**СИСТЕМА КОНТРОЛЯ КАЧЕСТВА ОКАЗАНИЯ УСЛУГ**

Документация, содержащая описание функциональных характеристик программного обеспечения и информацию, необходимую для установки и эксплуатации программного обеспечения

на 6 страницах

Москва

2021

# Содержание

[Содержание 2](#_Toc81402793)

[1. Цель 3](#_Toc81402794)

[2. Функциональные возможности 3](#_Toc81402795)

[2.1. Мониторинг доступности оборудования 3](#_Toc81402796)

[2.2. Фиксация инцидентов в работе оборудования 4](#_Toc81402797)

[2.3. Определение качества оказанных поставщиками услуг 4](#_Toc81402798)

[2.4. Формирование отчета о недоступности оборудования 5](#_Toc81402799)

[2.5. Формирование отчета о работе оборудования 5](#_Toc81402800)

[2.6. Рассылка уведомлений о неработоспособности оборудования 5](#_Toc81402801)

[3. Информация для установки программного обеспечения 5](#_Toc81402802)

[3.1. Состав дистрибутива 5](#_Toc81402803)

[3.2. Развертывание СУБД 6](#_Toc81402804)

[3.3. Настройка модуля аутентификации агентов СКОУ 6](#_Toc81402805)

[3.4. Настройка модуля формирования отчетов 7](#_Toc81402806)

[3.5. Настройка модуля хранения отчётов 7](#_Toc81402807)

[3.6. Настройка модуля хранения текущих состояний 8](#_Toc81402808)

[3.7. Настройка модуля анализа текущих состояний 8](#_Toc81402809)

[3.8. Настройка модуля расчета отчетов СКОУ 9](#_Toc81402810)

[3.9. Настройка модуля рассылки электронных писем 10](#_Toc81402811)

[3.10. Настройка модуля сохранения статистики 10](#_Toc81402812)

[3.11. Настройка модуля сохранения таймлайнов 11](#_Toc81402813)

[3.12. Настройка модуля получения статистики от транспортных средств 11](#_Toc81402814)

[3.13. Настройка модуля проверки состояния rtsp потоков 13](#_Toc81402815)

[3.14. Настройка модуля получения ссылок HLS потоков 13](#_Toc81402816)

[3.15. Настройка модуля установки агентов СКОУ 13](#_Toc81402817)

[3.16. Запуск системы 14](#_Toc81402818)

[4. Информация для эксплуатации программного обеспечения 14](#_Toc81402819)

# Цель

Система контроля качества оказания услуг предназначена для оценки качества услуг, оказываемых поставщиками, и позволяет: формировать модели параметров качества услуг, контролировать соответствие фактических показателей заявленным, формировать отчеты.

# Функциональные возможности

Система контроля качества оказания услуг выполняет следующие функции:

* Мониторинг доступности оборудования;
* Фиксация инцидентов в работе оборудования;
* Определение качества оказанных поставщиками услуг;
* Формирование отчета о недоступности оборудования
* Формирование отчета о работе оборудования;
* Рассылка уведомлений о неработоспособности оборудования.

## Мониторинг доступности оборудования

Система контролирует работоспособность следующего оборудования:

* Серверы;
* Коммутаторы;
* Видеокамеры.

Работоспособность серверов определяются по следующим параметрам:

* Ping - время прохождения пакета, отправленного по протоколу ICMP на определённый адрес, в миллисекундах;
* Jitter - разброс минимального и максимального времени прохождения пакета от среднего времени прохождения пакета;
* Packetloss - количество пакетов, потерянных в сети во время передачи;
* RAM – загрузка оперативной памяти;
* HDD – загрузка жесткого диска;
* CPU – загрузка центрального процессора;
* Работоспособность ПО – проверка статуса работы программного обеспечения.

Работоспособность коммутаторов определяются по следующим параметрам:

* Ping - время прохождения пакета, отправленного по протоколу ICMP на определённый адрес, в миллисекундах;
* Jitter - разброс минимального и максимального времени прохождения пакета от среднего времени прохождения пакета;
* Packetloss - количество пакетов, потерянных в сети во время передачи.

Работоспособность видеокамер определяются по следующим параметрам:

* Ping - время прохождения пакета, отправленного по протоколу ICMP на определённый адрес, в миллисекундах;
* Jitter - разброс минимального и максимального времени прохождения пакета от среднего времени прохождения пакета;
* Packetloss - количество пакетов, потерянных в сети во время передачи.
* FPS - количество кадров в секунду;
* Bitrate - количество бит, используемых для передачи данных в единицу времени;
* resolution - разрешение видеоизображения.

## Фиксация инцидентов в работе оборудования

При нарушении пороговых значений система фиксирует инцидент в работе оборудования с указанием времени возникновения инцидента, параметра, вышедшего за пределы допустимых значений и времени завершения инцидента.

## Определение качества оказанных поставщиками услуг

На основании работы оборудования за отчетный период, система формирует отчет о уровне качества оказанных услуг. При выявлении нарушений в работе оборудования, система рассчитывает уровень фактически оказанных услуг, штрафы за нарушение работы, стоимость фактически оказанных услуг.

## Формирование отчета о недоступности оборудования

Система формирует отчет о недоступности оборудования за период времени в разрезе контракта. Отчет содержит в себе информацию о:

* Времени фиксации инцидента;
* Времени устранения инцидента;
* Параметрах, вышедших за пределы пороговых значений.

## Формирование отчета о работе оборудования

По окончанию отчетного периода, система формирует отчет о качестве оказанных услуг, включающий в себя:

* Продолжительность отчетного периода;
* Количество контролируемых объектов;
* Продолжительность работы оборудования;
* Продолжительность простоя оборудования;
* Объем архивных записей;
* Продолжительность работы оборудования со статусом «Не исправность»;
* Продолжительность потерянных архивных записей;
* Стоимость фактически оказанных услуг.

## Рассылка уведомлений о неработоспособности оборудования

При фиксации нарушений в работе оборудования, система рассылает оповещение о наличии инцидента по средствам электронных писем на адреса электронной почты поставщикам услуг

# Информация для установки программного обеспечения

## Состав дистрибутива

Для установки и настройки программы должно быть предварительно установлено системное и прикладное программное обеспечение:

1. RedHat 7 или CentOS 7
2. Программное обеспечение Docker

Для установки программы необходимо скопировать docker-образы и файлы конфигурации на настраиваемый сервер.

Для развертывания сервисов системы “Безопасный регион” необходимо:

1. загрузить docker-образы командой docker load < docker.tar.gz;
2. создать каталоги /opt/docker-configs, /opt/nginx/config, /opt/safecity-web, /opt/statistic\_monitoring, /opt/StatsAgent;
3. скопировать скрипты и файлы установки агента СКОУ в каталоги /opt/statistic\_monitoring и /opt/StatsAgent;
4. скопировать конфигурационные файлы в созданные каталоги и внести в них соответствующие изменения;
5. скопировать файлы запуска системы в каталог /opt/startup.

## Настройка модуля аутентификации агентов СКОУ

Конфигурационный файл /opt/docker-configs/agent\_auth\_server.conf:

app\_http\_port = "80"

# настройка подключения к сервису аутентификации

auth\_server\_address = "";

auth\_server\_username = "root";

auth\_server\_password = "12345";

server\_secret = "some\_secret";

# указание адресов внешних сервисов

vehicle\_server\_address = ""

immovable\_server\_address = ""

## Настройка модуля формирования отчетов

Конфигурационный файл /opt/docker-configs/reportexport-server.conf

[server\_settings]

host = 0.0.0.0

port = 80

log\_level = DEBUG

pdf\_storage = /storage

request\_timeout = 1800

# указание адресов внешних сервисов

[outer\_resource\_addresses]

immovable\_server\_address =

timeline\_server\_address =

clicksaver\_server\_address =

statssaver\_server\_address =

vehicle\_server\_address =

## Настройка модуля хранения отчётов

Конфигурационный файл /opt/docker-configs/stats\_saver.conf:

[uwsgi]

;plugin = python3

http-socket = :80

;http-keepalive = true

;add-header = Connection: Keep-Alive

chdir = ./

wsgi-file = app.py

processes = 1

;stats = :28765

requests\_timeout = 100

# настройка подключения к базе данных

database\_connection\_data = {"database\_address": "postgresql-staging-01", "port": 5432, "db\_name": "saferegion\_stats\_saver\_database", "user\_name": "postgres", "user\_password": "0064789"}

sqlalchemy\_log\_level = INFO

server\_log\_level = INFO

spooler = /tmp/spool\_dir

spooler-processes = 1

# указание адреса файлового хранилища

storage\_settings = {"location":"/app/storage"}

color\_logs = False

outer\_services\_errors\_context = False

add-header = Connection: Close

## Настройка модуля хранения текущих состояний

Конфигурационный файл /opt/docker-configs/status\_collector.conf:

[server]

host = 0.0.0.0

port = 80

log\_level = INFO

# указание адреса брокера сообщений

[kafka]

address =

consumer\_timeout\_ms = 1000

group\_id = stats\_collector\_staging

# указание адреса Redis

[redis]

address = saferegion-redis

port = 6379

db = 5

## Настройка модуля анализа текущих состояний

Конфигурационный файл /opt/docker-configs/status-stream-analyzer.conf:

[service]

push\_frequency\_sec = 15

status\_invalidation\_sec = 900

log\_level = INFO

maximum\_pack\_size = 1000

update\_scene\_change\_db\_interval = 60

min\_coef\_of\_scene\_change = 150

# указание адресов внешних сервисов

[immovable\_server]

address =

[vehicle\_server]

address =

[videoanalyzer]

update\_db\_interval = 60

[videoanalyzer\_server]

address =

# указание адреса брокера сообщений

[kafka]

address =

consumer\_timeout\_ms = 1000

## Настройка модуля расчета отчетов СКОУ

Конфигурационный файл /opt/docker-configs/straight-report-server.conf:

[server]

interface = 0.0.0.0

port = 80

[worker]

allowed\_ticket\_types = [1, 3]

# указание адресов Clickhouse

[stats\_clickhouse]

address =

max\_connections = 5

[semistats\_clickhouse]

address =

max\_connections = 5

# указание адресов внешних сервисов

[immovable]

address =

[vehicle]

address =

[saver]

address =

[logging]

level = INFO

## Настройка модуля рассылки электронных писем

Конфигурационный файл /opt/docker-configs/email\_service.conf:

[uwsgi]

http-socket = :80

chdir = ./

wsgi-file = app.py

processes = 4

stats = :28765

spooler = /tmp/spool\_stats\_master

spooler-processes = 1

email\_senders = {"objects": [{"name": "SR", "email": "example@abm-jsc.ru", "smtp\_port": "1025", "password": "12345", "smtp\_server": "127.0.0.1", "login": "example"}]}

database\_path = /storage/example.db

storage\_settings = {"location":"/storage"}

server\_log\_level = DEBUG

## Настройка модуля сохранения статистики

Конфигурационный файл /opt/docker-configs/clickhouse-saver-raw.conf:

[server\_settings]

host = 0.0.0.0

port = 80

log\_level = INFO

# указание параметров подключения к Clickhouse

[database\_settings]

clickhouse\_host = saferegion-clickhouse-raw

clickhouse\_port = 8123

clickhouse\_database = default

clickhouse\_username = default

clickhouse\_password =

insert\_max\_delay\_ms = 30000

insert\_buffer\_size = 6000

clickhouse\_compression = true

max\_partitions\_per\_insert\_block = 10000

[buffer\_settings]

insert\_buffer\_size = 100000

insert\_max\_delay\_sec = 40

# указание параметров подключения к Kafka

[kafka\_settings]

kafka\_server =

consumer\_timeout\_ms = 1000

topics = ["camera\_status", "vehicle\_camera\_status", "vehicle\_status", "server\_status", "network\_device\_status", "user\_logs"]

error\_topics = ["server\_status\_error", "camera\_status\_error", "vehicle\_status\_error", "vehicle\_camera\_status\_error"]

group\_id = clickhouse\_saver\_staging

## Настройка модуля сохранения таймлайнов

Конфигурационный файл /opt/docker-configs/clickhouse-saver-timeline.conf:

[server\_settings]

host = 0.0.0.0

port = 80

timeline\_semaphore = 1

log\_level = DEBUG

# указание параметров подключения к Clickhouse

[database\_settings]

clickhouse\_host = saferegion-clickhouse-timeline

clickhouse\_port = 8123

clickhouse\_database = default

clickhouse\_username = default

clickhouse\_password =

insert\_max\_delay\_ms = 30000

insert\_buffer\_size = 6000

clickhouse\_compression = true

[buffer\_settings]

insert\_buffer\_size = 6000

insert\_max\_delay\_sec = 20

## Настройка модуля получения статистики от транспортных средств

Конфигурационный файл /opt/docker-configs/vehicle-server-status-openresty.config:

local config = {

 kafka\_broker\_list = {{ host = "", port = 9092 }},

 kafka\_producer\_options = { producer\_type = "sync", keepalive\_size = 2000, socket\_timeout = 5000, request\_timeout = 4000 },

 expire\_key = 900,

 secret\_key = "QhNkNVBnAsKlSdBN",

 auth\_check = false

}

config.nginx = {

 log\_level = "info",

 resolver = "127.0.0.11"

}

config.servers = {

 vehicle = {

 cameras = {

 url = "",

 json\_request = { limit = 1000000, columns = { "id", "parts.cameras.id" } }

 },

 check\_interval = 5,

 intervals = 24,

 timeout = 10,

 expire\_keys = 300

 },

 server = {

 cameras = {

 url = "",

 json\_request = { limit = 1000000, columns = { "restreamer\_id", "id" }, filter\_by = { attribute = "restreamer\_id", operator = "is\_null", value = false }}

 },

 servers = {

 url = "",

 json\_request = { limit = 1000000, columns = { "id" } }

 },

 check\_interval = 5,

 intervals = 24,

 timeout = 10,

 expire\_keys = 300

 }

}

return config

Конфигурационный файл /opt/docker-configs/vehicle-server-status-updater.config:

[updater]

db\_url: postgres:0064789@postgresql-staging-01:5432/saferegion\_vehicle\_database

openresty\_url: http://saferegion-vehicle-server-status-openresty

log\_level: ERROR

## Настройка модуля проверки состояния rtsp потоков

Конфигурационный файл /opt/docker-configs/rtsp-checker.conf:

CheckDatabaseHost = "saferegion-redis"

CheckDatabaseNumber = "1"

WorkersAmount = "1"

MaximumNewChecksPerCycle = "100"

KafkaAddr = "10.10.27.38:9092"

StreamsToCheckTopic= "rtsp\_streams\_to\_check"

HostsLimitsTopic= "rtsp\_host\_connection\_limits"

EnableKafkaSend = "1"

## Настройка модуля получения ссылок HLS потоков

Конфигурационный файл /opt/docker-configs/netris\_hls\_server.config:

[server\_settings]

host = 0.0.0.0

port = 80

log\_level = DEBUG

netris\_cctv\_login = example

netris\_cctv\_password = 12345

public\_address = https://br.mosreg.ru

netris\_address = <http://10.10.27.80:8084>

## Настройка модуля установки агентов СКОУ

Конфигурационный файл /opt/docker-configs/statistics\_agent\_installer\_server.conf:

[server\_settings]

host = 0.0.0.0

port = 6543

request\_timeout = 400

[database\_settings]

host = 10.10.27.38

port = 5432

database\_name = statistics\_agent\_installer\_database

user = postgres

password = 0064789

[immovable\_server\_settings]

server\_address = http://10.10.27.38:80/immovable\_api

[storage\_settings]

storage\_path = /app/storage

[ansible\_settings]

private\_data\_dir\_path = /app/storage/private\_data\_dir

files\_relation\_path = files

playbook\_file\_name = playbook.yml

inventory\_file\_name = inventory

static\_file\_templates\_dir\_path = /app/file\_templates

[logging\_settings]

log\_level\_name = INFO

[server\_token\_generator\_settings]

agent\_type = 2

secret\_key = QhNkNVBnAsKlSdBN

## Запуск системы

Запуск системы выполняется последовательным запуском скриптов в папке /opt/startup. Очередность запуска определяется номером в названии скрипта.

# Информация для эксплуатации программного обеспечения

Система работает в автоматическом режиме. Программный агент с определенной периодичностью опрашивает оконечное оборудование и собирает данные о его работе. По окончанию отчетного периода, система формирует отчет о качестве оказанных услуг, включающий в себя:

* Продолжительность отчетного периода;
* Количество контролируемых объектов;
* Продолжительность работы оборудования;
* Продолжительность простоя оборудования;
* Объем архивных записей;
* Продолжительность работы оборудования со статусом «Не исправность»;
* Продолжительность потерянных архивных записей;
* Стоимость фактически оказанных услуг.