **postgres**.

Если миграция данных не требуется, достаточно установить БД Command из начального дампа дистрибутива по п 7.1. Если требуется начальная миграция, необходимо следовать инструкциям в документе «СДИ «Базис» 14.0. Руководство по миграции».

5.1. Подготовка каталогов

Перед началом установки необходимо распаковать содержимое архива:

basis-standard-dbpg на узле СУБД PostgreSQL **basis-dbpg** в домашний каталог непривилегированного пользователя **postgres ~/basis-distr**

```
$ cd ~/
$ unzip -o basis-standard-dbpg-[VERSION].zip
```

После распаковки дистрибутива рекомендуется скопировать директории бэкапа и миграции в домашнюю директорию **postgres**:

```
$ cp -r ~/basis-distr/pg-backup ~/
$ cp -r ~/basis-distr/pg-stuff ~/
```

5.2. Установка из дампа данных дистрибутива

Создать необходимые роли и расширения в БД Postgres:

```
$ cd ~/pg-stuff
$ psql -U postgres -f ./recreate_pgdb.sql
```

Восстановить начальный дамп из дистрибутива:

```
$ chmod a+x ~/pg-backup/*.sh
$ ~/pg-backup/pg_restore_latest.sh
```

Обновить прикладные объекты в БД command PostgreSQL:

```
$ psql -d command -U basis -f ./upgrade_pgdb.sql
```

Пересоздать роль **cmd_meta** для генератора отчетности:

```
$ psql -U postgres -f ./recreate_meta.sql
```

5.2.1. Завершающие процедуры

В Postgres необходимо создать роль с именем пользователя, под которым выполняется приложение Tomcat (Базис). Например **tomcat**:

```
$ psql
psql> \c command
psql> CREATE ROLE "tomcat" WITH SUPERUSER CREATEDB CREATEROLE
INHERIT LOGIN REPLICATION BYPASSRLS CONNECTION LIMIT -1;
```

Возможность подключения к Базис по https контролируется в конфигурационной таблице STFCFG_SETTING по ключу 'SYS_HTTP_SECURITY' одним из двух значений: **SSL_only** или **request_dependent**.

Подключение только по HTTPS:

```
psql> update STFCFG_SETTING set VALUE = 'SSL_only' where NAME =
'SYS_HTTP_SECURITY' and USER_ELID = 'DEFAULT___USER' and MAN_ID = 0;
```

Подключение по HTTP(S):

```
update STFCFG_SETTING set VALUE = 'request_dependent' where NAME =
'SYS_HTTP_SECURITY' and USER_ELID = 'DEFAULT___USER' and MAN_ID = 0;
```

5.3. Установка сервиса динамической трансляции

Скопировать из дистрибутива каталог **pg-stuff/btob-service** в **/var**:

```
$ sudo cp -r ~/basis-distr/pg-stuff/btob-service /var/
```

Скопировать из дистрибутива начальную конфигурацию **btob-service**:

```
$ sudo cp ~/basis-distr/pg-stuff/config/RuleSet.json /var/
```

Назначить права на каталог **btob-service**:

```
$ sudo chown -R postgres /var/btob-service
$ chmod a+x /var/btob-service/btob-service-1.0-SNAPSHOT.jar
```

Установить порт сервиса и параметры соединения с БД command в конфигурационном файле **/var/btob-service/application.properties**:

// Создание прикладной базы данных Базис

```
server.port=9090
spring.datasource.url=jdbc:postgresql://basis-dbpg:5432/command
spring.datasource.username=basis
spring.datasource.password=password
```

Создать следующий файл описания службы **/etc/systemd/system/btob-service.service**:

```
[Unit]
Description=Manage btob service

[Service]
WorkingDirectory=/var/btob-service
ExecStart=/bin/java -Xms256m -Xmx512m -jar btob-service-1.0-
SNAPSHOT.jar
User=postgres
Type=simple
Restart=on-failure
RestartSec=10
StandardOutput=syslog
StandardError=syslog
SyslogIdentifier=btob-service

[Install]
WantedBy=multi-user.target
```

Создать следующий файл конфигурации **/etc/rsyslog.d/btob-service.conf**:

```
if $programname == 'btob-service' then /var/log/btob-service.log
& stop
```

Обновить конфигурацию **systemd**, рестартовать **rsyslog**, установить **btob-service** в автозапуск и стартовать **btob-service**:

```
systemctl daemon-reload

systemctl restart rsyslog

systemctl enable btob-service
systemctl start btob-service
```

6. Опционально: базовые задачи администрирования сервисов Postgres

В разделе дана краткая справка по типовым задачам администрирования сервисов Базис Postgres: логирование, резервирование и восстановление для случая простой стандартной установки.

6.1. Компоненты администрирования

В таблице ниже приведены пути к основным компонентам установленных сервисов Базис / PostgreSQL.

	postgresql 15/16	postgres pro 15	btob-service
service	`systemctl start|stop postgresql`	`systemctl start|stop postgrespro-std-15`	`systemctl start|stop btob-service`
home	`/usr/local/pgsql/`	`/opt/pgpro/std-15/`	`/var/btob-service`
config	`/usr/local/pgsql/data/pg_hba.conf` `postgresql.conf`	`/opt/pgpro/std-15/data/pg_hba.conf` `postgresql.conf`	`/var/btob-service/application.properties`
log	`/usr/local/pgsql/data/log/postgresql.log`	`/opt/pgpro/std-15/data/log/postgresql.log`	`/var/log/btob-service.log`
backup	`/home/postgresql/pg-backup/pg_backup.config`	`/home/postgresql/pg-backup/pg_backup.config`	

6.2. Резервирование и восстановление БД Базис

Выполняется на узле **basis-dbpg** под непривилегированным пользователем **postgres**

Сделать бэкап всех локальных БД Postgres:

```
$ ~/pg-backup/pg_backup_rotated.sh
```

Скрипт сделает бэкап ВСЕХ баз Postgres на **localhost** и поместит его копии в 2 каталога:

- ~/pg-backup/db-all/YYYY-MM-DD-daily (или weekly) – ежедневные бэкапы перезаписываются каждые 7 дней, еженедельные – каждый месяц.
- ~/pg_backup/db-all/latest – последний сделанный бэкап всех БД, перезапишет предыдущий

Восстановить все базы данных, бэкапы которых находятся в каталоге **latest**:

```
$ /home/postgres/ora2pg/pg_backup/pg_restore_latest.sh
```

6.3. Настройка Logrotate сервиса динамической трансляции

В Linux большинство сервисов и программ, которые работают в фоне, таких как Apache, Nginx, Postfix и др. записывают информацию о своем состоянии, результатах работы и ошибках в лог-файлы. Стандартное расположение логов или как их еще называют – журналов – в папке **/var/log**.

Logrotate – это популярная утилита, поэтому в большинстве дистрибутивов она поставляется по умолчанию. В случае ее отсутствия для установки в **CentOS** выполните команду:

```
$ sudo yum install logrotate
```

Все основные настройки программы находятся в файле **/etc/logrotate.conf**; дополнительные настройки, касающиеся правил и других возможностей, могут быть размещены в папке **/etc/logroate.d/**. Чтобы конфигурационные файлы из этой папки загружались программой, необходимо добавить в основной конфигурационный файл такую строчку:

```
include /etc/logrotate.d
```

Каждый лог, который подлежит ротации, описывается таким образом:

```
адрес_файла_лога {
 директивы
}
```

Необходимо создать в папке **/etc/logroate.d/** файл **btob-service-log.conf** и дополнить его следующим кодом:

```
/var/log/btob_service.log {
 daily
 rotate 1
 size 10M
 compress
 delaycompress
}
```

Эти настройки означают, что ротация журналов будет выполняться ежедневно, будет храниться один последний журнал, более старые копии будут автоматически удаляться. Минимальный размер для ротации – 10 мегабайт; ротация не будет выполнена, если лог не превышает 10 мегабайт.

Проверьте расписание cron запуска утилиты:

```
$ sudo ls /etc/cron.daily/
```

Или запустите ротацию принудительно:

```
$ sudo logrotate -f /etc/logrotate.d/rsyslog.conf
```

7. Установка приложения Базис

Роль: Tomcat Admin

В стандартном варианте в качестве веб-сервера используется Apache Tomcat, который соединен с сервером базы данных Postgres для хранения данных Базис. При этом оба сервера могут работать как на одном общем оборудовании, так и на двух отдельных единицах оборудования.

Данная инструкция не покрывает процедуру установки Apache Tomcat. Для этого следует обратиться к оригинальной документации Apache Tomcat.

Ниже в инструкции предполагается, что к каталогу необходимой инсталляции Tomcat имеется псевдоним /app/tomcat/latest. Фактический каталог может быть иной, в этом случае нужно либо использовать его значение в конфигурационных файлах вместо псевдонима /app/tomcat/latest, либо создать соответствующую символьную ссылку:

```
$ ln -s /app/apache-tomcat-9.0.85 /app/tomcat/latest
```

Далее в документе:

$TOMCAT_HOME=/app/tomcat/latest

Все действия выполняются на узле **basis-app** от непривилегированного пользователя **tomcat**. Приложение Apache Tomcat должно выполняться от непривилегированного пользователя **tomcat**.

1. Скопируйте каталог **~/basis-distr/command** в директорию **$TOMCAT_HOME/webapps**.
2. Распакуйте архив basis-standard-wmicons.zip в директорию **$TOMCAT_HOME/wmicons**
3. Распакуйте архив basis-standard-online_help.zip в директорию **$TOMCAT_HOME/webapps/doc**

7.1. Настройка сервера приложений

Для корректной работы с файлами рекомендуется использовать универсальную кодировку UTF-8. Для этого необходимо добавить в файле **$TOMCAT_HOME\bin\catalina.sh** в переменную **JAVA_OPTS** выражение “**-Dfile.encoding=UTF-8**”.

```
JAVA_OPTS="-Xms512M -Xmx2048M -XX:MaxMetaspaceSize=512m -
Djava.awt.headless=true -Dfile.encoding=UTF-8"
```

Добавьте в файл **$TOMCAT_HOME\conf\server.xml** в раздел ***<Host>*** следующие атрибуты и строки:

```
<Host name="localhost"  appBase="webapps" unpackWARs="true"
autoDeploy="false" deployOnStartup="false">

   <Context path="" crossContext="true"
docBase="/app/tomcat/latest/webapps/command" debug="0"/>

   <Context path="/wmicons" crossContext="true"
docBase="/app/tomcat/latest/webapps/wmicons" debug="0"/>

   <Context path="/doc" crossContext="true"
docBase="/app/tomcat/latest/webapps/doc" debug="0"/>
…
</Host>
```

7.1.1. Настройка HTTPS

Протокол HTTPS используется для шифрования потока данных между клиентом (браузером) и веб-сервером (Tomcat). HTTPS позволяет шифровать поток данных, предоставляя тем самым надежную защиту от прослушивания информации. Если у клиента и веб-сервера отсутствует сертификат для посещаемого сайта, браузеры классифицируют такой сайт как ненадежный и предупреждают пользователя о возможной опасности, исходящей от такого сайта. Для предотвращения подобной опасности используется сертификат. Сертификат предоставляется органом сертификации и подтверждает подлинность веб-сайта или его владельца. Это означает, что пользователь может быть уверен, что лицо или фирма действительно существует и что в соединении между клиентом и сервером отсутствуют промежуточные узлы. В Tomcat такой сертификат запакован в файле Keystore и может быть настроен соответствующим образом.

Настройка файла Keystore описывается в руководстве по установке Tomcat. Файл Keystore – это файл, создаваемый Java и содержащий сертификат для сертификации SSL. За создание и сертификацию сертификата SSL, а также за создание файла Keystore отвечает заказчик.

По умолчанию при установке Tomcat для порта HTTPS Connector параметр Keystore не устанавливается. В зависимости от используемого источника установки Connector можно закомментировать или раскомментировать. В нашем примере мы исходим из того, что он закомментирован.

Порт HTTPS Connector из файла server.xml перед настройкой:

```
<!-- Define an SSL/TLS HTTP/1.1 Connector on port 8443
    ***************************
-->
<!--
<Connector port="8443" maxThreads="150" SSLEnabled="true" />
-->
```

Для настройки необходимо удалить знаки комментариев и добавить запись **keystoreFile**. Для защиты Keystore паролем необходимо добавить запись **keystorePass**. Запись **keystoreType** является дополнительной функцией, которая требуется, только если KeyStore не является файлом KeyStore (например, pkcs12).

Порт HTTPS Connector из файла server.xml после настройкой (на примере **Apache Tomcat 8.5**):

```
<!-- Define an SSL/TLS HTTP/1.1 Connector on port 8443
    ***************************
-->
<Connector port="8443" maxThreads="150" SSLEnabled="true"
  keystoreType="PKCS12"
  keystoreFile="/path/to/.keystore/file"
  keystorePass="xxxxxxx"/>
```

Поскольку Базис блокирует незащищенные HTTP запросы, HTTP Connector может быть удален:

```
<Connector port="8080" protocol="HTTP/1.1"
               connectionTimeout="20000"
               redirectPort="8443" />
```

В любом случае, https redirect должен быть запрещен:

```
<Connector port="8080" protocol="HTTP/1.1"
               connectionTimeout="20000"
               redirectPort="8443" <- необходимо удалить />
```

По соображениям безопасности, Базис всегда работает с параметром **SSL_only**. Следующая команда SQL позволяет задать / изменить в базе данных настройку в схеме **command** посредством **psql**:

```
psql> update STFCFG_SETTING set VALUE = 'SSL_only' where NAME =
'SYS_HTTP_SECURITY' and USER_ELID = 'DEFAULT___USER' and MAN_ID = 0;
```

7.2. Конфигурация параметров

Для определения параметров конфигурации системы Базис используется файл у **$TOMCAT_HOME\conf\command_config.properties**.

Укажите порт, установленный в конфигурации сервера приложений:

```
command.config.specific.apachePortSSL=<PORT>
```

Задайте при необходимости постоянный протокол передачи данных. Например, это может понадобиться, если прокси изменяет протокол HTTP на HTTPS:

```
command.config.specific.constantProtocol=https
```

Присвойте экземпляру уникальный идентификатор (APP_SERVER_ID). Данный идентификатор не должен использоваться ни для одного из других экземпляров. Длина ограничена 14-ю знаками.

```
command.config.specific.applicationServerId=<APP_SERVER_ID>
```

Укажите информацию об используемой базе данных PostgreSQL. Установите значения для адреса сервера [url], SID [instance] и ListenerPort [port] (по умолчанию – 1521, возможны также другие значения), пользователя [username] и пароля [password].

```
command.config.dbserver.default.type=4
command.config.dbserver.default.url=basis-db-pg
command.config.dbserver.default.instance=command
command.config.dbserver.default.password=<PASSWORD>
command.config.dbserver.default.username=<USERNAME>
```

Установите URL для подключения к **btob-service**:

```
command.config.dbserver.default.btobServiceUrl=http://
<DATABASE_SERVER_HOST>:9090
```

7.2.1. Опционально: конфигурирование контекста

Поскольку приложение Базис должно работать в своем собственном контексте, атрибут **path** элементов **<Context>** в файле **<$TOMCAT_HOME>/conf/server.xml** необходимо дополнить пользовательским контекстом. Например, для контекста **/app/basis** запись должна выглядеть следующим образом:

```
<Context path="/app/basis" docBase="basis" debug="0"/>
```

Также необходимо изменить конфигурацию приложения Базис в соответствии с указаниями раздела 3.5.2.

ПРИМЕЧАНИЕ

Контекст – это часть URL-лицензии для экземпляра Базис. При изменении контекста необходимо также загрузить новую URL-лицензию. Начиная с версии 13, требуется только одна запись контекста, записи "...axis" необходимо удалить.

7.2.2. Опционально: включение сжатия

Последовательность действий

Сервер Tomcat может сжимать передаваемые клиенту данные, в результате чего передаваемый объем данных уменьшается и требуется меньшая пропускная способность канала передачи данных. Сжатие рекомендуется использовать при низкой скорости передачи данных или небольшой ширине полосы пропускания между клиентом и сервером.

Для включения сжатия в элементе **<Connector>** в файле **<$TOMCAT_HOME>/conf/server.xml** необходимо установить следующие дополнительные атрибуты, определяющие порт http:

```
compression="on"
compressionMinSize="2048"
```

Если используемый элемент **connector** поддерживает функцию **sendfile**, ее необходимо отключить, поскольку в противном случае сжатие будет невозможным:

```
useSendfile=”false”
```

Пример элемента <**Connector**> целиком:

```
<Connector port="443" maxThreads="150" minSpareThreads="25" debug="0"
connectionTimeout="20000" compression="on" compressionMinSize="2048"/>
```

7.2.3. Опционально: конфигурирование Apache для статического контента

Если для обработки статических файлов используется Apache, возможно потребуется изменение конфигурации, например, для активирования глубинных ссылок Deep Link. В этом нет необходимости, если Apache только пересылает запросы на сервер приложений (напр., выполняет функцию прокси или балансировки нагрузки).

Обычно, HtmlFrontendServlet в сервере приложений принимает все запросы к **/html/**** и в отсутствие запрашиваемого файла пересылает их на **index.html**. Фронтенд-маршрутизатор автоматически находит указанный путь и использует внутреннюю маршрутизацию.

В случае отсутствия запрашиваемого файла Apache перенаправляет все запросы к **/html** на **index.html**. Для этого необходимо, чтобы был установлен и активирован модуль **mod_rewrite**.

Модуль Apache **mod_rewrite**: https://httpd.apache.org/docs/current/mod/mod_rewrite.html

В папке **html** создайте **.htaccess** и добавьте следующий контент:

```
RewriteEngine On
RewriteCond %{DOCUMENT_ROOT}%{REQUEST_URI} -f [OR]
RewriteCond %{DOCUMENT_ROOT}%{REQUEST_URI} -d
RewriteRule ^ - [L]
RewriteRule ^ index.html
```

Данный код можно также добавить в конфигурацию Apache.

7.2.4. Опционально: отключение информации о сервере

По соображениям безопасности рекомендуется не показывать информацию о сервере на страницах об ошибке. Такая информация может быть использована злоумышленниками для нахождения уязвимостей на сервере и атаки на систему через эти уязвимости.

Последовательность действий

Сначала в папке **<$TOMCAT_HOME>/lib/** необходимо создать следующий каталог:

```
org/apache/catalina/util/
```

Затем в папке **util** нужно создать текстовый файл **ServerInfo.properties**.

Добавлением в этот файл следующей строки можно регулировать, как информация должна отображаться:

```
server.info=
```

После запуска Tomcat на страницах об ошибке отображается информация о сервере, указанная после знака **=**.

Если ничего не указано, не отображается вся строка.

7.2.5. Опционально: переадресация с HTTP на HTTPS

По соображениям безопасности компания "СДИ Софт" не рекомендует использовать данную конфигурацию. Переадресацию можно выполнить, например, соответствующим образом настроив Tomcat. См. документацию на Tomcat на предмет процедуры настройки переадресации.

ПРИМЕЧАНИЕ

Для применения изменений необходимо перезапустить ***Tomcat.***

7.3. Конфигурирование лицензий

7.3.1. Установка идентификатора лицензий

Для настройки поведения лицензий в файле ***command_config.properties*** необходимо прописать следующее:

```
command.config.specific.licenseSystemId=<licSystemId>
```

Идентификатор ***<licenseID>*** указан в договоре

7.3.2. Загрузка лицензий

В системе Базис версии 14.0 лицензии загружаются в модуль "Администрирование". При этом необходимо, чтобы администратор вошел в систему под именем "***licadmin***".

7.4. Дополнительные варианты конфигурирования

7.4.1. Активирование серверных задач

Поскольку управление индексированием осуществляется при помощи серверных задач, их нужно активировать в файле **basis_config.properties**. Для этого для **basis.config.specific.serverJobs** установите значение **enabled**.

После чего нужно указать правильный путь к индексу и удалить символы комментария (#):

```
basis.config.specific.serverJobs=enabled
basis.config.searchindex.maxParallelThreads=4
basis.config.searchindex.commitThreshold=250
basis.config.searchindex.default.id=default
basis.config.searchindex.default.path=[абсолютный_путь]
```

7.4.2. Опционально: настройка индексации языков

При помощи параметра **SYS_EASY_SEARCH_LANGUAGES** можно задавать, какие из лицензированных языков должны включаться в индекс. Для указания нескольких языков используется символ разделительной линии (|). Поддерживаются следующие значения:

- ru_RU для русского языка
- en_US для английского языка

7.4.3. Опционально: конфигурирование папки для хранения индекса

При помощи свойства ***basis.config.searchindex.default.path*** в файле ***basis_config.properties*** можно изменять текущий путь к папке с индексом.

Для создания нескольких синхронизируемых индексов в узле ***searchindex*** необходимо создать дополнительный узел. При этом резервные индексы распределяются по нескольким различным серверам, что повышает отказоустойчивость индекса; отпадает необходимость создания нового индекса при появлении проблем с доступом.

```
basis.config.searchindex.default.id=default
basis.config.searchindex.default.path=[абсолютный_путь]

basis.config.searchindex.alternative.id=alternative
basis.config.searchindex.alternative.path=[абсолютный_путь]
```

7.4.4. Опционально: конфигурирование максимального количества параллельных процессов для индексирования

Определить максимальное число параллельных процессов для индексирования можно при помощи параметра **basis.config.searchindex.maxParallelThreads** в файле **basis_config.properties**.

```
basis.config.searchindex.maxParallelThreads=2
```

7.4.5. Опционально: конфигурирование буферизации данных

Задать, после какого числа проиндексированных объектов данные должны сохраняться в индекс, можно при помощи параметра **basis.config.searchindex.commitThreshold** в файле **basis_config.properties**.

Пример с двумя синхронизированными индексами и рекомендованными значениями для

индексирования (4 параллельных процесса и 250 объектов):

```
basis.config.searchindex.maxParallelThreads=4
basis.config.searchindex.commitThreshold=250
basis.config.searchindex.default.id=default
basis.config.searchindex.default.path=[абсолютный_путь]
basis.config.searchindex.alternative.id=alternative
basis.config.searchindex.alternative.path=[абсолютный_путь]
```

7.4.6. Опционально: шифрование basis_config.properties

При необходимости файл **basis_config.properties** можно сохранить в зашифрованном виде. Шифрование осуществляется посредством алгоритма AES с использованием автоматически сгенерированного ключа. Затем последний шифруется по алгоритму RSA при помощи заданного открытого ключа и добавляется к новому файлу **basis_config.properties**.

Для того, чтобы зашифровать существующий файл **basis_config.properties**, необходимо вызвать специальную утилиту, т.е. исполняемый файл **Jar** с именем **basis-file-encrypter-[номер версии].jar**

Он входит в состав пакета **basis-standard-[номер версии]-basis_file_encrypter.zip** из папки инструментов **tools** и запускается с помощью следующей команды (в качестве примера – Базис 13.4):

```
java -jar basis-file-encrypter-13.4.0.jar "путь/basis_config.properties"
```

Путь к файлу **basis_config.properties** передается в виде параметра.

Для его исполнения требуется Java версии не ниже 11.

Зашифрованный файл **Config** сохраняется вместе с существующим файлом.

После создания и проверки зашифрованного файла в исходном файле **basis_config.properties** можно изменить важные позиции.

7.4.7. Опционально: интеграция LDAP

Последовательность действий

Конфигурирование

Внесите следующие изменения в файл **<$TOMCAT_HOME>/conf/basis_config.properties**:

Свойства нужно определить отдельно для каждого сервера LDAP. Возможно также конфигурирование нескольких серверов LDAP. При этом все серверы LDAP будут опрашиваться до тех пор, пока либо не последует успешное подтверждение подлинности, либо не будет достигнут конец списка серверов LDAP. Следующие параметры являются обязательными в зависимости от используемого протокола:

protocol: Протокол, который должен использоваться для обмена данными между сервером приложений и сервером LDAP. Возможные значения: ***ldap*** и ***ldaps***. При ***ldaps*** обмен данными шифруется с помощью SSL.

host: Имя или IP-адрес сервера LDAP, который должен использоваться.

port: Порт, используемый для доступа к службе LDAP.

searchUser: Пользователь, для которого Базис проверяет введенные данные аутентификации.

searchPassword: Пароль пользователя searchUser.

authMech: simple – данные аутентификации передаются серверу LDAP в незашифрованном виде.

опционально – **keystore**: Путь к файлу Keystore. Требуется только в случае ldaps.

опционально – **keystorePassword**: Пароль для указанного файла Keystore.

serverParam: В зависимости от введенного сервера возможны несколько свойств **serverParam**

searchContext: Узел дерева LDAP, на котором осуществляется поиск введенных данных аутентификации.

searchFilter: Свойство LDAP, которое должно совпадать с введенным именем пользователя {имяпользователя}.

Пример:

```
basis.config.ldapserver.ldapsrv1.active=false
basis.config.ldapserver.ldapsrv1.protocol=ldap
basis.config.ldapserver.ldapsrv1.host=[сервер_базы_данных]
basis.config.ldapserver.ldapsrv1.port=389
basis.config.ldapserver.ldapsrv1.searchUser=[пользователь_LDAP]
basis.config.ldapserver.ldapsrv1.searchPassword=[пароль_пользователя_LDAP]
basis.config.ldapserver.ldapsrv1.authMech=simple
basis.config.ldapserver.ldapsrv1.keystore=[путь_к_файлу_Keystore]
basis.config.ldapserver.ldapsrv1.keystorePassword=[пароль_Keystore]
basis.config.ldapserver.ldapsrv1.passwordEscaped=false
basis.config.ldapserver.ldapsrv1.context1.searchContext=[контекст_поиска_
LDAP]
basis.config.ldapserver.ldapsrv1.context1.searchFilter=sAMAccountName={имя
_пользователя}
```

Чтобы включить аутентификацию через LDAP, параметр **SYS_USE_LDAP_LOGIN** в таблице **STFCFG_SETTING** в схеме Базис нужно изменить следующим образом:

- Войдите в базу данных под именем пользователя **basis** через SQL*Plus.
- Выполните следующее выражение и подтвердите оператором **commit**:

```
update stfcfg_setting set value='TRUE'
where name='SYS_USE_LDAP_LOGIN';
```

- В модуле "Управление доступом" в разделе "Пользователь" активируйте аутентификацию через LDAP

ВНИМАНИЕ

Имя пользователя в Базис должно полностью совпадать с именем пользователя LDAP.

Кроме того, аутентификацию через LDAP для всех пользователей можно активировать через базу данных с помощью следующего выражения:

```
update stcsys_user set authorization_mode='3';
```

- Завершите транзакцию оператором **commit;**.

Если необходимо, чтобы к Базис могли подключаться пользователи, существующие только в LDAP, в Базис можно создать пользователя **fallback** и добавить его в любую группу, например, **role_read**. Такого пользователя можно создать в файле **basis_config.properties** с помощью свойства **fallbackUser**. В результате пользователь, существующий в LDAP, но отсутствующий в Базис, будет заходить в Базис под именем, заданным в свойстве **fallbackUser**.

Пример:

```
basis.config.common.fallbackUser=[пользователь_FALLBACK]
```

7.4.8. Опционально: шифрование паролей в базе данных

Шифрование может потребоваться, когда пароль необходимо проверять не только на соответствие требованиям, но когда он используется в виде простого текста для входа во внешнюю систему.

Соединения, требующие пароля в формате простого текста:

- Соединения базы данных с другими системами из модуля "Отчетность".
  Таблица: STCCFG_DATABASE_CONNECTION
- Соединение с базой данных из подсистемы "ЦОД".
  Таблица: STFDCE_CONNECTION_DATABASE
- Соединения по FTP из подсистемы "ЦОД".
  Таблица: STFDCE_CONNECTION_FTP

Для этих таблиц шифруется каждое имя пользователя с соответствующим паролем.

Для декодирования требуется пароль. Он указан в **basis_config.properties** для каждого случая использования. Таким образом, внутренний доступ всегда осуществляется посредством псевдонима. Он в данный момент идентичен имени таблицы, содержащей столбцы, подлежащие шифрованию.

Поскольку псевдоним, использующийся для шифрования, "зашит" в коде, пользователь не может его изменить.

Для каждого пароля требуются три свойства.

- **Alias**
  Псевдоним, для которого должен быть задан пароль. Он в данный момент идентичен имени таблицы, содержащей столбцы, подлежащие шифрованию.
- **Secret**
  Пароль, используемый для шифрования.
- **oldSecret**
  Пароль, который использовался для шифрования ранее.
  Эти три свойства необходимы для смены паролей.

```
basis.config.secrets.STFSYS_TEST_COLUMN.alias=STFSYS_TEST_COLUMN
basis.config.secrets.STFSYS_TEST_COLUMN.secret=TEST_SECRET
basis.config.secrets.STFSYS_TEST_COLUMN.oldSecret=
```

Шифрование таблицы:

Случай 1: Таблицу необходимо зашифровать.

- Задайте пароль в атрибуте "secret".
- Перезапустите сервер приложений.

Случай 2: Пароль для шифрования нужно изменить

- Укажите предыдущий пароль в атрибуте "oldSecret".
- Задайте новый пароль в атрибуте "secret".
- Перезапустите сервер приложений.
- В ходе первого перезапуска пароли декодируются с помощью старого пароля, шифруются с помощью нового пароля и обновляются в базе данных. После завершения обновления записей в базе данных значение свойства "oldSecret" можно удалить из **basis_config.properties.**

Случай 3: Таблицу необходимо декодировать (после чего значения будут отображаться в базе данных в виде простого текста)

В этом случае используется та же процедура, что и при смене пароля.

- Свойство "secret" остается пустым, свойство "oldSecret" содержит пароль, с помощью которого содержимое таблицы зашифровано в данный момент.
- После перезапуска значения хранятся в базе данных в виде простого текста.
- После завершения декодирования значения "secret" и "oldSecret" можно удалить.

7.4.9. Опционально: изменение языка экрана авторизации и пользовательских языков

Для изменения языка экрана авторизации используйте следующий скрипт:

```
update stfcfg_setting set value = '<lang>'
where name = 'MOD_LANGUAGE' and user_elid = 'DEFAULT_USER'
/
commit
/
```

Для изменения языка для всех существующих пользователей используйте следующий скрипт:

```
update stfcfg_setting set value = '<lang>'
where name = 'MOD_LANGUAGE'
/
commit
/
```

Для параметра <lang> можно задать следующие значения:

- en_US
- ru_RU (необходимо установить дополнительно)

7.4.10. Опционально: работа приложения Базис со шлюзом единого входа

При работе приложения Базис со шлюзом единого входа необходимо настроить сервер приложений и систему Базис под другой контекст.

В качестве контекста необходимо использовать **/app/basis**. Поддерживается только тип доступа "bearer-only". Необходимо наличие JWT-токена. Проверка осуществляется по сконфигурированному Keycloak.

```
basis.config.common.keycloakUrl=
basis.config.common.keycloakRealm=SDI-Application
basis.config.common.keycloakCertUrlPattern=%s/auth/realms/%s/protocol/op
enid-connect/certs
```

7.4.11. Опционально: работа приложения Базис с OAuth

Базис предусматривает возможность аутентификации посредством сервера авторизации, поддерживающего стандарт OpenID Connect.

Для этого необходимо выполнить следующие настройку в **basis_config.properties**. В качестве примера используется Keycloak.

```
# Если true – аутентификация по OpenlD Connect активирована
basis.config.oidc.enabled=true
# URL-адрес центра выдачи OpenID Connect
basis.config.oidc.issuerUrl=https://keycloak-host/auth/realms/basis
basis.config.oidc.clientId=basis
basis.config.oidc.clientSecret=5961d228-1cd9-4a32-9155-bb765bbc21aa
# URL-адрес, по которому осуществляется доступ к интерфейсу "Базис"
basis.config.oidc.frontendUrl=https://basis-frontend-host/basis
```

Клиент должен определить **clientID** на сервере аутентификации самостоятельно. Сервер авторизации генерирует соответствующий пароль **clientSecret**. Если сам сервер приложений Базис не сообщает адрес внешнего интерфейса Базис, т.е. он имеет другой URL-адрес, нужно точно указать параметр **frontendUrl**.

ПРИМЕЧАНИЕ

"СДИ" осуществляет конфигурирование только приложения Базис. За конфигурирование сервера аутентификации отвечает клиент.

Чтобы обеспечить прямой доступ к приложению Базис несмотря на использование OpenID Connect, для избранных пользователей можно заранее санкционировать прямой вход. Это нужно сделать перед конфигурированием OpenID Connect, поскольку в противном случае единственной возможностью будет аутентификация через OpenID Connect.

Чтобы активировать прямой вход, для соответствующих пользователей в модуле "Управление доступом" нужно активировать опцию "**Разрешить прямой вход при использовании OIDC**". Если такой пользователь хочет входить в систему напрямую, URL-адрес Базиса нужно дополнить следующим: **?directlogin=true**

Возможность прямого входа останется до момента удаления cookies или добавления следующего параметра: **?directlogin=false**

Опционально можно сконфигурировать атрибут OpenID Connect Claim and Basis User,

который используется для определения пользователя basis для входа. По умолчанию Базис использует preferred_username на сервере OpenID Connect и имя пользователя. Поэтому в ID-токене должен существовать используемый клейм.

При другой конфигурации необходимо сконфигурировать **basis_config.properties** следующим образом. В качестве примера здесь описывается вход по адресу электронной почты.

```
# имя клейма, используемое для входа
# напр. preferred_username, email
basis.config.oidc.userLoginClaim=email
# Атрибут таблицы пользователей, используемый для входа
# напр. USER_NAME, E_MAIL
basis.config.oidc.userLoginAttribute=E_MAIL
# Флаг, определяющий, учитывается ли регистр при сравнении атрибутов
аутентификации при входе пользователя
basis.config.oidc.userLoginCaseInsensitive = true
```

8. Дополнительные варианты установки

8.1. Вариант 1

В этом варианте к одной общей базе данных параллельно подключены несколько серверов TOMCAT. Его можно использовать, например, для равномерного распределения нагрузки или повышения отказоустойчивости.

8.1.1. Настройки для многосерверной архитектуры (кластера)

Для того, чтобы клиентский сеанс был действителен на всех серверах приложений, приложение Базис должно быть сконфигурировано особым образом. Кроме того, в таком варианте весь динамически создаваемый контент (например, графические изображения для новых импортированных компонентов) должен автоматически распределяться по всем серверам приложений.

Для конфигурирования кластера необходимо соблюдение следующих требований:

- Отдельные узлы в кластере должны находиться в одной сети, должны видеть друг друга и взаимодействовать друг с другом. Порты членов кластера должны быть закрыты для доступа со стороны конечного пользователя, и могут быть доступны только для других узлов кластера. Это может быть сделано с помощью MTLS (Mutual TLS), соответствующей настройки межсетевого экрана или сети.
- В ходе установки ПО для всех узлов кластера должна быть сконфигурирована одна база данных.

Конфигурирование

Внесите следующие изменения в файл **<$TOMCAT_HOME>/conf/basis_config.properties**:

Для параметра **multiserver** нужно указать true на всех серверах приложений.

```
basis.config.common.multiserver=true
```

Параметр **serverJobs** нужно установить на одном сервере приложений на **enabled**, а на всех остальных серверах приложений на **disabled** или **fallback**.

```
# настройка управления задачами для конкретного сервера
# указывает, может ли сервер создавать поток команд и должны ли планироваться задачи. Многосерверная архитектура: только один сервер должен создавать потоки команд!
# disabled - ThreadManager и задачи отключены,
# enabled - ThreadManager и задачи активированы,
# fallback - сервер может создавать потоки команд и планировать задачи в случае выхода из строя основного сервера
# threadManagerOnly - ThreadManager активирован, а задачи отключены,
# cronjobsOnly - ThreadManager отключен, а задачи активированы

basis.config.specific.serverJob
```

База данных имеет различные настройки для конфигурирования порта члена кластера:

- **SYS_CLUSTER_BIND_PORT**: Значение по умолчанию: 5701. Член кластера резервирует данный порт, чтобы другие члены могли взаимодействовать с ним.

Если данный порт занят, член кластера ищет свободный порт путем перебора портов.

- **SYS_CLUSTER_BIND_PORT_COUNT**: Значение по умолчанию: 100. Стандартно, резервируется один из ста портов, который используется для взаимодействия в рамках кластера. Т.е. если вы указывает 5701 в качестве номера порта члена кластера, при входе членов в кластер Базис пытается найти порты в диапазоне между 5701 и 5801. Количество портов можно изменить.
- **SYS_CLUSTER_OUTBOUND_PORT**: Стандартно, Базис разрешает поиск любого порта в ходе действия socket-bind. При этом политики безопасности/межсетевые экраны могут ограничивать использование портов исходящего трафика. Чтобы соблюсти данное требование, Базис можно настроить так, чтобы использовались исключительно определенные исходящие порты. Можно задать диапазон портов и/или несколько портов через запятую.
- **SYS_CLUSTER_PREFER_IPV4**: Значение по умолчанию – Y. При выборе локального адреса приоритетным является сетевой интерфейс IPv4.

Функция поиска

При работе в многосерверном режиме поисковый индекс необходимо сохранить на общем устройстве, проиндексировав экземпляр Базис, и предоставить доступ к нему другим серверам.

ПРИМЕЧАНИЕ

При том, что для следующих вариантов также необходимо конфигурировать многосерверный режим, эти настройки далее по документу не описываются отдельно.

8.2. Вариант 2

В этом варианте разные сервера Apache и Tomcat работают на разных аппаратных устройствах. Распределитель нагрузки распределяет нагрузку между разными физическими серверами, так что для всех пользователей обеспечивается максимальная производительность. Все работающие веб-серверы обращаются к одной базе данных Oracle.

Чтобы сократить вероятность ошибки в многосерверном режиме, необходимо активировать функцию **BALANCER_SESSION_STICKY** Распределителя нагрузки. В противном случае другие функции, например, функция импорта данных объектов, могут создавать проблемы.

8.2.1. Конфигурирование распределителя нагрузки

Для работы распределителя нагрузки необходимо, чтобы на сервере Apache был установлен модуль **mod_proxy_balancer**, и чтобы он загружался при запуске сервера Apache.

Для базовой функциональности распределителя нагрузки достаточно добавить следующую запись в файл **httpd-vhosts.conf** сервера Apache, на котором установлен распределитель нагрузки.

```
<Proxy balancer://project1>
BalancerMember http://192.168.0.30:80
BalancerMember http://192.168.0.31:80
</Proxy>
<Location>
ProxyPass / balancer://project1/
</Location>
```

Можно настроить соответствующие IP-адреса и порты сервера Apache, между которыми должна распределяться нагрузка.

ПРИМЕЧАНИЕ

Поскольку в конфигурации серверов Apache ограничено количество IP-адресов для доступа для формирования страницы, IP-адрес сервера должен находиться в том же определенном диапазоне IP-адресов, что и в записи, в противном случае нужно изменить следующие строки в файлах **httpd-vhosts.conf** серверов Apache.

```
<Proxy *>
AddDefaultCharset Off
Order deny,allow
Allow from all
Allow from 192.168 <= Этот адрес нужно соответствующим образом изменить
или удалить
</Proxy>
```

ПРИМЕЧАНИЕ

Если удалить строку **Allow from xxx.xxx**, доступ может быть выполнен с любого IP-адреса. Это несет в себе опасность, поэтому лучше указать диапазон IP-адресов сервера, на котором установлен распределитель нагрузки.

8.2.2. Конфигурирование функции BALANCER_SESSION_STICKY

ПРИМЕЧАНИЕ

Данная функция протестирована только для Apache версии до 2.0 включительно. Для более новых версий или версий от других поставщиков Apache данная конфигурация может отличаться, функция может отсутствовать или иметь другое название. В этом случае информацию о необходимых настройках ищите в соответствующей документации на сервер приложений.

Как было упомянуто в начале описания данного варианта, для корректной работы Базис данная функция должна быть активирована. Поскольку, пользователь, запустив выгрузку файла, после ее завершения может попасть на другой веб-сервер в результате повторного установления соединения. Но, так как файл был загружен на другой сервер, пользователь не получит доступ к своему файлу. Функция **Session Sticky** обеспечивает, что каждый пользователь будет работать с одним веб-сервером вплоть до завершения сеанса работы.

Нужно изменить конфигурацию в файле **httpd-vhosts.conf** сервера с распределителем нагрузки следующим образом:

```
<Proxy balancer://project1>
BalancerMember http://192.168.0.30:80 route=19216803080
BalancerMember http://192.168.0.31:80 route=19216803180
</Proxy>
<Location>
ProxyPass    balancer://project1/    lbmethod=byrequests
stickysession=JSESSIONID
ProxyPassReverse http://192.168.0.30:80/
ProxyPassReverse http://192.168.0.31:80/
</Location>
```

Для параметра **lbmethod** на данный момент существуют 3 возможных значения:

1. **Byrequest**: Распределяет нагрузку между серверами по количеству запросов.

При помощи параметра **loadfactor** (число в пределах между 0 и 100) в **BalanceMember** можно задать процентное соотношение нагрузки между серверами.

```
BalancerMember http://192.168.0.30:80 loadfactor=4 route=19216803080
BalancerMember http://192.168.0.31:80 loadfactor=6 route=19216803180
```

В данном примере первый сервер группы получает 40% нагрузки, а второй – 60%.

2. **Bytraffic**: Распределяет нагрузку между серверами по объему трафика.
3. **Bybusyness**: Передает нагрузку тому серверу, который меньше всего загружен.

9. Конфигурирование опциональных модулей

9.1. Почтовый сервер по умолчанию

Условия

Различные модули Базис (напр., модуль "Отчетность") предусматривают возможность отправки электронных сообщений на определенный почтовый сервер. Для этого необходимо настроить почтовый сервер по умолчанию.

Последовательность действий

Конфигурирование

Внесите следующие изменения в файл **<$TOMCAT_HOME>/conf/basis_config.properties:**

- Почтовый сервер, через который Базис отправляет электронные сообщения:

```
basis.config.common.mailServerName=smtp.localhost.ru
```

- Порт, используемый для доступа к почтовому серверу:

```
basis.config.common.mailServerPort=25
```

- Настройка, ожидает ли почтовый сервер аутентификации для отправки электронных писем (true/false)

```
basis.config.common.mailServerUseAuth=false
```

- Имя пользователя и пароль для аутентификации на почтовом сервере, когда **mailServerUseAuth – true**:

```
basis.config.common.mailServerUsername=
basis.config.common.mailServerPassword=
```

- Отправитель для почтовых сообщений, отправляемых из Базис, если не указан никакой другой отправитель:

```
basis.config.common.mailSenderAddress=basis@example.ru
```

- Адрес для направления ответа от почтового сервера в случае сбоя (напр., когда адрес получателя неизвестен):

```
basis.config.common.mailErrorReplyAddress=basis@example.ru
```

- Настройка, определяющая, должен ли Базис сохранять сообщение в базе данных после его отправки:

```
basis.config.common.saveMailInDB=false
```

9.2. Модуль "Отчетность"

Условия

Модуль "Отчетность" СДИ «Базис» может создавать определенные отчеты в качестве задач. Для этого требуется соответствующая настройка входа в СДИ «Базис» (пользователь/группа/домен). Задача может включать в себя экспорт отчетов в файлы заданных форматов и их отправку по электронной почте определенной группе получателей. Для отправки таких электронных сообщений используется почтовый сервер по умолчанию, о котором шла речь выше. Поэтому для автоматической отправки писем из модуля "Отчетность" необходимо настроить почтовый сервер так, как это описано в разделе "Почтовый сервер по умолчанию".

Необходимое условие: Конфигурирование метасхемы Базис (см. раздел 5.1).

Последовательность действий

Конфигурирование

Внесите следующие изменения в файл **<$TOMCAT_HOME>/conf/basis_config.properties**:

- Пользователь, от имени которого задача модуля "Отчетность" подключается к приложению Базис:

```
basis.config.common.qedJobUserName=basis
```

- Роль, под которой должны выполняться задачи модуля "Отчетность":

```
basis.config.common.qedJobUserGroupId=role_admin_1001
```

- Домен, в котором выполняются задачи модуля "Отчетность".

```
basis.config.common.qedJobManId=1001
```

- Если пользователь является членом определенной группы или домена (по умолчанию – группы):

```
basis.config.common.qedJobUGType=G
```

- Адрес электронной почты отправителя, с которого модуль "Отчетность" отправляет сообщения:

```
basis.config.common.qedMailSenderAddress=[MAIL_ADDRESS_SENDER]
```

Чтобы отчеты модуля "Отчетность" запускались в качестве задач, нужно это активировать:

```
basis.config.specific.serverJobs=enabled
```

ПРИМЕЧАНИЕ

При многосерверном режиме данная настройка должна быть задана только для одного сервера.

9.3. Модуль "ЦОД"

9.3.1. Активирование модуля "ЦОД"

Чтобы активировать модуль "ЦОД", необходимо прежде всего остановить Tomcat.

Модуль также необходимо активировать в таблице **stfsys_sw_unit**. Для этого в схеме Базис выполните следующее выражение SQL и подтвердите его с помощью оператора “**commit;**":

```
update stfsys_sw_unit set enabled = 'Y'
where sw_unit in
('AIRCON','DATACENTERCOCKPIT','DATACENTERCOCKPIT:FOOTPRINT',
'POWERMGMT','POWERMGMT:CIRCUITDIAGRAMLIST','POWERMGMT:GRAPHICAL',
'DATACENTERCOCKPIT:FOOTPRINT3D');
```

После чего перезапустите Tomcat.

ПРИМЕЧАНИЕ

Активирование модуля "ЦОД" следует выполнять только после консультаций с "СДИ Софт". Поскольку модуль не может активироваться/деактивироваться по собственному усмотрению. Компания "СДИ Софт" должна активировать данные ЦОД в файле DAT, загружаемом вместе с данным модулем.

Установка

- Вручную запустить импорт данных измерений можно через меню **Импорт – Запустить** в модуле "Электропитание". Какой именно файл будет считываться в ходе импорта – определяется экземпляром базы данных в таблице **STFJOB_SETTING**. Для настройки пути к файлу в схеме Базис нужно запустить следующее выражение SQL (завершив его оператором "**commit;**").

```
update stfjob_setting s
set where s.value = 'абсолютный адрес с именем файла>'
s.job_spec_elid = (select m.infos from stcjob_master m
                   where m.name =
'ImportMeasuringData' and m.job_group = 'Powermgmt')
and s.name = 'IMPORT_FILE';
```

- Чтобы импорт данных измерений выполнялся в качестве задачи, необходимо активировать задачу **ImportMeasuringData** из группы задач **Powermgmt** в таблице **STCJOB_MASTER**. Для этого запустите следующее выражение SQL в схеме Базис:

```
update stcjob_master m set
      m.user_name = '<USER_NAME>',
      m.man_id = <MANDATOR_ID>,
      m.group_name = '<USER_GROUP_NAME>',
      m.group_type = '<USER_GROUP_TYPE>'
where m.name ='ImportMeasuringData' and m.job_group =
'Powermgmt';
```

- USER_NAME: Пользователь, от имени которого выполняется задача.
- MANDATOR_ID: Домен, в который входит пользователь.
- USER_GROUP_NAME: Имя пользователя или группы, под которым пользователь входит в домен.
- USER_GROUP_TYPE: Определяет, что представляет значение в параметре **user_group_id** – группу (G) или пользователя (U).
- Также в интерфейсе пользователя модуля "Администрирование" – Задачи – Сервер необходимо указать время выполнения задачи. Для автоматического выполнения задачи нужно поставить флажок у опции "Действующая".
- Для того, чтобы сконфигурированные задачи могли выполняться в приложении Базис, параметр **serverJobs** в файле **<$TOMCAT_HOME>/conf/basis_config.properties** должен быть установлен на значение **enabled**.

```
basis.config.specific.serverJobs=enabled
```

ПРИМЕЧАНИЕ

При многосерверном режиме данная настройка должна быть задана только для одного сервера.

Переключение с БТЕ/ч на кВт:

В модуле "ЦОД" можно изменить единицу БТЕ/ч на кВт. Для этого в схеме Базис выполните следующие выражения SQL:

```
update stfcfg_setting set value = 'kW'
where name = 'SYS_CLIMATE_CAPACITY_UNIT';
update stfsys_msg_catalog set msg_text = 'kW', custom = 'Y'
where catalog_name = 'CLIMATE_CAPACITY_UNIT' and msg_id = 28;
```

9.4. Модуль "Телеком"

Если сетевой элемент не привязан вручную к определенной зоне, он назначается стандартной зоне. Если такой зоны еще не существует, ее можно переименовать с помощью следующего выражения SQL:

```
update stfcfg_setting set value='[Автономная система]'
where name='TCO_NE_DEFAULT_ZONE_NAME;
```

Если зона уже существует, переименовать ее может только служба поддержки компании "СДИ Софт".

Чтобы символы из других алфавитов (например, с умляутами) корректно импортировались в модуль "Телеком", кодировка должна совпадать с кодировкой импортируемого файла Excel. Для изменения кодировки используется запись **MOD_EXCEL_IMPORT_JXL_ENCODING** в таблице базы данных **STFCFG_SETTING**.

Необходимо в схеме Базис выполнить следующее выражение SQL, добавив в конце оператор "**commit;**":

```
update stfcfg_setting set value='<кодировка>'
where name='MOD_EXCEL_IMPORT_JXL_ENCODING';
```

9.5. Модуль "3D план помещения"

3D план помещения – графический модуль, предназначенный для объемной визуализации ЦОДов. Модуль "3D план помещения" запускается из модуля "ЦОД". Он создает объемные виртуальные представления на основе существующих двухмерных схем.

Активирование модуля

- Остановите Tomcat.
- В SQL Plus / SQL Editor выполните следующее выражение, добавив в конце "**commit;**".

```
update stfsys_sw_unit set enabled='Y'
where sw_unit = 'DATACENTERCOCKPIT:FOOTPRINT3D';
```

- Запустите Tomcat.

9.6. Запросы Scroll (для интеграционного слоя)

Запросы scroll используются, чтобы запрашивать большие объемы данных посредством интеграционного слоя. При использовании запросов scroll данные для кэширования сохраняются в файловой системе. Путь для сохранения таких данных указан в файле **<$TOMCAT_HOME>/conf/basis_config.properties**. Чтобы отделить эти данные от данных Базис, рекомендуется выделить место для их хранения на другом жестком диске или в другом разделе диска. Это позволит избежать проблем в работе в случае нехватки свободного дискового пространства. За дополнительной информацией о необходимом размере свободного места на диске обращайтесь к документации с требованиями к системе.

ВНИМАНИЕ

Использование запросов **scroll** в многосерверном режиме с использованием распределителя нагрузки не предусмотрено. Все запросы всегда должны генерироваться с одного сервера приложений. Кроме того, путь к запросу не может использоваться несколькими серверами приложений.

Конфигурирование

Внесите следующие изменения в файл **<$TOMCAT_HOME>/conf/basis_config.properties.**

// Конфигурирование опциональных модулей

Задайте путь к запросу **scroll**:

```
basis.config.specific.scrollQueryDataLocation=${basis.config.specific
.wmIconsHome}/scrollQueryData/
```

10. Обозначение модулей системы Базис

Пример активирования модуля представлен в разделе 8.3.

Сегмент	Программный блок	Модуль
Базовый учет	ACCESSMGMT	Управление доступом
Базовый учет	ACM	Контракты и Организации
Базовый учет	ADMINISTRATION	Администрирование
Расширенный учет	AIRCON	Кондиционирование
Кабельный учет	ASSIGNMENTLIST	Кабельный журнал
Базовый учет	CIMGMT	Управление КЕ
Базовый учет	CIMGMT:GRAPHIC	Управление КЕ – Графическое изображение КЕ
Расширенный учет	POWERMGMT:CIRCUITDIAGRAMLIST	Список подключенных устройств
Расширенный учет	CLIENTMGMT	Рабочие места
Базовый учет	EDM	Конфигурационные данные
Кабельный учет	CONNECTIONMATRIX	Матрицы коммутации
Кабельный учет	CONNECTION	Соединения
Расширенный учет	DATACENTERCOCKPIT	ЦОД
Базовый учет	ENTITYMGR	Моделирование
Расширенный учет	DATACENTERCOCKPIT:FOOTPRINT	План помещения
Расширенный учет	ARCGIS	Геокарты
Расширенный учет	POWERMGMT:GRAPHICAL	Графическое представление
Кабельный учет	INVENTORYMGMT	Склады
Расширенный учет	IPMGMT	IP-адресация
Кабельный учет	JUNCTIONBOX	Муфты

// Обозначение модулей системы Базис

Базовый учет	LCM	Жизненный цикл
Базовый учет	NAVIGATOR	Навигатор
Базовый учет	NAVIGATOR:PLANVIEW	Навигатор (план)
Кабельный учет	NETSPIDER	Схемы
Базовый учет	NETWORKINVENTORY	Сеть
Расширенный учет	NODEMGMT	Узлы
Базовый учет	OBJMGMT	Учетные карточки
Базовый учет	PLANNING	Планирование
Расширенный учет	POWERMGMT	Электропитание
Расширенный учет	QUERYEDITOR	Отчетность
Расширенный учет	RADIOMGMT	Радиосети
Расширенный учет	SRM	Серверы и СХД
Расширенный учет	SIGNALMGMT	Доступ и телефония
Кабельный учет	SIGNALTRACING	Трассировка сигнала
Расширенный учет	SOFTWAREMGMT	Лицензии и ПО
Кабельный учет	CABINET	Стойка
Кабельный учет	TCCABINET	Кросс
Расширенный учет	TELCO	Телеком
Расширенный учет	TELCO:CIRCUITRYREPORT	Телеком – Отчет о подключенных услугах
Кабельный учет	TRAYMGMT	ЛКС
Базовый учет	BASIS:ATTACHMENT	Вложения
Расширенный учет	WAN	WAN
Расширенный учет	WEBGIS_EXTERNAL	Интерфейс WebGIS
Расширенный учет	WEBGISMGMT	WebGIS

// Обозначение модулей системы Базис

Базовый учет	WORKFLOW	Поток операций
Расширенный учет	DATACENTERCOCKPIT:FOOTPRINT3D	3D план помещения

11. Выходные данные

«СДИ Софт»

Россия, 107045, г. Москва,
ул. Трубная, д.12
Телефон: +7 (499) 495-10-42
Интернет: http://www.sdisoft.ru
Электронная почта: info@sdisoft.ru

Несмотря на тщательную подготовку текста, в документе могут содержаться ошибки и неточности.

Компания «СДИ Софт» не несет ответственности за возможный ущерб в результате использования изложенной в документе информации.

Любые предложениям по улучшению содержания документа и указания на неточности и ошибки приветствуются.