

АО «РТ Лабс»

Руководство системного администратора

**ИНФОРМАЦИОННАЯ СИСТЕМА «БИОМЕТРИЧЕСКАЯ СИСТЕМА КОНТРОЛЯ
ДОСТУПА»**

АО «РТ ЛАБС»

На 24 листах

Москва 2019

ОГЛАВЛЕНИЕ

РУКОВОДСТВО СИСТЕМНОГО АДМИНИСТРАТОРА.....	1
АННОТАЦИЯ	4
ТЕРМИНЫ И ОПРЕДЕЛЕНИЯ.....	5
1. ОБЩИЕ СВЕДЕНИЯ	6
1.1. Назначение документа	6
1.2. Уровень подготовки пользователей	6
2. УСТАНОВКА И НАСТРОЙКА POSTGRESQL 10.....	6
2.1. Подготовка серверов PGSQL	6
2.1.1. Разметка диска по стандарту	6
2.1.2. Установка пакетов и настройка базы данных.....	7
2.2. Установка postgresql 10.....	7
2.3. Конфигурация базы данных	8
2.4. Перезапуск сервиса для применения конфигураций	12
2.5. Скрипт для создания системных и прикладных пользователей	12
3. УСТАНОВКА И НАСТРОЙКА RABBITMQ.....	13
3.1. Подготовка серверов	13
3.2. Разметка диска	13
3.3. Установка Rabbitmq.....	13
4. УСТАНОВКА И НАСТРОЙКА JAVA STACK.....	14
4.1. Разметка диска	14
4.2. создание системного пользователя BOISKUD	15
4.3. JAVA (JDK1.8.181).....	15
4.4. FRONTEND (BOISKUD-WEB.JAR)	15
4.4.1. Подготовка OS (Под пользователем root)	15
4.4.2. Установка FRONTEND (BOISKUD-WEB.WAR).....	16
4.4.3. Конфигурация FRONTEND (BOISKUD-WEB.WAR).....	16
4.4.4. Выставить по необходимости -Xms -Xmx, убедиться что используется правильный путь в JAVA_HOME	17
4.4.5. Настройки логирования	18
4.5. BACKEND (BOISKUD-BKG.WAR)	18
4.5.1. Подготовка OS (Под пользователем root)	18
4.5.2. Установка BACKEND (boiskud-bkg.war)	19
4.5.3. Конфигурация BACKEND (boiskud-bkg.war)	19
4.5.4. Выставить по необходимости -Xms -Xmx, убедиться что используется правильный путь в JAVA_HOME	20

4.5.5.	Настройки логирования	21
5.	УСТАНОВКА И НАСТРОЙКА NGINX (РАЗДАЧА СТАТИКИ, БАЛАНСИРОВКА)	21
5.1.	Подготовка O.S.	21
5.2.	Разметка диска	22
5.3.	Установка NGINX	22
5.3.1.	Конфигурации NGINX	22
5.4.	Статика (BOISKUD-FRONT.TAR.GZ)	23

АННОТАЦИЯ

В настоящем документе представлено руководство системного администратора для эксплуатации Информационной системы «Биометрическая система контроля доступа».

Руководство системного администратора предназначено для работников, задействованных в процессе:

- установка, настройка и эксплуатация серверного и прикладного программного обеспечения.
- установка, настройка и эксплуатация средств защиты информации и средств криптографической защиты информации.

В рамках настоящего руководства приведено описание пошаговой установки и настройки серверного программного обеспечения Информационной системы «Биометрическая система контроля доступа» в Standalone конфигурации, описана и проиллюстрирована последовательность действий для выполнения вышеприведённых процессов.

Настоящее руководство состоит из следующих основных частей:

1. Установка и настройка PostgreSQL 10 (РСУБД).
2. Установка и настройка RabbitMQ (Сервер очередей).
3. Установка и настройка Java и приложений.
4. Установка и настройка NGINX (Балансировка компонентов, раздача статического контента).

ТЕРМИНЫ И ОПРЕДЕЛЕНИЯ

ТЕРМИН	ОПРЕДЕЛЕНИЕ
Авторизация	Предоставление определённому лицу или системе прав на выполнение определённых действий, а также процесс проверки (подтверждения) данных прав при попытке выполнения этих действий. Перед авторизацией выполняется аутентификация
Аутентификация	Процесс сопоставления предъявленных неизвестным лицом или системой сведений с имеющимися (полученными ранее из доверенного источника) сведениями зарегистрированных в ИС пользователей или систем с целью подтверждения факта принадлежности учётной записи лицу или системе и, как следствие, установления доверенных отношений в контексте сессии информационного взаимодействия. Аутентификация предшествует авторизации
Биометрический процессор	Обработчик запросов на выполнение биометрических операций.
ИС	Информационная Система
Микросервис	Комплекс программных компонент, реализованных на единой программной платформе и подлежащих повторному использованию
Системный пользователь	Лицо, имеющее системную учетную запись в системе Linux

1. ОБЩИЕ СВЕДЕНИЯ

1.1. НАЗНАЧЕНИЕ ДОКУМЕНТА

Документ «Руководство системного администратора» предназначен для системных инженеров и системных администраторов.

1.2. УРОВЕНЬ ПОДГОТОВКИ ПОЛЬЗОВАТЕЛЕЙ

Системный администратор должен иметь знания и опыт в области настройки и администрирования Microsoft Windows и Linux, настройки и администрирования СУБД PostgreSQL, создания/ хранения/ восстановления резервных копий данных Системы, установки обновлений и настройки операционной системы, установки оборудования и настройки рабочего места для снятия биометрии (веб-камера, микрофон).

2. УСТАНОВКА И НАСТРОЙКА POSTGRESQL 10

2.1. ПОДГОТОВКА СЕРВЕРОВ PGSQL

2.1.1. Разметка диска по стандарту

Необходимо выполнить разметку, форматирование диска и подключение к системе в следующем порядке:

1. Создать размеры и выполнить форматирование

```
pvccreate /dev/sdb ;
vgcreate vg_postgres /dev/sdb ;
lvcreate --name lv_u01 --size 5G vg_postgres '
lvcreate --name lv_main -l 100%FREE vg_postgres ;
mkfs.xfs /dev/mapper/vg_postgres-lv_u01 ;
mkfs.xfs /dev/mapper/vg_postgres-lv_main ;
```

1. После создания файловой системы в конце прописываем постоянные записи в /etc/fstab

/dev/vg_postgres/lv_u01	/u01	xfs	defaults	1 1
/dev/vg_postgres/lv_main	/u01/postgres/main	xfs	defaults	1 1

2. Создать точки монтирования. Монтирование необходимо осуществлять последовательно

```
mkdir /u01 && mount /u01; mkdir /u01/postgres/main -p && mount -a
```

3. Проверить, что доступное место корректно определилось в разметке командой

```
df -h
...
/dev/mapper/vg_postgres-lv_u01 5,0G 33M 5,0G 1% /u01
/dev/mapper/vg_postgres-lv_main 59G 33M 59G 1% /u01/postgres/main
```

2.1.2. Установка пакетов и настройка базы данных

1. Перед установкой необходимо подключить репозитории пакетов

```
zypper ar -f https://download.postgresql.org/pub/repos/zypp/repo/pgdg-sles-10.repo
```

2. Для обеспечения всех зависимостей необходимо включить в yast2

“Suse Linux Enterprise Software Development Kit 12 SP3 x86_64, SUSE Linux Enterprise High Availability Extension 12 SP3 x86_64, SUSE Package Hub 12 SP3 x86_64

3. Выполнить

```
SUSEConnect -p sle-sdk/12.3/x86_64 ;  
SUSEConnect -p PackageHub/12.3/x86_64 ;
```

4. Установить платные расширения

```
SUSEConnect -p sle-ha/12.3/x86_64 -r <регистрационный код> ;
```

2.2. УСТАНОВКА POSTGRESQL 10

Выполнить в консоли команду

```
zypper install postgresql10 postgresql10-contrib postgresql10-server postgresql10-devel
```

1. Перенести данные пользователя postgres в выделенную директорию

```
usermod -d /u01/postgres postgres  
cp -f /etc/skel/.bash* /u01/postgres/  
chown -R postgres: /u01/postgres
```

2. Для указания выделенного раздела при запуске для данных PGSQL следует создать файл /etc/systemd/system/postgresql-10.service.d/override.conf с содержанием

```
[Service]  
Environment=PGDATA=/u01/postgres/main/pgdata  
ExecStart=  
ExecStart=/usr/pgsql-10/bin/postmaster -D ${PGDATA}
```

Для этого выполнить

```
mkdir /etc/systemd/system/postgresql-10.service.d;  
cat <<EOF >/etc/systemd/system/postgresql-10.service.d/override.conf  
[Service]  
Environment=PGDATA=/u01/postgres/main/pgdata  
ExecStart=  
ExecStart=/usr/pgsql-10/bin/postmaster -D ${PGDATA}  
  
EOF
```

```
;
```

3. Перечитать конфигурацию systemd

```
systemctl daemon-reload
```

4. Переключиться в пользователя postgres

```
su - postgres;
```

ВНИМАНИЕ

Перед созданием БД необходимо привести ~/.bash_profile в такой вид

```
[ -f /etc/profile ] && source /etc/profile
PGDATA=/u01/postgres/main/pgdata
export PGDATA
export PATH=$PATH:/usr/pgsql-10/bin
# If you want to customize your settings,
# Use the file below. This is not overridden
# by the RPMS.
[ -f /var/lib/pgsql/.pgsql_profile ] && source /var/lib/pgsql/.pgsql_profile
```

5. Создать пустую базу данных

Перечитать окружение source ~/.bash_profile

Проверить echo \$PGDATA

```
/u01/postgres/main/pgdata
```

Выполнить

```
initdb --locale=ru_RU.utf8 --data-checksums
```

2.3. КОНФИГУРАЦИЯ БАЗЫ ДАННЫХ

1. В первую очередь необходимо прописать все сетевые разрешения для подключения в файле /u01/postgres/main/pgdata/pg_hba.conf
2. Подсеть 192.168.3.0/24 необходимо заменить в соответствии с реальной
3. Строка разрешения подключения для пользователя приложения BOISKUD и приложения luna

#	TYPE	DATABASE	USER	ADDRESS	METHOD
# IPv4 local connections:					
host	faceis_db	faceis	192.168.3.0/24	md5	
host	faceis_ui	faceis	192.168.3.0/24	md5	
host	admin_faceis_db	faceis	192.168.3.0/24	md5	
host	boiskud	boiskud	192.168.3.0/24	md5	

4. Следующие строки должны находиться в конце файла

```
# Replication:
# IPv4 local connections:
host all postgres 192.168.3.0/24 md5
# IPv4 local connections:
host postgres db_backup 192.168.3.0/24 md5
# IPv4 local connections:
host replication db_backup 192.168.3.0/24 trust
# IPv4 local connections:
host replication db_replication 192.168.3.0/24 trust
# IPv4 local connections:
host replication db_replication 192.168.3.0/24 trust
# IPv4 local connections:
host replication db_replication 192.168.3.0/24 trust
```

5. Файл конфигурации /u01/postgres/main/pgdata/postgresql.conf
следует привести к следующему виду

```
# Параметры PostgreSQL 10 (Соответствуют стандарту v1.5+)
# Распределение памяти для 8GB

# FILE LOCATIONS

data_directory = '/u01/postgres/main/pgdata'
# hba_file = '{{postgresql_hba_file}}'
# ident_file = '{{postgresql_ident_file}}'

# CONNECTIONS AND AUTHENTICATION

# - Connection Settings -

listen_addresses = '*'
port = 5432
max_connections = 500

# - Security and Authentication -

# - TCP Keepalives -

# RESOURCE USAGE (except WAL)

# - Memory -

shared_buffers = 2GB
work_mem = 16MB
maintenance_work_mem = 512MB
autovacuum_work_mem = 256MB

# - Kernel Resource Usage -

shared_preload_libraries = 'pg_stat_statements'
```

```
# WRITE AHEAD LOG

# - Settings -

wal_level = replica
fsync = on
synchronous_commit = on

# - Checkpoints -

checkpoint_timeout = 30min
max_wal_size = 4GB
min_wal_size = 1GB
checkpoint_completion_target = 0.9
checkpoint_warning = 60s

# - Archiving -

archive_mode = on
archive_command = '/bin/true'
archive_timeout = 600

# REPLICATION

# - Sending Server(s) -

max_wal_senders = 10
wal_keep_segments = 1000

# - Master Server -

# - Standby Servers -

hot_standby = on

# QUERY TUNING

# - Planner Method Configuration -

# - Planner Cost Constants -

effective_cache_size = 3GB

# ERROR REPORTING AND LOGGING

# - Where to Log -

log_destination = 'stderr'
log_directory = 'pg_log'
logging_collector = on
log_directory = '/var/log/postgres'
```

```
log_filename = 'postgresql-%Y-%m-%d_%H%M%S.log'
log_file_mode = '0600'
log_truncate_on_rotation = off
log_rotation_age = 1d
log_rotation_size = 0
```

```
# - When to Log -
```

```
log_min_duration_statement = '250'
```

```
# - What to Log -
```

```
log_checkpoints = on
log_connections = on
log_disconnections = on
```

```
log_line_prefix = '%t [%p] %q%h %a %u@%d '
```

```
log_lock_waits = on
log_statement = 'ddl'
log_replication_commands = on
log_temp_files = 0
```

```
# RUNTIME STATISTICS
```

```
# - Query/Index Statistics Collector -
```

```
track_io_timing = on
```

```
# AUTOVACUUM PARAMETERS
```

```
autovacuum = on
log_autovacuum_min_duration = 60000
autovacuum_max_workers = 3
autovacuum_vacuum_scale_factor = 0.01
autovacuum_analyze_scale_factor = 0.05
```

```
# CLIENT CONNECTION DEFAULTS
```

```
# - Locale and Formatting -
```

```
lc_messages = 'en_US.UTF-8'
```

```
# CUSTOMIZED OPTIONS
```

```
# Add settings for extensions here
```

```
# - pg_stat_statements -
```

```
pg_stat_statements.track = 'top'
pg_stat_statements.max = '10000'
```

```
pg_stat_statements.track_utility = 'off'
```

2.4. ПЕРЕЗАПУСК СЕРВИСА ДЛЯ ПРИМЕНЕНИЯ КОНФИГУРАЦИЙ

Перезагрузка сервиса для применения конфигураций осуществляется командой

```
systemctl restart postgresql-10;
```

либо, выполнив в консоли под пользователем postgres, команду

```
psql -c "SELECT pg_reload_conf();"
```

2.5. СКРИПТ ДЛЯ СОЗДАНИЯ СИСТЕМНЫХ И ПРИКЛАДНЫХ ПОЛЬЗОВАТЕЛЕЙ

Подключаемся к БД локально

Выполнить в консоли команду

```
psql
```

```
BEGIN;
```

```
-- Создание стандартных ролей
```

```
-- Роль для подключения к БД персонализированных пользователей
```

```
CREATE ROLE user_connect;
```

```
COMMENT ON ROLE user_connect
```

```
IS 'Разрешение на подключение к БД с адресов 0.0.0.0/0';
```

```
-- Роль для выдачи разрешений системам мониторинга
```

```
CREATE ROLE monitoring_role;
```

```
COMMENT ON ROLE monitoring_role
```

```
IS 'Разрешения для систем мониторинга';
```

```
-- Пользователь для резервного копирования
```

```
CREATE ROLE db_backup LOGIN
```

```
ENCRYPTED PASSWORD 'md5c940ad9213634f663c1dc725367f5150' REPLICATION;
```

```
COMMENT ON ROLE db_backup IS 'Резервное копирование';
```

```
-- Пользователь для репликации
```

```
CREATE ROLE db_replication LOGIN
```

```
ENCRYPTED PASSWORD 'md5f7cdb91bd94492806374fbd6c539d9d7' REPLICATION;
```

```
COMMENT ON ROLE db_replication IS 'Репликация';
```

```
-- Grant on database
```

```
REVOKE ALL ON DATABASE postgres FROM public;
```

```
GRANT CONNECT,TEMPORARY ON DATABASE postgres TO monitoring_role;
```

```
GRANT CONNECT,TEMPORARY ON DATABASE postgres TO db_backup;
```

```
-- Создание прикладных ролей
```

```
-- Пользователь Luna
```

```
create role faceis LOGIN PASSWORD 'faceis';
```

```
COMMENT ON ROLE faceis IS 'Пользователь Luna';
```

```
-- Пользователь boiskud
create role boiskud LOGIN PASSWORD 'boiskud';
COMMENT ON ROLE boiskud IS 'Пользователь BOISKUD';

COMMIT;
```

3. УСТАНОВКА И НАСТРОЙКА RABBITMQ

Сервера очередей для обработки очередей заданий компонентов.

3.1. ПОДГОТОВКА СЕРВЕРОВ

1. Предварительно необходимо подключить репозиторий erlang

```
zypper ar -f
http://download.opensuse.org/repositories/devel:/languages:/erlang/SLE_12/devel:languages:erlang.repo
```

2. Установить зависимости

```
zypper install erlang-20.3.8.6-12.1.x86_64
```

3.2. РАЗМЕТКА ДИСКА

```
pvccreate /dev/sdb;
vgcreate vg_data /dev/sdb;
lvcreate --name lv_log --size 10G vg_data;
lvcreate --name lv_rabbit -l100%FREE vg_data;

mkfs.xfs /dev/vg_data/lv_log;
mkfs.xfs /dev/vg_data/lv_rabbitmq;
```

1. Создать соответствующие директории командами

```
mkdir /var/lib/rabbitmq
mkdir /var/log/rabbitmq
```

2. Прописать записи для постоянного монтирования, добавив в конец файла /etc/fstab строки

/dev/vg_data/lv_rabbitmq	/var/lib/rabbitmq	xfs	defaults	1 1
/dev/vg_data/lv_log	/var/log/rabbitmq	xfs	defaults	1 1

3. Смонтировать разделы командой

```
mount -a
```

3.3. УСТАНОВКА RABBITMQ

Установить ПО rabbitmq, выполнив в консоли

```
zypper install https://github.com/rabbitmq/rabbitmq-server/releases/download/v3.7.9/rabbitmq-server-3.7.9-1.suse.noarch.rpm
```

1. Скачать плагин и распаковать в папку rabbitmq командами

```
wget https://dl.bintray.com/rabbitmq/community-plugins/3.7.x/rabbitmq\_delayed\_message\_exchange/rabbitmq\_delayed\_message\_exchange-20171201-3.7.x.zip
```

```
unzip rabbitmq_delayed_message_exchange-20171201-3.7.x.zip -d /usr/lib/rabbitmq/lib/rabbitmq_server-3.7.*/plugins
```

2. Подключить плагины в брокере, запустить и включить автозагрузку

```
rabbitmq-plugins enable rabbitmq_management ;  
rabbitmq-plugins enable rabbitmq_delayed_message_exchange ;  
systemctl restart rabbitmq-server ;  
systemctl enable rabbitmq-server ;
```

4. УСТАНОВКА И НАСТРОЙКА JAVA STACK

Дистрибутивы для установки доступны:

- <https://portal.rtlabs.ru/~cI8Lr> (JDK1.8)
- <https://portal.rtlabs.ru/~Dw1gO> (linux-amd64.tgz)
- <https://portal.rtlabs.ru/~a82gj1> (bioskud-web.tar.gz)
- <https://portal.rtlabs.ru/~SsTR3> (bioskud-bkg.tar.gz)

На серверах d00boiskud-app необходимо выполнить следующие действия:

- Подготовить OS
- Разметить диск
- Установить JDK1.8.141
- Скопировать systemd скрипт управления приложением
- скофигурировать application.properties
- Сконфигурировать setenv.sh

4.1. РАЗМЕТКА ДИСКА

1. Убедимся, что дополнительный диск присутствует

```
fdisk -l | grep sdb  
Disk /dev/sdb:
```

2. Выполнить

```
pvccreate /dev/sdb ;  
vgcreate vg_data /dev/sdb ;  
lvcreate --name lv_data --size 10G vg_data ;  
lvcreate --name lv_log -l 100%FREE vg_data ;  
mkfs.xfs /dev/mapper/vg_data-lv_data ;  
mkfs.xfs /dev/mapper/vg_data-lv_log ;
```

3. После создания файловой системы в конце прописываем постоянные записи в /etc/fstab

/dev/vg_data/lv_boiskud	/boiskud	xfs	defaults	1 1
/dev/vg_data/lv_log	/var/log/boiskud	xfs	defaults	1 1

4. Создать точки монтирования и монтируем

```
mkdir /boiskud -p && mount /boiskud  
mkdir /var/log/boiskud -p && mount -a
```

4.2. СОЗДАНИЕ СИСТЕМНОГО ПОЛЬЗОВАТЕЛЯ BOISKUD

1. Создать группу пользователей в системе

```
groupadd app_grp;
```

2. Создать пользователя boiskud с группой app_grp и домашней директорией /boiskud
3. Выдать права на директорию с логами

```
useradd -d /boiskud -g app_grp boiskud && chown -R boiskud: /boiskud /var/log/boiskud;
```

4.3. JAVA (JDK1.8.181)

1. Создать директорию для установки

```
mkdir /usr/java
```

2. Положить установочный пакет в /usr/java/jdk-8u181-linux-x64.rpm и выполним стандартную установку

```
zypper install /usr/java/jdk-8u181-linux-x64.rpm ;
```

3. По ходу установки подтвердить все положительно

```
jdk1.8-2000:1.8.0_181-fcs.x86_64 (Plain RPM files cache): Signature verification failed [4-Signatures public  
key is not available]  
Abort, retry, ignore? [a/r/i] (a): i
```

Результат: Java установлена

4.4. FRONTEND (BOISKUD-WEB.JAR)

Дистрибутив boiskud-web.tar.gz необходимо скопировать в директорию /boiskud на сервере d00boiskud-app01 и распаковать под пользователем boiskud

```
sudo -iu boiskud;  
tar xzvf boiskud-web.tar.gz;
```

4.4.1. Подготовка OS (Под пользователем root)

1. Установить лимиты для группы пользователя. Для этого вставить следующее содержимое в конец файла /etc/security/limits.conf

```
...
```

```
# End of file
@app_grp soft core unlimited
@app_grp soft nproc 147456
@app_grp - nofile 65535
```

2. Разрешить управлением сервиса под sudo для системного пользователя. Для этого выполнить следующую команду

```
cat <<EOF>/etc/sudoers.d/app_grp
boiskud ALL=(ALL) NOPASSWD:SETENV: /bin/systemctl status boiskud-web, /bin/systemctl
reload boiskud-web, /bin/systemctl restart boiskud-web, /bin/systemctl stop boiskud-web ,
/bin/systemctl start boiskud-web
boiskud ALL=(ALL) NOPASSWD:SETENV: /bin/systemctl status boiskud-bkg, /bin/systemctl
reload boiskud-bkg, /bin/systemctl restart boiskud-bkg, /bin/systemctl stop boiskud-bkg ,
/bin/systemctl start boiskud-bkg
EOF
```

4.4.2. Установка FRONTEND (BOISKUD-WEB.WAR)

1. Под пользователе root установить systemd скрипт запуска\останова сервиса

```
cp /boiskud/boiskud-web/systemd/boiskud-web.service /etc/systemd/system/;
```

2. При необходимости выставить параметры JVM -xms, -mx в файле /boiskud/boiskud-web/bin/setenv.sh

4.4.3. Конфигурация FRONTEND (BOISKUD-WEB.WAR)

Следующие параметры необходимо настроить в соответствии с реальными

- server.servlet.context-path (Рабочий контекст приложения boiskud-web)
- boiskud.host.name (Внешний адрес системы BOISKUD)
- boiskud.context.path (Внешний URL контекст системы BOISKUD)
- rabbitmq.addresses (Адрес:порт, серверов очередей RabbitMQ)
- rabbitmq.username, rabbitmq.password (Доступы к серверам очередей RabbitMQ)
- spring.datasource.url (Адреса серверов баз данных Postgres)
- spring.datasource.username. spring.datasource.password (Доступы к серверам баз данных Postgres)
- spring.datasource.hikari.maximumPoolSize (Максимальное количество возможных соединений к базе данных)

/boiskud/boiskud-web/application.properties

```
# SITE CONFIG
# boiskud.host.name=https://dev-ekassir-ap8.espp.ru #
#Название системы
system_name=boiskud-web
# LISTEN
server.port=8080
server.servlet.context-path=/boiskud
boiskud.host.name=https://biometrics.rtlabs.ru
boiskud.context.path=/boiskud

# Сервера очередей Rabbit
# Sample: rabbitmq.addresses=t00boiskud-cmq01:5672,t00boiskud-cmq02:5672
rabbitmq.addresses=u00boiskud-db01:5672,u00boiskud-sign:5672
rabbitmq.username=boiskud
rabbitmq.password=boiskud

# Сервера базы данных Postgres
# Sample: spring.datasource.url=jdbc:postgresql://t00boiskud-db01:6432,t00boiskud-
db02:6432/boiskud?targetServerType=master
spring.datasource.url=jdbc:postgresql://u00boiskud-db01:6432,u00boiskud-
db02:6432/boiskud?targetServerType=master
spring.datasource.username=boiskud
spring.datasource.password=boiskud
spring.datasource.hikari.maximumPoolSize=40
spring.datasource.hikari.connectionTimeout=60000
spring.datasource.hikari.minIdle=5
spring.datasource.hikari.idle-timeout=10000
```

4.4.4. Выставить по необходимости -Xms -Xmx, убедиться что используется правильный путь в JAVA_HOME

/boiskud/boiskud-web/bin/setenv.sh

```
JAVA_HOME=/usr/java/default
export JAVA_HOME

# CONFIG="-Dspring.config.location=$SERVICE_PATH/application.properties"
CATALINA_OPTS="-Djava.net.preferIPv4Stack=true\
-Dfile.encoding=utf-8\
-Dspring.profiles.active=rabbitmq,smev\
-Dlogging.config=/boiskud/boiskud-web/logging/logback.xml\
-Dspring.config.additional-location=file:/boiskud/boiskud-web/application.properties"
export CATALINA_OPTS

JAVA_OPTS="-server\
-Xms512M\
-Xmx1G\
-XX:MaxMetaspaceSize=256M\
-XX:+DoEscapeAnalysis\
-XX:+UseG1GC\
-XX:+ExplicitGCInvokesConcurrent\
```

```
-XX:MaxGCPauseMillis=500\  
-XX:CompressedClassSpaceSize=1024M\  
-XX:+UseCompressedOops\  
-XX:+UseCompressedClassPointers\  
-verbose:gc\  
-XX:+PrintGCDetails\  
-XX:+PrintGCDateStamps\  
-XX:+PrintGCApplicationStoppedTime\  
-XX:+UseGCLogFileRotation\  
-XX:NumberOfGCLogFiles=10\  
-XX:GCLogFileSize=20M\  
-Xloggc:/var/log/boiskud/boiskud-web/gc.log.`date +%Y%m%d`\  
-XX:+HeapDumpOnOutOfMemoryError\  
-XX:HeapDumpPath=/tmp/  
export JAVA_OPTS
```

4.4.5. Настройки логирования

1. Сконфигурировать параметры логирования в файле
/boiskud/boiskud-web/logging.properties

- уровень логирования задаётся параметром в файле logging.properties (logging.logback.level=INFO\ERROR\DEBUG\TRACE)
- можно изменить директорию для логов в параметре: logging.path=
- qradar.logserver = localhost > qradar.logserver = ip_address_qradar
- syslog.logserver = localhost > syslog.logserver = ip_address_syslogserver

2. Выполнить перезагрузку systemd, и включить автозапуск приложений

```
systemctl daemon-reload  
systemctl enable boiskud-web.service  
systemctl restart boiskud-web.service
```

4.5. BACKEND (BOISKUD-BKG.WAR)

Дистрибутив boiskud-bkg.tar.gz скопировать в директорию /boiskud на сервере d00boiskud-app01 и распаковать под системным пользователем Linux boiskud

```
sudo -iu boiskud;  
tar xzvf boiskud-bkg.tar.gz;
```

4.5.1. Подготовка OS (Под пользователем root)

1. Установить лимиты для группы пользователя. Для этого вставить следующее содержимое в конец файла /etc/security/limits.conf

```
...  
# End of file  
@app_grp soft core unlimited
```

```
@app_grp soft nproc 147456
```

```
@app_grp - nofile 65535
```

2. Разрешить управление сервиса под sudo для системного пользователя. Для этого выполнить следующую команду

```
cat <<EOF>/etc/sudoers.d/app_grp
```

```
boiskud ALL=(ALL) NOPASSWD:SETENV: /bin/systemctl status boiskud-web, /bin/systemctl  
reload boiskud-web, /bin/systemctl restart boiskud-web, /bin/systemctl stop boiskud-web ,  
/bin/systemctl start boiskud-web
```

```
boiskud ALL=(ALL) NOPASSWD:SETENV: /bin/systemctl status boiskud-bkg, /bin/systemctl  
reload boiskud-bkg, /bin/systemctl restart boiskud-bkg, /bin/systemctl stop boiskud-bkg ,  
/bin/systemctl start boiskud-bkg
```

```
EOF
```

4.5.2. Установка BACKEND (boiskud-bkg.war)

Под пользователем root установить systemd скрипт
запуска/останова сервиса

```
cp /boiskud/boiskud-web/systemd/boiskud-web.service /etc/systemd/system/;
```

Под пользователем root установить systemd скрипт
запуска/останова сервиса

```
cp /boiskud/boiskud-web/systemd/boiskud-bkg.service /etc/systemd/system/;
```

4.5.3. Конфигурация BACKEND (boiskud-bkg.war)

Следующие параметры необходимо настроить в соответствии с
реальными

- ebs.reqbio.esiaHashPdn.url (URL ЕБС для запроса БКИШ персоны по хешу ПДн ЕСИА)
- ebs.reqbio.esiaOid.url (URL ЕБС для запроса БКИШ персоны по OID ЕСИА)
- ebs.reqbio.project.name (Код проекта ЕБС)
- ebs.reqbio.client.certAlias (Алиас сертификата БиоСКУД для аутентификации в ЕБС)
- boiskud.truststore.path (Путь к truststore с сертификатом БиоСКУД для аутентификации в ЕБС)
- boiskud.truststore.pwd (Пароль к truststore с сертификатом БиоСКУД для аутентификации в ЕБС)
- spring.datasource.url (Адреса серверов баз данных Postgres)
- spring.datasource.username. spring.datasource.password (Доступы к серверам баз данных Postgres)
- spring.datasource.hikari.maximumPoolSize (Максимальное количество возможных соединений к базе данных)

```
/boiskud/boiskud-bkg/application.properties
```

```

#Название системы
system_name=boiskud-bkg
#LISTEN
server.port=8099
server.servlet.context-path=/boiskud

# Sample: rabbitmq.addresses=t00boiskud-cmq01:5672,t00boiskud-cmq02:5672
rabbitmq.addresses=u00boiskud-db01:5672,u00boiskud-sign:5672
rabbitmq.username=boiskud
rabbitmq.password=boiskud

#DB
# Sample: spring.datasource.url=jdbc:postgresql://t00boiskud-db01:6432,t00boiskud-
db02:6432/boiskud?targetServerType=master
spring.datasource.url=jdbc:postgresql://u00boiskud-db01:6432,u00boiskud-
db02:6432/boiskud?targetServerType=master
spring.datasource.username=boiskud
spring.datasource.password=boiskud
spring.datasource.hikari.maximumPoolSize=40
spring.datasource.hikari.connectionTimeout=60000
spring.datasource.hikari.minIdle=5
spring.datasource.hikari.idle-timeout=10000

```

4.5.4. Выставить по необходимости -Xms -Xmx, убедиться что используется правильный путь в JAVA_HOME

/boiskud/boiskud-bkg/bin/setenv.sh

```

JAVA_HOME=/usr/java/default
export JAVA_HOME

CATALINA_OPTS="-Djava.net.preferIPv4Stack=true\
-Dfile.encoding=utf-8\
-Dspring.profiles.active=rabbitmq,smev\
-Dlogging.config=/boiskud/boiskud-bkg/logging/logback.xml\
-Dspring.config.additional-location=file:/boiskud/boiskud-bkg/application.properties"
export CATALINA_OPTS

JAVA_OPTS="-server\
-Xms512M\
-Xmx1G\
-XX:MaxMetaspaceSize=256M\
-XX:+DoEscapeAnalysis\
-XX:+UseG1GC\
-XX:+ExplicitGCInvokesConcurrent\
-XX:MaxGCPauseMillis=500\
-XX:CompressedClassSpaceSize=1024M\
-XX:+UseCompressedOops\
-XX:+UseCompressedClassPointers\
-verbose:gc\
-XX:+PrintGCDetails\
-XX:+PrintGCDateStamps\

```

```
-XX:+PrintGCApplicationStoppedTime\  
-XX:+UseGCLogFileRotation\  
-XX:NumberOfGCLogFiles=10\  
-XX:GCLogFileSize=20M\  
-Xloggc:/var/log/boiskud/boiskud-bkg/gc.log.`date +%Y%m%d`\  
-XX:+HeapDumpOnOutOfMemoryError\  
-XX:HeapDumpPath=/tmp/"  
export JAVA_OPTS
```

4.5.5. Настройки логирования

Сконфигурировать параметры логирования в файле
/boiskud/boiskud-bkg/logging.properties

- уровень логирования задать параметром
(logging.logback.loglevel=INFO\ERROR\DEBUG\TRACE)
- можно изменить директорию для логов в параметре: logging.path=
- syslog.logserver = localhost > syslog.logserver = ip_address_syslogserver

Выполнить перезагрузку systemd, и включить автозапуск
приложений

```
systemctl daemon-reload  
systemctl enable boiskud-bkg.service  
systemctl restart boiskud-bkg.service
```

5. УСТАНОВКА И НАСТРОЙКА NGINX (РАЗДАЧА СТАТИКИ, БАЛАНСИРОВКА)

5.1. ПОДГОТОВКА O.S.

Дистрибутив с конфигурациями доступен по ссылке: <https://portal.rtlabs.ru/~aY4A1>
Предварительно необходимо подключить официальный репозиторий nginx

```
zypper addrepo -G -t yum -c 'http://nginx.org/packages/sles/12' nginx;  
wget http://nginx.org/keys/nginx\_signing.key ;  
rpm --import nginx_signing.key ;
```

Перенести дистрибутив boiskud-front.tar.gz на сервер в удобную директорию
Распаковать tar xzvf boiskud-front.tar.gz
Для конфигурации параметров ядра необходимо

Скопировать файл

```
boiskud-front/sysctl.d/99-sysctl.conf  
в  
/etc/sysctl.d/99-sysctl.conf
```

Применить изменения

```
sysctl -p
```

5.2. РАЗМЕТКА ДИСКА

1. Убедимся, что дополнительный диск присутствует

```
fdisk -l | grep sdb  
Disk /dev/sdb:  
Выполняем:  
pvcreate /dev/sdb ;  
vgcreate vg_data /dev/sdb ;  
lvcreate --name lv_log -l 100%FREE vg_data ;  
mkfs.xfs /dev/mapper/vg_data-lv_log ;
```

2. После создания файловой системы в конце прописываем постоянные записи в /etc/fstab

```
/dev/vg_data/lv_log /var/log/nginx xfs defaults 1 1
```

Далее, создать точки монтирования и монтируем

```
mkdir /var/log/nginx -p && mount -a
```

5.3. УСТАНОВКА NGINX

1. На сервере выполнить

```
zypper install nginx
```

2. Проверим версию

```
nginx -v  
nginx version: nginx/1.14.0
```

5.3.1. Конфигурации NGINX

Файлы необходимо скопировать

```
boiskud-front/nginx/nginx.conf  
в  
/etc/nginx/nginx.conf на сервере  
(Подтвердить с заменой)
```

```
worker_processes <Должно равняться количеству ЦПУ на сервере>
```

При использовании HTTPS необходимо положить *.key, *.pem в директорию на сервере /etc/nginx/ssl

Скопировать конфигурационный файл

```
boiskud-front/nginx/conf.d/boiskud_https.conf  
в  
/etc/nginx/conf.d/boiskud_https.conf
```

В блоке upstream boiskud-web значение server_web_fqdn > изменить на реальный сервер сервиса boiskud-web.war.

Продублировать строчку и заменить server_fqdn на второй сервер в балансировке.

В блоке upstream boiskud-bkg значение server_bkg_fqdn > изменить на реальный сервер сервиса boiskud-bkg.war.

Продублировать строчку и заменить server_fqdn на второй сервер в балансировке.

- site_name > Внешний DNS системы
- ssl_key > Имя файла *.key в /etc/nginx/ssl/*.key
- ssl_cert > Имя открытой части сертификата в /etc/nginx/ssl/*.{pem\cert}

При использовании HTTP необходимо использовать boiskud_http.conf

Скопировать nginx/conf.d/boiskud_http.conf в /etc/nginx/conf.d/boiskud_http.conf

В блоке upstream boiskud-web значение server_web_fqdn > изменить на реальный сервер сервиса boiskud-web.war

Продублировать строчку и заменить server_fqdn на второй сервер в балансировке.

В блоке upstream boiskud-bkg значение server_bkg_fqdn > изменить на реальный сервер сервиса boiskud-bkg.war

Продублировать строчку и заменить server_fqdn на второй сервер в балансировке

site_name > Внешний DNS системы

5.4. СТАТИКА (BOISKUD-FRONT.TAR.GZ)

1. Дистрибутив со статикой boiskud-front/boiskud-front.tar.gz необходимо перенести в папку на сервере /usr/share/nginx/boiskud-front.tar.gz и разархивировать

```
cd /usr/share/nginx ;  
tar xzvf boiskud-front.tar.gz
```

2. Скорректировать пользователя и права на папку

```
chown -R nginx: /usr/share/nginx ;  
chmod -R 0755 /usr/share/nginx ;
```

3. Запустим nginx и добавим в автозагрузку сервис

```
systemctl enable nginx ;  
systemctl restart nginx ;
```

