s8.capital
Акционерное общество
«С8 Капитал»
Волгоградский пр-т, д. 43, корп.3,
Москва, 109316
Тел.: +7 (495) 204-88-88
Факс: +7 (495) 204-88-87

# Программное обеспечение «Программный модуль игровой среды проведения розыгрыша»

## Описание функциональных характеристик программного обеспечения

## Оглавление

Акционерное общество
«С8 Капитал»
Волгоградский пр-т, д. 43, корп.3,
Москва, 109316
Тел.: +7 (495) 204-88-88
Факс: +7 (495) 204-88-87

## Введение

Документ содержит описание основных функциональных характеристик программного обеспечения (далее-ПО) «Программный модуль игровой среды проведения розыгрыша».

## Термины и определения

| Термин | Определение |
|---|---|
| СУБД | Систем управления базами данных |
| ОС | Операционная система |
| ВМ | Виртуальная машина - программная и/или аппаратная система, имитирующая аппаратное обеспечение некоторой платформы и позволяющая исполнять программы предназначенные для данной платформы |
| ПО | Программное обеспечение |
| JVM | Java Virtual Machine - виртуальная машина Java - основная часть исполняющей системы JAVA |

# 1. Назначение документа

В настоящем документе содержится инструкция по настройке сред Программного модуля игровой среды проведения розыгрыша версии 2.

Цель документа — обеспечить возможность создания и настройки продуктивной и тестовой сред и компонентов системы, для корректного её функционирования.

# 2. Назначение и функции программного обеспечения «Программный модуль игровой среды проведения розыгрыша»

Программный модуль игровой среды проведения розыгрыша предназначен для автоматизации управления пулом лототронов, имеющих встроенный компьютер и соответствующих требованиям, которые могут быть уточнены на этапе проектирования Системы, в частности:

- наличие встроенного компьютера, позволяющего принимать и исполнять команды от внешней системы
- возможность подключения к сети Ethernet, обмена данными в формате ASCII по протоколу TCP/IP
- возможность поддерживать одновременно более одного вида вида соединения (например, контролирующее воздействие и получение данных о состоянии лототрона)
- возможность поддержания непрерывного соединения (без необходимости прерывать связь между командами)
- возможность контролировать состояние лототорона внешней Системой вне проведения розыгрыша

Программный модуль игровой среды проведения розыгрыша выполняет следующие функции:

- автоматизация управления пулом лототронов в части проведения проверки работоспособности каждого лототрона и комплектности набора шаров; проведения розыгрышей (старт игры, обеспечение возможности визуального контроля корректности считывания RFID-метки лототроном посредством передачи изображения на монитор для оператора)
- обеспечение возможности настраивать параметры игр, которым должны соответствовать розыгрыши при их проведении в автоматическом режиме с применением лототронов
- хранение данных проведенных розыгрышей, в т.ч. ФИО ответственных лиц, данные выигрышных комбинаций, параметры проведенных розыгрышей; данные проведенных проверок работоспособности лототронов

# 3. Дистрибутивы

## 3.1 Виртуальные машины

Согласно Архитектуре все основные компоненты Системы, за исключением интеллектуального балансировщика нагрузки , размещаются в виртуальный машинах VMware vSphere.

### 3.2 Операционная система

Все виртуальные машины Системы работают под управлением ОС CentOS 7 64 bit.

### 3.3 СУБД

Для хранения данных и используется СУБД MariaDB версии 10.4

### 3.4 Java

Для запуска прикладного ПО требуется JVM версии 11-openjdk из репозиториев Centos.

## 4. Настройка кластера СУБД.

- Создать 3 ВМ для каждого узла кластера.
- Установить ОС на каждый узел.

После установки ОС на каждом узле кластера выполнить [*] ниже перечисленные пункты.

### 4.1 Настройка ОС

- Отредактировать файл /etc/hosts.

Добавить строки:
*<адрес_узла_1> <имя_узла_1.домен> <имя_узла_1>*
*<адрес_узла_2> <имя_узла_2.домен > <имя_узла_2>*
*<адрес_узла_3> <имя_узла_3.домен > <имя_узла_3>*

- При необходимости отключить SELinux.

Отредактировать /etc/selinux.config:
*SELINUX=disabled*
или настроить:
*semanage port -a -t mysqld_port_t -p udp 4567*
*semanage port -a -t mysqld_port_t -p tcp 4567*
*semanage port -a -t mysqld_port_t -p tcp 4568*
*semanage port -a -t mysqld_port_t -p tcp 4444*
*semanage permissive -a mysqld_t*

- Настроить firewall

*firewall-cmd --add-port=4567/tcp --permanent*
*firewall-cmd --add-port=4568/tcp --permanent*
*firewall-cmd --add-port=4444/tcp --permanent*
*firewall-cmd --add-port=3306/tcp --permanent*
*firewall-cmd --reload*

### 4.2 Установка MariaDB

- Добавить репозиторий MariaDB.

В папке / etc/yum.repo.d/ создать файл MariaDB_10_1.repo с содержанием
*[mariadb]*
*name = MariaDB*
*baseurl = http://yum.mariadb.org/10.4/centos7-amd64*
*gpgkey=https://yum.mariadb.org/RPM-GPG-KEY-MariaDB*
*gpgcheck=1*

- Обновить список репозиториев:

*yum update*

- Установить MariaDB и требующиеся зависимости:

*yum install net-tools MariaDB-server MariaDB-client rsync galera-4*

### 4.3 Настройка MariaDB

- Запустить СУБД

*systemctl start mariadb.service*

- Выполнить первоначальную настройку

*mysql_secure_installation*
*установить пароль для пользователя root*

- Добавить пользователя репликации

*mysql -p*
ввести пароль пользователя root
Выполнить следующие команды:
*grant usage on *.* to repl_user@'%' identified by '< пароль >';*
*grant all privileges on *.* to repl_user@'%';*
*flush privileges;*
*exit;*

- Остановить СУБД

*systemctl stop mariadb*

- Настроить кластер

Отредактировать файл /etc/my.cnf.d/server.cnf
*[mysqld]*
*bind-address=0.0.0.0*
*collation-server = utf8_general_ci*
*init-connect = 'SET NAMES utf8'*
*character-set-server = utf8*
*[galera]*
*wsrep_on=ON*
*binlog_format=ROW*
*innodb_autoinc_lock_mode=2*
*innodb_locks_unsafe_for_binlog=1*
*default_storage_engine=InnoDB*
*innodb_log_file_size=100M*
*innodb_file_per_table=1*
*innodb_flush_log_at_trx_commit=1*
*wsrep_provider=/usr/lib64/galera-4/libgalera_smm.so*
*wsrep_cluster_address=”gcomm://< имя _ узла _1>,< имя _ узла _2>,< имя _ узла _3>”*
*wsrep_cluster_name=’galera_cluster’*
*wsrep_node_address=’< адрес _ узла для которого производится настройка >’*
*wsrep_node_name=’<имя_узла для которого производится настройка>’*
*wsrep_sst_method=rsync*
*wsrep_sst_auth=repl_user:< пароль >*

- Запустить кластер

Для первого узла выполнить
*/usr/bin/galera_new_cluster*
Для остальных узлов
*systemctl start mariadb.service*

Проверить состояние кластера можно следующей командой
*show status where Variable_name in ('wsrep_incoming_addresses', 'wsrep_cluster_size');*

- Добавить службу СУБД в автозагрузку, при необходимости

*systemctl enable mariadb.service*

## 5. Настройка кластера приложения

- Создать 3 ВМ для каждого узла кластера.
- Установить ОС на каждый узел.

После установки ОС на каждом узле кластера выполнить [*] ниже перечисленные пункты.

### 5.1 Настройка ОС

- Отредактировать файл /etc/hosts.

Добавить строки:
*<адрес_узла_1> <имя_узла_1.домен> <имя_узла_1>*
*<адрес_узла_2> <имя_узла_2.домен > <имя_узла_2>*
*<адрес_узла_3> <имя_узла_3.домен > <имя_узла_3>*

- При необходимости отключить SELinux.

Отредактировать /etc/selinux.config:
*SELINUX=disabled*
или настроить:
*semanage port -a -t lms_port_t -p tcp 8080*
*semanage port -a -t lms_port_t -p tcp 5701*
*semanage port -a -t lms_port_t -p tcp 54327*
*semanage permissive -a lms_t*

- Настроить firewall

*firewall-cmd --add-port=8080/tcp --permanent*
*firewall-cmd --add-port=5701/tcp --permanent*
*firewall-cmd --add-port=54327/tcp --permanent*
*firewall-cmd --reload*

- Установить Java

*yum install java-11-openjdk*

### 5.2 Установка приложения

- Распаковать архив в папку */opt*

*tar -xvf lms.tar.gz -C /opt*
Структура папок приложения должна выглядеть следующим образом:
*lms*
*|--- bin*
*| |--- lms.war*
*|--- config*
*| |--- applecation.yml*

*|--- pics*
*|  |--- balls*
*|  |--- logos*
*|--- lms.sh*

- Создать системную службу

*mv /opt/lms/scripts/lms.service /usr/lib/systemd/system*
*systemct daemon-relod*

### 5.3 Настройка приложения

- Изменить настройки приложения в файле */opt/lms/config/application.yml*

- Выполнить сценарии для создания БД и миграции данных (если необходимо перенести данные из предыдущей версии), предварительно изменив пользователя/пароль для подключения к БД

*sh /opt/lms/scripts/db/ create_db.sh*
*sh /opt/lms/scripts/db/migrate_data.sh*

- Запустить службу

*systemctl start lms.service*

- Добавить службу в автозагрузку

*systemctl enable lms.service*

## 6. Настройка балансировщика нагрузки на примере http-сервера nginx

- На каждый узел кластера балансировщика установить ОС.

После установки ОС на каждом узле кластера выполнить [*] ниже перечисленные пункты.

### 6.1 Настройка ОС

- Настроить firewall

*firewall-cmd --add-port=80/tcp --permanent*
*firewall-cmd –reload*

- Настроить SELinux

*setsebool -P httpd_can_network_connect 1*

### 6.2 Установка nginx

- Установить nginx

*yum install nginx*

### 6.3 Настройка nginx

- Отредактировать конфигурационные файлы

В файле /etc/nginx/nginx.conf внутрь секции http {} добавить следующее содержимое:

```
map $http_upgrade $connection_upgrade {
 default upgrade;
 '' close;

}

upstream lms {
 server <application-node1-ip>:8080 max_fails=2 fail_timeout=2s;
 server <application-node2-ip>:8080 max_fails=2 fail_timeout=2s;
 server <application-node3-ip>:8080 max_fails=2 fail_timeout=2s;
}

server {
 listen 80;
 location / {
  proxy_pass http://lms;
  proxy_http_version 1.1;
  proxy_set_header Upgrade $http_upgrade;
  proxy_set_header Connection $connection_upgrade;
  proxy_set_header Host $host;
  proxy_set_header X-Forwarded-Proto $scheme;
  proxy_set_header X-Real-IP $remote_addr;
  proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
 }
}
```

- Запустить nginx

*systemctl start nginx*

- Добавить службу nginx в автозагрузку

*systemctl enable nginx*

# 7. Настройка сервера для трансляции видео-потока

## 7.1 Установка rtps-ws-proxy

- Загрузить архив rtsp-ws-proxy.tar.gz

*wget -O rtsp-ws-proxy.tar.gz https://owncloud.otr.ru/index.php/s/ihegzSo6tOIMcXg /download --no-check-certificate*

- Распаковать архив rtsp-ws-proxy.tar.gz

*tar -xvf rtsp-ws-proxy.tar.gz*

- Переместить распакованные файлы в директории ОС [*]

*cd rtsp-ws-proxy*
*cp rtsp-ws-proxy /usr/bin*
*mkdir /etc/rtsp/proxy && cp streams.yml*

## 7.2 Настройка rtsp-ws-proxy

- Отредактировать файл /etc/rtsp-ws-proxy/streams.yml

Добавить актуальные адреса rtsp-потоков и прописать порты для websocket.

```
server:
port: 8080
cameras:
- name: camera1
  stream: rtsp://1.1.1.1:554/live_mpeg4.sdp
  wsPort: 8081
  protocol: tcp
  fps: 25
  bitrate: 5000k
- name: camera2
  stream: rtsp://2.2.2.2/live_mpeg4.sdp
  wsPort: 8082
  protocol: udp
  fps: 20
```

*server.port - порт прокси сервера .*

Секция cameras - описание камер.
name - название камеры. Может быть любое.
stream - адрес rtsp-потока.

wsPort - порт webSocket по которому будет доступен видео-поток.
Опциональные значения:
protocol - протокол по которому передается rtsp-поток. Если не указан, по умолчанию используется tcp.
fps - количество кадров в секунду. Если не указано, по умолчанию используется 25.
bitrate - количество бит в секунду. Если не указан, то не используется.

### 7.3 Настройка ОС

- Настроить firewall

Для каждого порта в файле /etc/rtsp-ws-proxy/streams.yml выполнить:
*firewall-cmd --add-port=<wsPort>/tcp –permanent*

После чего выполнить:
*firewall-cmd --reload*