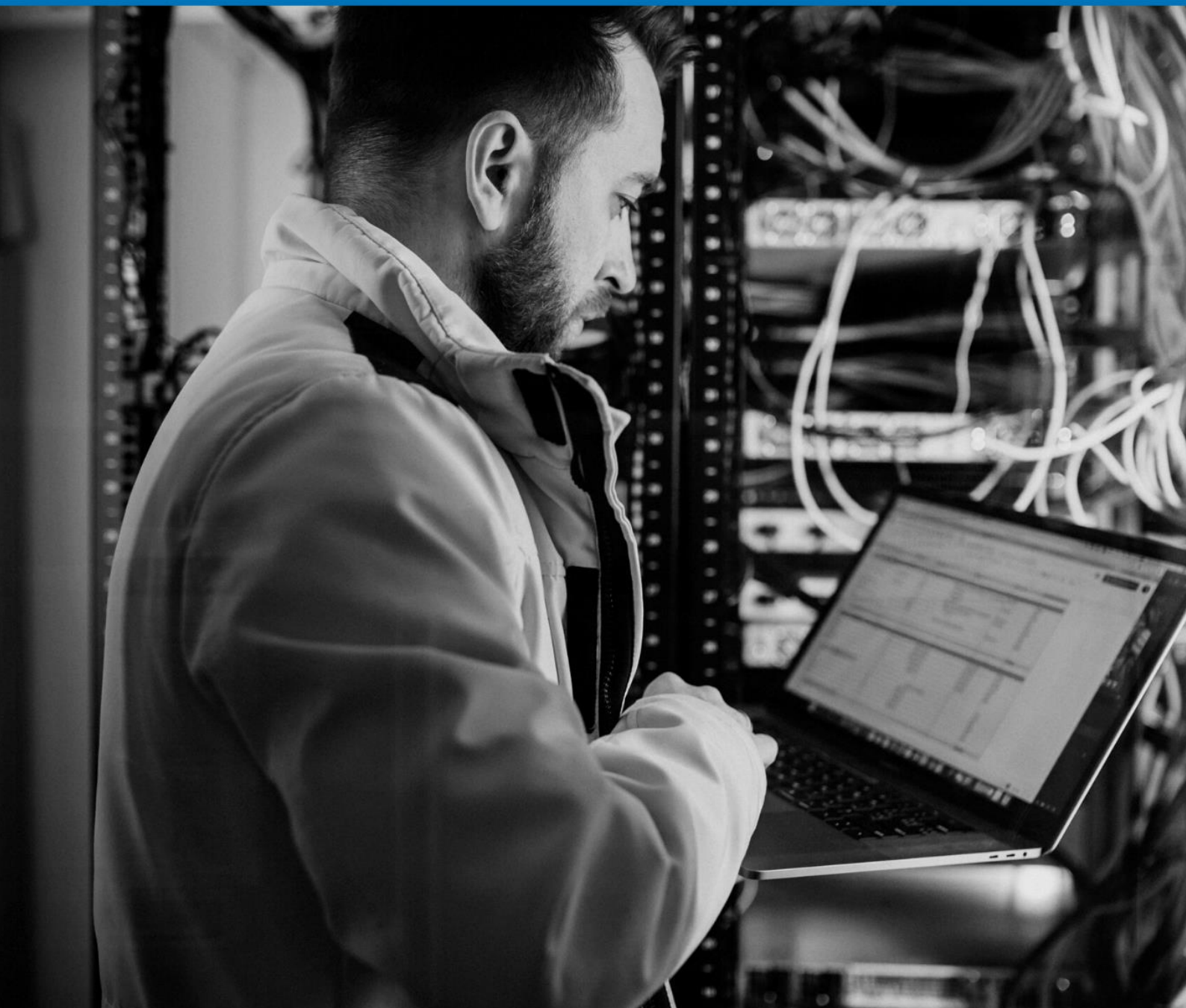
АРГУС
WFM CC

Руководство администратора

по работе с системой АРГУС WFM CC (Workforce Management for Call Centre)



Оглавление

1. Компоненты решения WFM CC

1.1 Назначение и состав программного решения WFM CC

Программное решение WFM CC предназначено для управления рабочими ресурсами Заказчика.

1.1.1 Перечень компонентов в составе решения WFM CC

В состав решения WFM CC входят следующие компоненты, включающих в себя функциональные сервисы и модули:

- Личный кабинет
- Мобильный API
- WFM CC
  - Модуль прогнозирования
  - Модуль планирования UI
  - Модуль мониторинга
- Планирование графиков работ и расписаний
  - Сервис планирования
  - Сервис Gateway
- Формирование отчетов
- Отправка уведомлений
- Интеграция с внешними системами
  - Сервис мониторинга

1.1.2 Опциональность поставки, типовые варианты поставки решения WFM CC

Компоненты в составе решения WFM CC могут поставляться в любом составе: как совместно, так и раздельно.

Поставка может осуществляться в виде пакета или в виде версии.

В пакет входит единичное изменение(исправление) по какому-либо из компонентов.

В версию входят изменения(исправление) сразу нескольких компонентов.

Компоненты могут быть установлены на СП и в БД

- в состав СП[1] входят:
  - Дистрибутивы (jar-файлы)
  - Плагины (jar-файлы)
  - Сервисы в образах Docker и конфигурационные файлы для загрузки образов
  - Сопроводительная документация (ПМУ, ПМИ, Протокол испытаний)
- в состав БД входят:
  - SQL-скрипты обновления (или утилита *dbmaintain*)
  - Сопроводительная документация (ПМУ, ПМИ, Протокол испытаний)

[1] Состав поставки для СП может варьироваться в зависимости от функционального назначения СП

1.2 Схема взаимодействия компонентов решения WFM CC

На рис. 1.2 стрелками показано направление взаимодействия между компонентами решения WFM CC

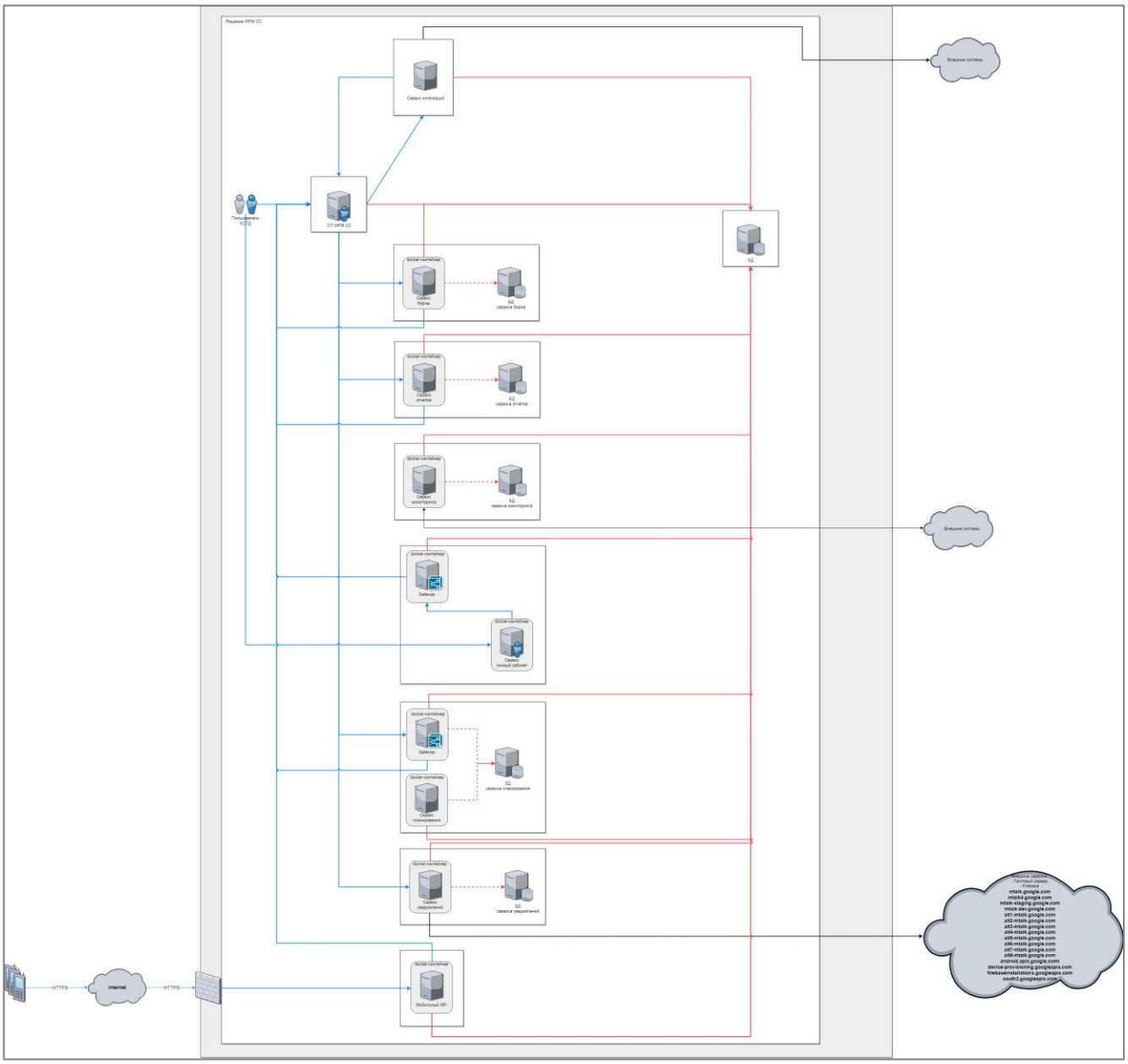


Рисунок 1.2 - Схема взаимодействия компонентов решения WFM CC

1.3 Жизненный цикл компонентов решения WFM CC

Жизненный цикл решения WFM CC состоит из этапов его внедрения и последующего сопровождения.

Степень ответственности на каждом этапе между ***Аргус*** и ***Заказчиком*** оговаривается индивидуально и фиксируется в виде приложения к договору под названием: ***Матрица ответственности***.

Матрица ответственности снабжается комментариями, поясняющими специфику проводимых работ.

Существуют следующие уровни ответственности:

R – Responsible (исполняет);

A – Accountable (несет ответственность);

C – Consult before doing (консультирует до исполнения);

I – Inform after doing (оповещает после исполнения);

S – Supported (оказывает поддержку).

Пример ***Матрицы ответственности*** представлен в таблице 1.3

Таблица 1.3 - Матрица ответственности

№	Процедура/Роль	Этап внедрения	Этап ГО/ПГО
1	Монтаж и настройка аппаратной составляющей ТА (серверов, СХД, системы резервного копирования).	НТЦ "Аргус": C Заказчик: RAIS	НТЦ "Аргус": C Заказчик: RAIS
2	Организация сетевого доступа сотрудникам Аргуса в сеть заказчика к оборудованию ТА.	НТЦ "Аргус": CS Заказчик: RAI	НТЦ "Аргус": CS Заказчик: RAI
3	Администрирование СУБД PostgreSQL[2].	НТЦ "Аргус": C Заказчик: RAIS	НТЦ "Аргус": Заказчик: RACIS
4	Администрирование экземпляров (прод/резерв) БД системы Аргус[3].	НТЦ "Аргус":CS Заказчик: RAI	НТЦ "Аргус": CS Заказчик: RAI
5	Мониторинг доступности экземпляров БД (прод/резерв)[4].	НТЦ "Аргус": Заказчик: RACIS	НТЦ "Аргус": Заказчик: RACIS
6	Администрирование системного программного обеспечения серверов ТА[5].	НТЦ "Аргус": C Заказчик: RAIS	НТЦ "Аргус": C Заказчик: RAIS

[2] Работы включают в себя установку СУБД, настройку СУБД.
[3] Работы с экземпляром БД включают в себя установку, мониторинг производительности.
[4] Мониторинг доступности должен осуществляться из сети/сетей пользователей, сети администраторов заказчика.
[5] Под администрированием понимается:
- начальная установка;
- установка патчей;
- настройка системного ПО;
- мониторинг работы системного ПО;
- мониторинг работы оборудования;
- оптимизация производительности;
- другие действия.

№	Процедура/Роль	Этап внедрения	Этап ГО/ПГО
7	Администрирование прикладного программного обеспечения на серверах ТА[6].	НТЦ "Аргус": CS Заказчик: RAI	НТЦ "Аргус": CS Заказчик: RAI
8	Администрирование рабочих мест/станций операторов системы Аргус[7].	НТЦ "Аргус": C Заказчик: RAIS	НТЦ "Аргус": C Заказчик: RAIS
9	Настройка процессов резервирования.	НТЦ "Аргус": C Заказчик: RAIS	НТЦ "Аргус": C Заказчик: RAIS
10	Мониторинг процессов резервирования и отказоустойчивости.	НТЦ "Аргус": Заказчик: RACIS	НТЦ "Аргус": Заказчик: RACIS

1.3.1 Внедрение решения WFM CC

Внедрение решения WFM CCпроходит следующие этапы:

- поставка решения
- развёртывание в тестовой зоне
- опытная эксплуатация
- развёртывание в продуктивной зоне
- опытно-промышленная эксплуатация
- приёмо-сдаточные испытания
- промышленная эксплуатация

1.3.2 Сопровождение решения WFM CC

Сопровождение решения WFM CC проходит следующие этапы:

- поставка обновления
- развёртывание в тестовой зоне
- приёмо-сдаточные испытания
- установка на продуктивную зону

[6] Относительно остальных серверов (не БД). Работы включают в себя установку и обновление прикладного ПО.

[7] Под администрированием понимается:
- установка и настройка системного и пользовательского ПО;
- обновление системного и пользовательского ПО.

2. Техническая архитектура решения WFM CC

2.1 Состав и аппаратные требования к компонентам решения WFM CC

В состав решения WFM CC входят следующие компоненты:

- БД WFM CC
- СП WFM CC
- Сервис Личный кабинет WFM CC
- Сервис Мобильный API WFM CC
- Сервис планирования
- Сервис отчетов
- Сервис уведомлений
- Сервис интеграций
- Клиентская система
- Сетевая система
- Внешние IT-системы (интеграция)
- Система мониторинга

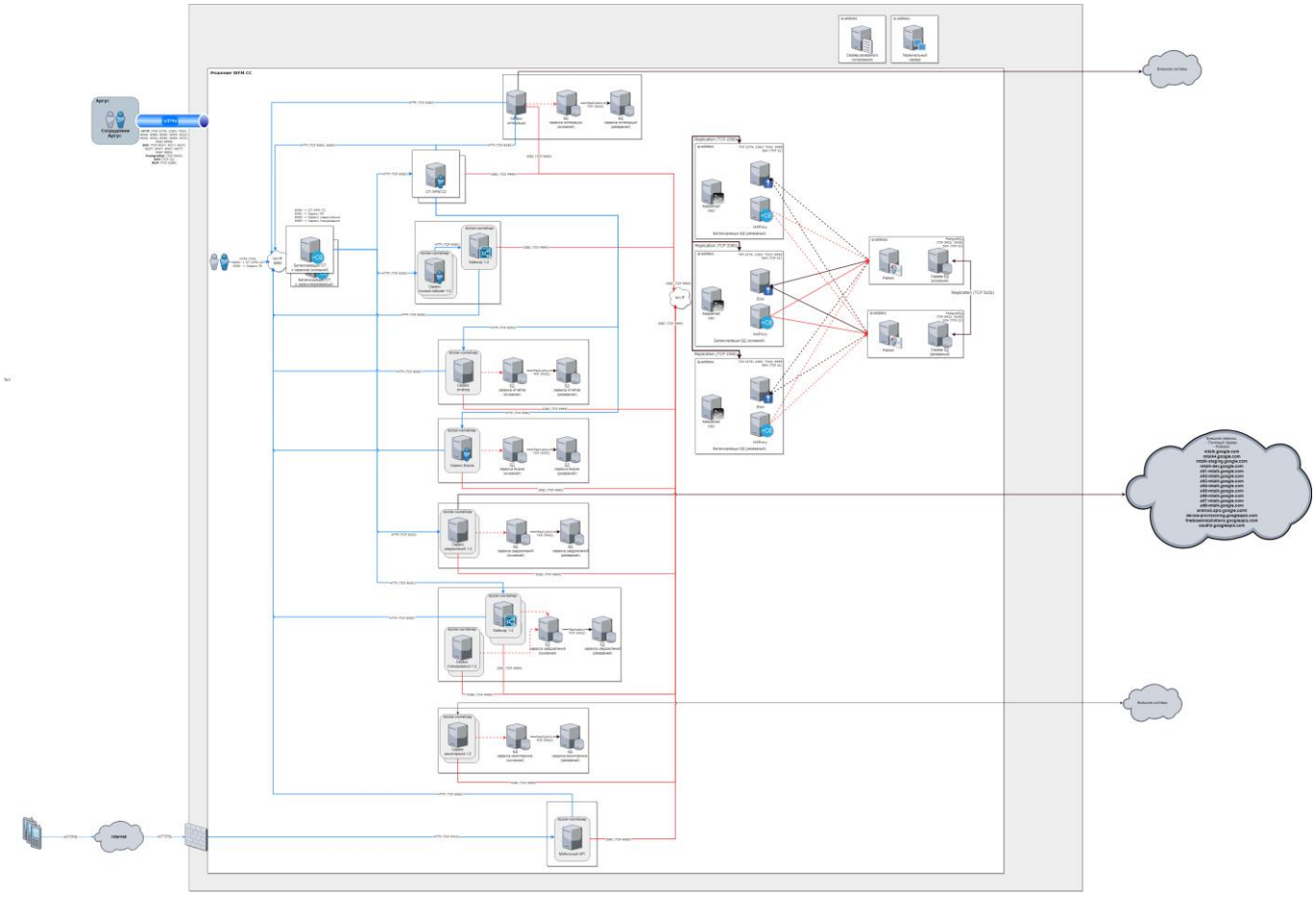


Рисунок 2.1 - Техническая архитектура решения WFM CC (пример)

Отказоустойчивость решения обеспечивается дублироваанием серверных компонентов и архитектурно предусматривает их горизонтальное масштабирование.

2.1.1 БД решения WFM CC

В решении WFM CC для каждой из БД

- БД WFM CC
- БД интеграции
- БД планирования
- БД уведомлений
- БД отчетов

поддерживается СУБД Postgres от версии 15.x, СУБД Arenadata Prosperity

Защита от потери данных реализуется с использованием технологи репликации баз данных методом Master-Salve

2.1.1.1 Требования к CPU, RAM БД WFM CC

Требования к ресурсам CPU[8], RAM рассчитываются исходя из суммарной нагрузки, оказываемой каждым из компонентов решения WFM CC, использующих БД WFM CC

Для работы системных процессов ОС необходимо

- CPU: 1 ядро
- RAM: 2G

Таблица 2.1.1.1 - Требования к ресурсам БД

Источник нагрузки	**Требования к ресурсам БД**
СП WFM CC	• CPU (БД): 1 ядро на каждые 10 одновременно открытых (конкурентных) сессий (прогнозирования, планирования, мониторинга) • RAM (БД): 4G на каждые 10 одновременно открытых (конкурентных) сессий (прогнозирования, планирования, мониторинга)
Сервис ЛК	• CPU (БД): 1 ядро на каждые 100 одновременно открытых (конкурентных) сессаий пользователей • RAM (БД): 4G на каждые 100 одновременно открытых (конкурентных) сессий пользователей
Сервис интеграции	для каждой из интеграций • CPU (БД): 1 ядро • RAM (БД): 2G
Сервис отчетов	• CPU (БД) 1 ядро • RAM (БД) 2G
Сервис Мобильный API	при 20 запросах в секунду (req\sec) и средней продолжительности запроса в 3 секунды для каждых 500 операторов: • CPU (БД): 1 ядро • RAM (БД): 2G
Сервис уведомлений	ресурсов не требуется

[8] CPU типа Intel Xeon e5-2640 (или его аналог)

Источник нагрузки	Требования к ресурсам БД
Сервис планирования	ресурсов не требуется

Итоговые требования

CPU (ОС) = суммарное количество ядер CPU (БД) + 1 ядро (для системных процессов ОС)

RAM (ОС) = (суммарное RAM (БД) * 1.5) + 2G (для системных процессов ОС)

После итогового расчёта по ресурсам следует взять понижающий коэффициент 0.75 для CPU и для RAM[9], поскольку маловероятно, что все перечисленные источники нагрузки будут одновременно обращаться к БД в своих пиковых значениях.

2.1.1.2 Требования к сетевым интерфейсам БД WFM CC

На хосте с сервером БД требуется наличие не менее двух сетевых интерфейсов Gigabit Ethernet.

2.1.1.3 Требования к портам БД WFM CC

На хосте для обращения к БД должен быть открыт порт 5432.

Порт не должен использоваться операционной системой или другими приложениями.

2.1.1.4 Требования к СХД БД WFM CC

При выборе СХД следует учесть динамику роста системы, исходя из количества пользователей и групп, формирующих основной объем данных.

Таблица 2.1.1.4 – Прирост ресурсов

Таблица	Прирост
worker_chage_status_log	один пользователь формирует в сутки данных на 4К[10]
historical_data	одна группа формирует в сутки данных на 14К[11]

Требования к производительности СХД выбираются исходя из нагрузки.

2.1.2 СП WFM CC

2.1.2.1 Требования к CPU, RAM, HDD СП WFM CC

Ресурсы, необходимые для работы СП, рассчитываются исходя из числа конкурентных сессий пользователей, выполняющих задачи:

- прогнозирования (forecast open sessions - **fos**)

[9] При расчетах значение RAM должно быть не менее 8G вне зависимости от количества пользователей, так как есть возможность используя исторические данные делать запросы неконтролируемой сложности.
[10] 100 записей в сутки по 40К каждая
[11] 288 записей в сутки по одному проекту (в разрезе 5 минут) * 48 байт (на одну запись)

- планирования UI[12] (planning open sessions - **pos**)
- мониторинга (monitoring open session - **mos**)

и исходных данных для каждого модуля

- прогнозирования
  - длительность периода исторических данных для составления прогнозов, в годах (historical data period - **hdp**)
  - период прогнозирования, в годах (forecast data period - **fdp**)
- планирования UI
  - количество операторов в шаблоне планирования (schedule template worker number - **stwn**)
- мониторинга
  - количество групп, за которым может наблюдать один супервайзер (monitoring group number - **mgn**)

Требования к CPU (cores)[13]

- 1 ядро на каждую конкурентную пользовательскую сессию
- 1 ядро для системных процессов ОС

Требования к RAM

- 2G для системных процессов ОС

RAM для JVM (MB)

- запуск экземпляра СП: 2048 M
- модуль прогнозирования: 4096 + (**hdp** + **fdp**) * 512 * **fos**
- модуль планирования UI[14]:
- Отображение графиков e^(6.558+0.002 * **stwn**) * **pos**
- Отображение расписаний e^(4.693+0.004 * **stwn**) * **pos**
- модуль мониторинга: (1500 + 25 * **mgn**) * **mos**

RAM (JVM) MB = 2048
+ 4096 + (**hdp** + **fdp**) * 512 * **fos**
+ e^(6.558+0.002 * **stwn**) * **pos**
+ e^(4.693+0.004 * **stwn**) * **pos**
+ (1500 + 25 * **mgn**) * **mos**
RAM (итоговое для ОС) = (RAM для JVM) * 1.5 + 2G (для системных процессов ОС)

Требования к HDD

Полезная емкость HDD

- 50G для ОС
- 100G для хранения ПО и логов с учетом обычного режима работы (не включая DEBUG).

[12] отображение результатов планирования в UI
[13] CPU типа Intel Xeon e5-2640 (или его аналог)
[14] требования к ресурсам носят экспоненциальную зависимость

Рекомендуется использовать отказоустойчивые массивы (например, RAID-1, RAID-10)

2.1.2.2 Требования к сетевым интерфейсам СП WFM CC

На хосте для работы СП WFM CC требуется наличие сетевого интерфейса с пропускной способностью 100Mbit/s.

2.1.2.3 Требования к портам СП WFM CC

На хосте для работы СП WFM CC должны быть открыты следующие порты

Таблица 2.1.2.3 – Требования к портам СП WFM CC

Порт	Протокол	Назначение
8080	HTTP	обслуживание HTTP запросов от браузеров пользователей и других компонентов решения WFM CC
9990	JMX	management-порт для доступа к веб-интерфейсу управления сервером приложений

Порты не должны использоваться операционной системой или другими приложениями.

2.1.3 Сервис Личный кабинет WFM CC

2.1.3.1 Требования к CPU, RAM, HDD. Сервис Личный кабинет WFM CC

Ресурсы, необходимые для работы сервиса 'Личный кабинет', рассчитываются исходя из числа конкурентных сессий пользователей сервиса (personal area open sessions - **paos**)

Требования к CPU (cores)[15]

- 1 ядро на каждые 100 конкурентных пользовательских сессий
- 1 ядро для системных процессов ОС

Требования к RAM

- 2G для системных процессов ОС

RAM для JVM

- запуск экземпляра СП: 2048 M
- Личный кабинет: 120 M * paos

RAM для JVM = 2048 M + 120 M * **paos**

RAM (итоговое для ОС) = (RAM для JVM) * 1.5 + 2G (для системных процессов ОС)

Требования к HDD

Полезная емкость HDD

- 50G для ОС
- 100G для хранения ПО и логов с учетом обычного режима работы (не включая DEBUG).

Рекомендуется использовать отказоустойчивые массивы (например, RAID-1, RAID-10)

[15] CPU типа Intel Xeon e5-2640 (или его аналог)

2.1.3.2 Требования к сетевым интерфейсам. Сервис 'Личный кабинет' WFM CC

На хосте для работы сервиса требуется наличие сетевого интерфейса с пропускной способностью 100Mbit/s.

2.1.3.3 Требования к портам. Сервис 'Личный кабинет' WFM CC

На хосте для работы сервиса должны быть открыты следующие порты

Таблица 2.1.3.3 – Требования к портам ЛК WFM CC

Порт	Протокол	Назначение
9050	HTTP	обслуживание HTTP запросов от браузеров пользователей

Порты не должны использоваться операционной системой или другими приложениями.

2.1.4 Сервис Мобильный API WFM CC

2.1.4.1 Требования к CPU, RAM, HDD. Сервис Мобильный API WFM CC

Ресурсы, необходимые для работы сервиса 'Мобильный API', рассчитываются исходя из числа конкурентных сессий пользователей сервиса
и следующих параметрах нагрузки

- 20 запросов в сек.
- средняя продолжительность запроса 3 сек.

Требования к CPU (cores)[16]

- 2 ядра на каждые 500 конкурентных пользовательских сессий
- 1 ядро для системных процессов ОС

Требования к RAM

- 2G для системных процессов ОС

RAM для JVM

- 2G на каждые 500 конкурентных пользовательских сессий

Требования к HDD

Полезная емкость HDD

- 50G для ОС
- 100G для хранения ПО и логов с учетом обычного режима работы (не включая DEBUG).

Рекомендуется использовать отказоустойчивые массивы (например, RAID-1, RAID-10)

2.1.4.2 Требования к сетевым интерфейсам. Сервис Мобильный API WFM CC

На хосте для работы сервиса требуется наличие сетевого интерфейса с пропускной способностью 100Mbit/s.

2.1.4.3 Требования к портам. Сервис Мобильный API WFM CC

На хосте для работы сервиса должны быть открыты следующие порты

[16] CPU типа Intel Xeon e5-2640 (или его аналог)

Таблица 2.1.4.3 – Требования к портам Мобильный API WFM CC

Порт	Протокол	Назначение
9010	HTTP	обслуживание HTTP запросов от удаленных пользователей
9017	JMX	management-порт для доступа к интерфейсу управления сервисом

Порты не должны использоваться операционной системой или другими приложениями.

2.1.5 Сервис планирования

2.1.5.1 Требования к CPU, RAM, HDD. Сервис планирования

Сервис планирования включает в себя планирование графиков работ и расписаний.

Ресурсы, необходимые для работы сервиса планирования[17], рассчитываются исходя из числа конкурентных сессий пользователей

- количество конкурентных сессий планирования
- и следующих параметров нагрузки
- количество операторов в шаблоне планирования
- количество одновременно выполняемых задач планирования
- количество потоков на сессию планирования

Требования к CPU (cores)[18]

- 1 ядро для системных процессов ОС
- CPU (сервис планирования): 2 + (количество одновременно выполняемых задач планирования * количество потоков на сессию планирования)
- CPU (сервис gateway): 1 ядро[19]

Требования к RAM (MB)

- 2G для системных процессов ОС

RAM для JVM

- RAM (JVM сервис планирования) = (5M * количество операторов в шаблоне планирования * количество одновременно выполняемых задач планирования * количество потоков на сессию планирования)
- RAM (JVM сервис gateway) = 100Мб + (0,5 M * количество конкурентных сессий планирования)

RAM (ОС итоговое) = (RAM (JVM сервис планирования) + RAM (JVM сервис gateway)) * 1.5 + 2G (для системных процессов ОС)

Требования к HDD

Полезная емкость HDD

[17] Расчет требований для сервиса планирования осуществляется совместно с расчетом для сервиса gateway. оба сервиса работают на одном хосте
[18] CPU типа Intel Xeon e5-2640 (или его аналог)
[19] При нагрузке 10 одновременных запросов на планирование в секунду

- 50G для ОС
- 100G для хранения ПО и логов сервиса планирования с учетом обычного режима работы (не включая DEBUG).
- 100G для хранения ПО и логов сервиса gateway с учетом обычного режима работы (не включая DEBUG).

Рекомендуется использовать отказоустойчивые массивы (например, RAID-1, RAID-10)

2.1.5.2 Требования к сетевым интерфейсам. Сервис планирования

На хосте для работы сервиса требуется наличие сетевого интерфейса с пропускной способностью 100Mbit/s.

2.1.5.3 Требования к портам. Сервис планирования

На хосте для работы сервиса должны быть открыты следующие порты

Таблица 2.1.5.3 – Требования к портам СП Планирования

Порт	Протокол	Назначение
9030	HTTP	обслуживание HTTP запросов от других компонентов решения WFM CC
9037 9047	JMX	management-порт для доступа к интерфейсу управления сервисом

Порты не должны использоваться операционной системой или другими приложениями.

2.1.6 Сервис отчетов

2.1.6.1 Требования к CPU, RAM, HDD. Сервис отчетов

Ресурсы, необходимые для работы сервиса отчетов, рассчитываются исходя из количества конкурентных задач построения отчетов

Требования к CPU (cores)[20]

- 1 ядро для системных процессов ОС
- 1 ядро для одной задачи построения отчета

Требования к RAM

- 2G для системных процессов ОС

RAM для JVM

- 2G для одной задачи построения отчета

RAM (JVM) = 2G * количество конкурентных задач построения отчетов

RAM (ОС итоговое) = RAM (JVM) * 1.5 + 2G (для системных процессов ОС)

Требования к HDD

Полезная емкость HDD

[20] CPU типа Intel Xeon e5-2640 (или его аналог)

- 50G для ОС
- 500G[21] для хранения ПО и логов с учетом обычного режима работы (не включая DEBUG).

Рекомендуется использовать отказоустойчивые массивы (например, RAID-1, RAID-10)

2.1.6.2 Требования к сетевым интерфейсам. Сервис отчетов

На хосте для работы сервиса требуется наличие сетевого интерфейса с пропускной способностью 100Mbit/s.

2.1.6.3 Требования к портам. Сервис отчетов

На хосте для работы сервиса должны быть открыты следующие порты

Таблица 2.1.6.3 – Требования к портам Сервис отчётов

Порт	Протокол	Назначение
9000	HTTP	обслуживание HTTP запросов от других компонентов решения WFM CC
9007	JMX	management-порт для доступа к интерфейсу управления сервисом

Порты не должны использоваться операционной системой или другими приложениями.

2.1.7 Сервис уведомлений

2.1.7.1 Требования к CPU, RAM, HDD. Сервис уведомлений

Ресурсы, необходимые для работы сервиса уведомлений рассчитываются исходя из количества[22]

- одновременных потоков обработки уведомлений
- одновременных потоков рассылки

Требования к CPU (cores)[23]

- 1 ядро для системных процессов ОС
- 1 ядро * (количество одновременных потоков рассылки / 10)
- 1 ядро * (количество одновременных потоков обработки уведомлений / 10)

Требования к RAM

- 2G для системных процессов ОС

RAM для JVM

- 512M для рассылки
- 512M + 30M * количество одновременных потоков обработки уведомлений

RAM (JVM) = 512M + 512M + 30M * количество одновременных потоков обработки уведомлений

RAM (ОС итоговое) = RAM (JVM) * 1.5 + 2G (для системных процессов ОС)

Требования к HDD

[21] Окончательный значение HDD зависит от того, сколько места на диске занимает сформированный отчет, количества отчетов и времени их хранения

[22] Обычно 20 одновременных потоков обработки уведомлений и 10 одновременных потоков рассылки достаточно для работы сервиса уведомлений

[23] CPU типа Intel Xeon e5-2640 (или его аналог)

Полезная емкость HDD

- 50G для ОС
- 100G для хранения ПО и логов с учетом обычного режима работы (не включая DEBUG).

Рекомендуется использовать отказоустойчивые массивы (например, RAID-1, RAID-10)

2.1.7.2 Требования к сетевым интерфейсам. Сервис уведомлений

На хосте для работы сервиса требуется наличие сетевого интерфейса с пропускной способностью 100Mbit/s.

2.1.7.3 Требования к портам. Сервис уведомлений

На хосте для работы сервиса должны быть открыты следующие порты

Таблица 2.1.7.3 – Требования к портам Сервис уведомлений

Порт	Протокол	Назначение
9020	HTTP	обслуживание HTTP запросов от других компонентов решения WFM CC
9027	JMX	management-порт для доступа к интерфейсу управления сервисом

Порты не должны использоваться операционной системой или другими приложениями.

2.1.8 Сервис интеграций

2.1.8.1 Требования к CPU, RAM, HDD. Сервис интеграций

Ресурсы, необходимые для работы сервиса интеграций рассчитываются исходя из количества интеграций[24]

Требования к CPU (cores)[25]

- 1 ядро для системных процессов ОС
- 1 ядро для каждой интеграции

Требования к RAM

- 2G для системных процессов ОС

RAM для JVM

- 2G для каждой интеграции

RAM (JVM) = 2G * количество интеграции

RAM (ОС итоговое) = RAM (JVM) * 1.5 + 2G (для системных процессов ОС)

Требования к HDD

Полезная емкость HDD

- 50G для ОС
- 100G для хранения ПО и логов с учетом обычного режима работы (не включая DEBUG).

[24] В примере приведен типовой расчет. реальное требование к ресурсам может быть иным в зависимости от интенсивности нагрузки, создаваемой каждой из интеграций

[25] CPU типа Intel Xeon e5-2640 (или его аналог)

Рекомендуется использовать отказоустойчивые массивы (например, RAID-1, RAID-10)

2.1.8.2 Требования к сетевым интерфейсам. Сервис интеграций

На хосте для работы сервиса требуется наличие сетевого интерфейса с пропускной способностью 100Mbit/s.

2.1.8.3 Требования к портам. Сервис интеграций

На хосте для работы сервиса должны быть открыты следующие порты

Таблица 2.1.8.3 – Требования к портам Сервис интеграции

Порт	Протокол	Назначение
8080	HTTP	обслуживание HTTP запросов от других компонентов решения WFM CC

Порты не должны использоваться операционной системой или другими приложениями.

2.1.9 Сервис мониторинга

2.1.9.1 Требования к CPU, RAM, HDD. Сервис мониторинга

Является подсервисом сервиса интеграции.

2.1.9.2 Требования к сетевым интерфейсам. Сервис интеграций

На хосте для работы сервиса требуется наличие сетевого интерфейса с пропускной способностью 100Mbit/s.

2.1.9.3 Требования к портам. Сервис интеграций

На хосте для работы сервиса должны быть открыты следующие порты

Таблица 2.1.9.3 – Требования к портам Сервис мониторинга

Порт	Протокол	Назначение
9070	HTTP	обслуживание HTTP запросов от других компонентов решения WFM CC
9077	JMX	management-порт для доступа к интерфейсу управления сервисом

Порты не должны использоваться операционной системой или другими приложениями.

2.1.10 Клиентская система

В состав клиентской системы входят:

- Web-client, предназначенный для работы с СП WFM CC и Сервисом Личный кабинет WFM CC, используя web-браузер
- Mobile-client, предназначенный для работы с СП WFM CC используя мобильное приложение

2.1.10.1 Требования к CPU, RAM, HDD Web-client

Таблица 2.1.9.1 - Требования к аппаратной части

Название	Минимальные требования	Рекомендуемые требования
CPU	x86 двухъядерный выпуска 2010 года (или более новый);	x86 двухъядерный выпуска 2010 года (или более новый);

RAM	от 2048 Мб	от 8 Гб
HDD	10 Gb	30 Gb
Разрешение экрана	1280x1024	1920x1080

2.1.10.2 Требования к сетевым интерфейсам. Клиентская системы. Web-client

Рекомендуемая пропускная способность сетевого интерфейса клиентского рабочего места для клиентов КСПД: 100 Мбит/с

2.1.10.3 Требования к портам. Клиентская система. Web-client и Mobile-client

Для работы пользователей КСПД использующих Web-client, с рабочих мест должен быть открыт доступ к СП WFM CC и Сервису Личный кабинет WFM CC по портам[26], указанным в п. 2.1.2.3 Требования к портам СП WFM CC, п. 2.1.3.3 Требования к портам. Сервис 'Личный кабинет' WFM CC.

В случае применения отказоустойчивого решения[27] или необходимости использовать протокол HTTPS, нужно предоставить доступ с клиентских рабочих мест на балансировщик по настроенным на нем портам для каждого из сервисов: СП WFM CC и Сервис Личный кабинет WFM CC.

Для работы удалённых пользователей использующих Mobile-client, из интернет должен быть предоставлен доступ к Сервису Мобильный API WFM CC по портам[28], указанным в п. 2.1.4.3 Требования к портам. Сервис Мобильный API WFM CC

2.1.11 Сетевая система

2.1.11.1 Требования к каналам передачи данных

Каналы передачи данных между сетевыми интерфейсами систем, входящих в решение WFM CC, должны обеспечивать необходимую пропускную способность, указанную в пп.

2.1.11.2 Требования к портам

Для всех систем, входящих в решение WFM CC, должна быть обеспечена IP-связанность согласно ***рис.2.1 Техническая архитектура решения WFM CC***

п. 2.1 Состав и аппаратные требования к компонентам решения WFM CC, указанным в пп.

2.1.1.3 Требования к портам БД WFM CC

2.1.2.3 Требования к портам СП WFM CC

2.1.3.3 Требования к портам. Сервис 'Личный кабинет' WFM CC

2.1.4.3 Требования к портам. Сервис Мобильный API WFM CC

2.1.5.3 Требования к портам. Сервис планирования

---

[26] В базовой конфигурации это протокол HTTP и порт 8080 как для СП WFM CC, так и для Сервиса Личный кабинет WFM CC

[27] Наличие нескольких дублирующих сервисов развернутых на разных хостах

[28] В базовой конфигурации внутри КСПД это протокол HTTP и порт 8080. терминация трафика HTTPS - HTTP происходит на уровне сетевого оборудования заказчика.

2.1.11.3 Требования к балансировщику нагрузки сервисов WFM CC

На балансировщике должны быть сформированы балансируемые группы

Балансируемой группой называем группу сервисов, имеющих одинаковое предназначение, состоящую из N экземпляров сервисов в целях балансировки нагрузки и обеспечения failover.

- Для каждой группы на балансировщике открывается соответствующий порт (см. таблицу 2.1.10.3)
- Для некоторых балансируемых групп балансировщик обеспечивает sticky session (см. таблицу 2.1.10.3)

Таблица 2.1.10.3 - Пример балансируемых групп и портов

Имя группы	**Входящий порт на балансировщике для работы пользователей и систем**	**Состав группы**	**Требуется ли Sticky session**	**Проверка доступности сервисов с использованием managent-портов**
СП WFM CC	8080	argus-app01:8080 argus-app02:8080	Да	http://argus-app01:9990/ccwfm/ping http://argus-app02:9990/ccwfm/ping
Сервис Личный кабинет WFM CC	8081	argus-app03:8081, argus-app04:8081	Да	http://argus-app03:9990/api/v1/system/status http://argus-app04:9990/api/v1/system/status
Сервис уведомлений	8082	argus-app05:8082, argus-app06:8082	Нет	http://argus-app05:9990/api/v1/system/status http://argus-app06:9990/api/v1/system/status
Сервис планирования	8083	argus-app07:8083, argus-app08:8083	Нет	http://argus-app07:9990/api/v1/system/status http://argus-app08:9990/api/v1/system/status

- Балансировщик перенаправляет входящие на порт группы запросы на выбранный им экземпляр сервиса в балансируемой группе, обеспечивая балансировку нагрузки и отказоустойчивость

(***failover***), то есть автоматическое переключение пользователей с аварийного узла на рабочие в пределах балансируемой группы.

- Распределение сессий между активными узлами осуществляется по идентификаторам сессий с использованием механизма ***cookie***, а также в зависимости от статуса узла: активен/не активен. При этом выбирается наименее загруженный узел исходя из количества соединений или количества трафика на сервер.

Если сервисы, на которые перенаправляются запросы, имеют разные ресурсные характеристики, то каждому сервису в серверной ферме ставится в соответствие коэффициент – вес. Вероятность направления запроса на конкретный сервис определяется как отношение его собственного веса к сумме весов всех серверов фермы.

- Для обеспечения ***Sticky session*** балансировщик должен направлять очередной запрос на тот же экземпляр сервиса, на который был отправлен предыдущий запрос той же сессии.

Идентификатор сессии указывается в ***cookie*** с наименованием ***jsessionid***.

Если в запросе нет указанной ***cookie***, запрос идёт вне сессии и может быть направлен на любой экземпляр сервиса.

Cookie устанавливается экземпляром сервиса при обработке запроса. Значение ***cookie*** имеет формат:

<идентификатор_сессии>.<идентификатор_сервиса>

где:

<идентификатор_сервиса> – идентификатор экземпляра сервиса, указанный в конфигурации сервиса; должен использоваться балансировщиком для выбора экземпляра сервиса, на который следует отправить запрос

<идентификатор_сессии> - состоит из цифр и символов латинского алфавита, генерируется экземпляром сервиса и не должен анализироваться балансировщиком

- Таймаут хранения ***Sticky session*** на балансировщике отсчитывается от окончания последней активности пользователя и составляет не более, чем значение ***Http SessionTimeout*** сервиса.

- Балансировщик обеспечивает наличие в http-запросах заголовков, содержащих ip-адрес клиента или внешней системы, с которого отправлен запрос.

Если это аппаратный балансировщик, то в заголовке ***«X-Forwarded-For»*** должен передаваться исходный IP-адрес клиента.

- Проверка доступности сервисов осуществляется путем установки HTTP проб для членов группы каждой из групп при помощи get-запросов по адресам, приведенным в таблице 2.1.10.3

- код успешного ответа: 200
- интервал между пробами: 10 сек.

Если запрос не успешен два раза подряд, балансировщик должен считать сервис недоступным, иначе доступным.

Таймауты балансировщика:

- ***Таймаут подключения*** от балансировщика к узлу сервиса должен быть не менее 1 мин.
- ***Таймаут ожидания ответа по ajp-ping*** должен быть не менее 1 мин.

Так как время ***Full GC*** под нагрузкой может составлять до десятков секунд, в зависимости от размера кучи (***heap***).

- ***Таймаут ожидания ответа реквеста от узла сервиса*** должен быть не менее 24 мин, чтобы избегать случаев, когда долгий реквест (например, построение отчета) был прерван балансировщиком, а узел из-за этого признан аварийным.

Максимальное время выполнения реквеста на сервере 23 мин, далее он прерывается таймаутом на стороне сервиса.

В случае, если узел может быть признан аварийным на основании долгого выполнения одного реквеста (для ***mod_jk*** - параметр ***reply_timeout =*** *1380*),

время пребывания в аварийном состоянии (для ***mod_jk*** параметр ***recover_time*** = 60 (по умолчанию) .. 120) не должно превышать 1-2 мин (так как возможны ложные признания узла аварийным).

В случае, если возможно устанавливать порог количества длительных реквестов в единицу времени

(для ***mod_jk*** параметр ***max_reply_timeouts****)*,

время пребывания узла в аварийном состоянии может быть увеличено до 10 мин.

(для ***mod_jk*** увеличивается значение параметра ***recover_time*** = 600);

- Балансировщик должен считать запрос неуспешным если:

Сервис вернул http статус 500-599 как результат обработки запроса.

Остальные значения статусов, в частности 300-399, не должны считаться признаком неуспешности запроса.

- Балансировщик может прозрачно для пользователя повторить выполнение запроса на другом узле, кроме случаев:

  - Post запрос принят на обработку сервисом и обрабатывается слишком долго В этом случае балансировщик не должен перенаправлять запрос на другой сервис во избежание дублирования изменений выполняемых запросом в БД.

- Балансировщик не должен направлять запрос на недоступный сервис, если в балансируемой группе остались доступные сервисы.

Пока узел находится в аварийном статусе балансировщик не будет отправлять ему запросы.

Запросы новых сессий будут перенаправляться на другой узел и "не заметят" аварии узла.

Запросы в рамках сессий аварийного узла будет перенаправлены на другой узел и начнут там новые сессии, а значит пользователь должен будет авторизоваться.

- По возможности (если балансировщик поддерживает), следует использовать балансировку по протоколу ***ajp***, а не ***http***. Программные балансировщики ***mod_jk*** и ***mod_proxy_balancer*** поддерживают протокол ***ajp***.

В случае взаимодействия с узлами по ***http***, а не ***ajp***, балансировщик должен записывать http-заголовок ***X-Forwarded-For***. Например, для программных балансировщиков на базе Apache httpd в ***mod_proxy_balancer*** регулируется атрибутом ***ProxyAddHeaders***, который по-умолчанию уже включен.

- На балансировщике должны вестись логи доступа (accesslog).

По возможности (если балансировщик поддерживает) логироваться должна следующая информация:

- *Время завершения обработки реквеста;*
- *IP-адрес клиента;*
- *Значение поля X-Forwarder-For заголовка реквеста;*
- *Значение cookie JSESSIONID реквеста;*
- *Первая строка реквеста;*
- *Статус оригинального реквеста (не редиректа);*
- *Размер ответа в байтах;*
- *Время обработки реквеста в мс;*
- *Значение поля Referer заголовка реквеста;*
- *Значение поля User-Agent заголовка реквеста;*

Например, для программных балансировщиков на базе Apache ***httpd*** (***mod_jk, mod_proxy_balancer, mod_cluster***), требуется указать такое значение настройки для модуля ***mod_log_config***:

LogFormat "%t %a %{X-Forwarded-For}i - [-] %{JSESSIONID}C %r %s - %b %{ms}T %{Referer}i \"%{User-Agent}i\" -" custom

В нем несколько фиксированных столбцов ("-", "[-]") добавлены для совпадения формата с access-логами самих узлов.

- На балансировщике должна быть настроена ротация логов.

Логи должны храниться не менее чем за пять дней работы балансировщика.

Логи балансировщика могут быть запрошены НТЦ Аргус в рамках аварийно-восстановительных и контраварийных мероприятий.

2.1.11.4 Требования к балансировщику нагрузки БД WFM CC

Сервер БД WFM CC может быть быть запущен двумя способами:

- с единственным сервером БД (балансировка не требуется). В данном случае БД является единой точкой отказа всей системы.
- с несколькими серверами БД, которые должны запускаться как отказоустойчивый кластер[29] (все сервера БД работают одновременно, один в в режиме ***master***, второй в режиме ***slave*** с опцией ***hot-standby***).

К основному (***master***) серверу направляются ***read-write*** запросы, к зависимому (***slave***) серверу - запросы только на чтение.

Для реализации Failover[30] используется решение в составе следующих программных компонентов: **Keepalived** - **Haproxy** - **Etcd** - **Patroni** (рис. 2.1.10.4), развернутых на Балансировщиках БД и на самих хостах БД.

---

[29] Для реализации кластера используется встроенный в PostgreSQL механизм асинхронной репликации

[30] В случае Failover (события смены ролей БД со slave на master) и перевода сервера БД из состояния UP в состояние DOWN должно быть высылаться оповещение по почте заранее определённому списку лиц.

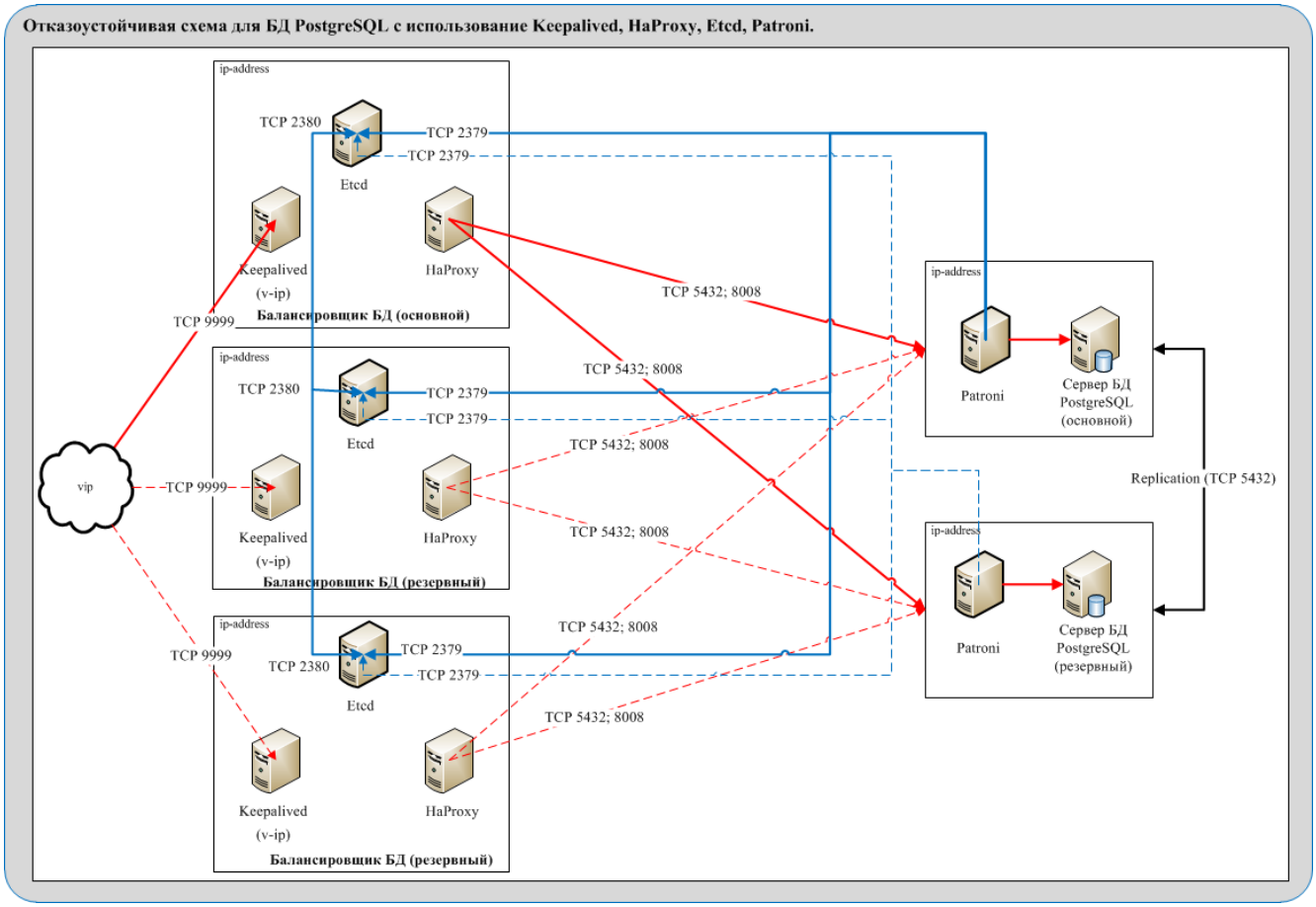


Рисунок 2.1.10.4 - Отказоустойчивое решение с механизмом Failover и балансировщиком нагрузки БД

На балансировщиках необходимо развернуть:

- **Keepalived** - используется для обеспечения наличия единой точки входа в кластер — виртуального IP.

  Keepalived – системный daemon на Linux системах, позволяющий организовать отказоустойчивость сервиса и балансировку нагрузки. Отказоустойчивость достигается за счет “плавающего» IP адреса, который переключается на резервный сервер в случае отказа основного. Для автоматического переключения IP адреса между серверами keepalived используется протокол VRRP.

  Keepalived выполняет две задачи:

  - Проверка состояния серверов;
  - Автоматическое переключение ресурсов в случае отказа сервера.

  Для защиты от падения haproxy daemon keepalived выполняют проверку, запуская раз в секунду команду «killall -0 haproxy». Она возвращает код 0 если процесс haproxy есть, и 1 если его нет.

  Если haproxy исчезнет, daemon keepalived просигналит аварию по vrrp, и снимет виртуальный ip. Виртуальный IP сразу же подхватит следующий по приоритету сервер, с живым haproxy.

- **Haproxy** - программный балансировщик. Необходим для отслеживания состояния серверов и перенаправления запросов на мастер сервер.

  Haproxy устанавливается на каждом хосте, и содержит в своем конфиге ссылки на все сервера PostgreSQL, проверяет какой сервер PostgreSQL сейчас является мастером, и отправляет запросы только на него.

  Для этой проверки в Patroni используется REST-интерфейс.

  С Haproxy идет проверка сервисов Patroni (server:8008 (где 8008 - порт по умолчанию)), в результате которой Patroni возвращает отчет по состоянию кластера в формате json, в котором содержится код ответа http: является ли данный сервер мастером (код 200) или нет (код 503).

  В соответствии с указанным ниже примером конфигурации:

  - haproxy слушает порт 9999 и перенаправляет трафик на мастер-сервер PostgreSQL
  - проверка статуса состояния сервера PostgreSQL (master|slave) происходит с интервалом в 1 секунду.
  - для перевода сервера PostgreSQL в down требуется 3 неудачных ответа (код 500), для переключения сервера PostgreSQL в up — 2 удачных ответа (кодом 200).

  В любой момент времени можно обратиться непосредственно на любой haproxy, и он корректно запроксирует трафик на мастер-сервер PostgreSQL.
- **Etcd**- это отказоустойчивое распределенное хранилище значений ключей, которое используется для хранения состояния кластера Postgres. С помощью него ноды Patroni определяют кто будет мастером. Необходимо использовать кластер с нечетным количеством серверов (в идеале иметь не меньше 3).

  Etcd устанавливается на балансировщики БД. Рекомендуется использовать не менее 3 балансировщиков с Etcd, т.к. у Etcd кворумная кластеризация (т.е. для выбора нового мастера необходимо N/2+1 живых нод).

На хостах БД необходимо развернуть (помимо базы PostgreSQL):

- **Patroni** - пакет Python с открытым исходным кодом, который управляет конфигурацией Postgres. Также отвечает за репликацию и Failover, и все настройки БД необходимо производить через него.

Для работы отказоустойчивого решения необходимо открыть порты согласно таблицы 2.1.10.4

Таблица 2.1.10.4 - Компоненты отказоустойчивого решения балансировщика БД и порты

Система источник	Система приемник	Порт	Назначение
etcd	etcd	2380	формирование кворума
patroni	etcd	2379	получение статуса кворума
haproxy	patroni	8008	проверка работоспособности сервиса
client	haproxy	7000	метрики состояния (работоспособности) haproxy

Система источник	Система приемник	Порт	Назначение
client	haproxy	9999	сетевой трафик
haproxy	postgresql	5432	сетевой трафик

2.1.12 Внешние IT-системы (интеграция)

Для взаимодействия с внешними системами предусмотрен выделенный Сервис интеграций

Также возможно взаимодействие внешних систем и компонентов решения WFM CC по протоколу SOAP/HTTP по выделенному IP-адресу и порту. напрямую или через балансировщик[31]

Конкретная реализация взаимодействия компонентов решения WFM CC и внешних систем определяется спецификой заказчика.

2.1.13 Система мониторинга

Для мониторинга компонентов, входящих в решение WFM CC используются:

- Система мониторинга Zabbix, при этом на хостах, подлежащих мониторингу, устанавливаются агенты, которые собирают метрики как с самих хостов, так и с сервисов, развернутых на хостах, пересылают метрики на сервера мониторинга, где они отображаются, анализируются и исходя из настроек высылаются в виде оповещений (алертов) о превышении пороговых значений (пороговые значения указаны в *п. 3.4.2 Мониторинг показателей*)
- Утилиты мониторинга (JVisualVM, JConcole, Command Line Interface, NMON).

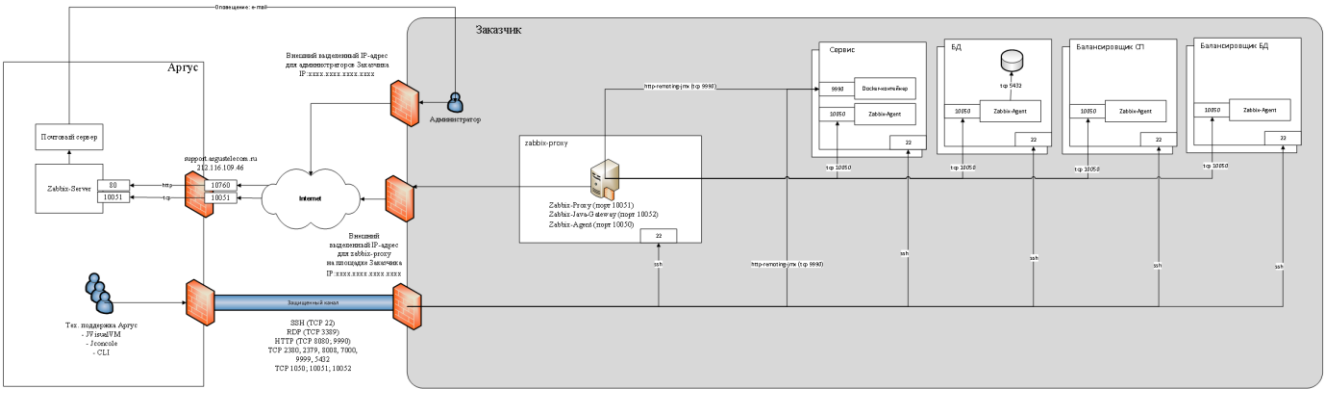


Рисунок 2.1.12 - Схема мониторинга

Состав системы мониторинга Zabbix

- Zabbix Server - сервер, отвечающий за работу с БД, сбор метрик и управление мониторингом и оповещением.
- Zabbix Proxy - сервер, который занимается промежуточным сбором и обработкой метрик и отправкой их на Zabbix Server.

Используется для повышения масштабируемости системы мониторинга и повышения отказоустойчивости.

Не имеет пользовательского интерфейса.

[31] В случае применения отказоустойчивого решения.

- Zabbix Java Gateway - аналог Zabbix Proxy для jmx-мониторинга (мониторинга СП).
- Zabbix Agent - предназначен для сбора и отправки данных на Zabbix Proxy/ Zabbix Server, выполнения (при необходимости) заранее определенных скриптов.

На схеме:

На стороне заказчика устанавливаются:

- Zabbix Proxy устанавливается совместно с Zabbix Java Gateway на одном хосте. Zabbix Proxy развертывается в активном режиме и является инициатором соединения с Zabbix Server - самостоятельно считывает информацию по конфигурации и передает данные мониторинга.
- Zabbix Agent развертывается на каждом хосте с компонента решения WFM CC

На стороне Аргус:

- Устанавливается Zabbix Server
- Осуществляется фильтрация входящего трафика по IP-адресу, предоставленному заказчиком, с которого осуществляется доступ к системе мониторинга.
- Размещаются утилиты мониторинга.

2.1.13.1 Требования к CPU, RAM Zabbix

- Zabbix Proxy + Zabbix Java Gateway

CPU: 1 (уровня AMD Athlon 3200+)

RAM: 2GB

HDD: 50GB

совместно с установкой Zabbix Proxy автоматически создаётся БД SQLite. Размер БД 10G

- Zabbix Agent

CPU: от 1 (уровня AMD Athlon 3200+)

RAM: от 256MB

HDD: от 10GB

2.1.13.2 Требования к портам Zabbix

У элементов системы мониторинга, используемые следующие стандартные порты:

- Zabbix Agent: 10050
- Zabbix Proxy: 10051
- Zabbix Java Gateway: 10052

Значения портов можно менять в конфигурационных файлах:

- zabbix_agentd.conf
- zabbix_proxy.conf
- zabbix_java_gateway.conf

2.2 Доступ для диагностики неисправностей для специалистов Аргус

Для проведения контраварийных, ремонтно-наладочных и других работ силами исполнителя, необходимо обеспечить доступ сотрудников исполнителя на тестовые и промышленные хосты серверов заказчика.

Доступ необходим для возможности чтения/скачивания логов для анализа; формирования дампов сервисов; проверки работы функционала ПО; формирования запросов в БД, анализирующих ее состояние; подключение к системам мониторинга и их настройка; возможности внесения изменений в конфигурационные файлы и перезапуска ПО компонентов решения WFM CC[32].

Для успешной диагностики неисправностей и проведение контраварийных работ необходим доступ специалистов исполнителя на хосты серверов заказчика, указанные в таблице 2.2

Таблица 2.2 - Доступ специалистов исполнителя к системам заказчика

Хост	Протокол	Порт
БД	TCP	5432
	SSH	22
Сервисы	HTTP/HTTPS	8080, 9990
	SSH	22
Балансировщик сервисов	HTTP/HTTPS	8080
	SSH	22[33]
Балансировщик БД	TCP	2380, 2379, 8008, 7000, 9999, 5432
	SSH	22
Система мониторинга[34]	TCP	5432, 10050, 10051, 10052
	SSH	22

У каждого сотрудника исполнителя, занятого технической поддержкой серверов приложений, должна быть своя УЗ (логин и пароль) для доступа на хост.

Под своей УЗ сотрудник может просматривать все содержимое каталога установки (обычно это каталог /argus), но не имеет прав для изменений в нем.

УЗ на хосте имеет домашний каталог, в который разрешается вносить изменения (создавать и редактировать файлы и каталоги).

---

[32] Выполняется по согласованию с заказчиком[2] В случае, если это программный балансировщик, например: Apache, Nginx

[33] В случае, если это программный балансировщик, например: Apache, Nginx

[34] В настоящий момент поддерживается система мониторинга: Zabbix

В зависимости от политики безопасности и технических возможностей заказчика, это может быть
- доступ к серверам заказчика напрямую из сети исполнителя по протоколам и портам, указанным в табл. 2.2
- доступ на терминальный сервер Заказчика из сети исполнителя по RDP 3389, а уже с него - доступ к серверам заказчика по протоколам и портам, указанным в табл. 2.2
По соображениям безопасности, некоторые из приведенных в таблице 2.2 протоколов/портов, могут быть закрыты[35].

2.3 Требования к квалификации обслуживающего персонала заказчика

Квалификация обслуживающего персонала заказчика должна соответствовать выполняемым функциям при эксплуатации решения WFM CC.

2.3.1 Эксплуатация БД

2.3.1.1 Обеспечение функционирования БД

Резервное копирование БД

- Запуск процедуры резервного копирования
- Мониторинг выполнения процедуры резервного копирования
- Контроль завершения процедуры резервного копирования

Восстановление БД

- Запуск процедуры восстановления БД
- Мониторинг выполнения процедуры восстановления БД
- Контроль завершения процедуры восстановления БД

Управление доступом к БД

- Назначение прав доступа пользователей к БД
- Изменение прав доступа пользователей к БД
- Контроль соблюдения прав доступа пользователей к БД

Установка и настройка ПО для обеспечения работы пользователей с БД

- Инсталляция ПО для поддержки работы пользователей с БД
- Настройка ПО для поддержки работы пользователей с БД
- Контроль результатов настройки ПО для поддержки работы пользователей с БД

Установка и настройка ПО для администрирования БД

- Инсталляция ПО для обеспечения работы администраторов с БД
- Настройка ПО для обеспечения работы администраторов с БД
- Контроль результатов настройки ПО для обеспечения работы администраторов с БД

Мониторинг событий, возникающих в процессе работы БД

- Наблюдение за работой БД
- Обнаружение отклонений от штатного режима работы БД
- Анализ отклонений от штатного режима работы БД и их устранение

[35] Для каждого заказчика индивидуально

Протоколирование событий, возникающих в процессе работы БД

- Фиксация отклонений от штатной работы БД
- Ведение журнала учета отклонений от штатной работы БД
- Информирование сотрудников, отвечающих за устранение отклонений от штатной работы БД

2.3.1.2 Оптимизация функционирования БД

Мониторинг работы БД, сбор статистической информации о работе БД

- Мониторинг работы БД, в том числе различными автоматизированными средствами
- Выбор основных статистических показателей работы БД
- Анализ полученных статистических данных, формирование выводов об эффективности работы БД

Оптимизация распределения вычислительных ресурсов, взаимодействующих с БД

- Анализ возможностей по управлению вычислительными ресурсами, взаимодействующими с БД
- Управление вычислительными ресурсами, взаимодействующими с БД
- Контроль результатов перераспределения вычислительных ресурсов, взаимодействующих с БД

Оптимизация производительности БД

- Анализ возможностей по управлению оптимизацией производительности БД
- Выбор критериев оптимизации производительности БД

Оптимизация компонентов вычислительной сети, взаимодействующих с БД

- Анализ компонентов вычислительной сети и возможностей по управлению их конфигурацией
- Выбор критериев оценки при изменении конфигурации компонентов вычислительной сети, взаимодействующих с БД
- Оптимизация компонентов вычислительной сети, взаимодействующих с БД, контроль произошедших изменений в работе БД

Оптимизация выполнения запросов к БД

- Статистический анализ запросов к БД, их классификация по различным признакам
- Выбор критериев оптимизации выполнения запросов к БД
- Оптимизация выполнения статистически значимых запросов к БД

Оптимизация управления жизненным циклом данных, хранящихся в БД

- Управление распределением данных в памяти
- Выбор стратегии управления распределением данных в памяти, предназначенной для размещения БД
- Контроль за соблюдением стратегии управления распределением данных в памяти, предназначенной для размещения БД

2.3.1.3 Предотвращение потерь и повреждений данных

Разработка регламентов резервного копирования БД

- Анализ функционирования прикладной системы с целью выявления подходящих временных интервалов для резервного копирования БД
- Выбор программных средств для выполнения резервного копирования
- Разработка и реализация сценария резервного копирования БД установленной прикладной системы
- Разработка сценариев по восстановлению БД в случае сбоев и подготовка соответствующей документации

Контроль выполнения регламента резервного копирования

- Корректировка действий при отклонении от регламента
- Сравнение выполняемых действий с регламентом резервного копирования

Разработка стратегии резервного копирования БД

- Изучение общих принципов выполнения резервного копирования
- Изучение архитектуры и графика эксплуатации прикладной системы

Разработка регламентов восстановления БД

- Выработка типовых сценариев восстановления БД при различных сбоях
- Анализ архитектуры прикладной системы с целью выявления наиболее подверженных сбоям компонентов БД

Разработка автоматических процедур для создания резервных копий БД

- Разработка скриптов для создания резервных копий БД
- Анализ характеристик программно-аппаратного обеспечения БД с точки зрения размещения резервных копий и производительности передачи данных

Проведение процедуры восстановления данных после сбоя

- Анализ возможных сбоев в работе БД и выработка сценариев мероприятий, необходимых для восстановления БД
- Написание скриптов по разработанным сценариям для быстрого устранения последствий сбоев

Контроль соблюдения регламента восстановления

- Корректировка действий при отклонении от регламента
- Сравнение выполняемых действий с регламентом восстановления БД

Анализ сбоев в работе БД и выявление их причин

- Мониторинг сбоев, возникающих в БД при обслуживании прикладной системы, и их документирование
- Выявление причин сбоев и своевременное их устранение
- Взаимодействие со службами технической поддержки БД и поставщиков компонентов вычислительного комплекса с целью локализации и устранения сбоев

Разработка методических инструкций по сопровождению БД

- Анализ основных этапов сопровождения БД
- Подготовка рекомендаций по сопровождению БД, включая оптимизацию критических процессов взаимодействия с БД

- Подготовка документации в соответствии с установленными правилами и требованиями

Мониторинг работы программно-аппаратного обеспечения БД

- Наблюдение за работой программно-аппаратного комплекса БД
- Фиксация отклонений от штатного режима работы БД

Настройка работы программно-аппаратного обеспечения БД

- Первоначальная установка программного обеспечения БД
- Применение результатов мониторинга БД для улучшения функционирования БД
- Настройка компонентов программно-аппаратного обеспечения БД для улучшения качества облуживания пользователей

Подготовка предложений по модернизации программно-аппаратных средств поддержки БД

- Анализ рынка программно-аппаратных средств поддержки БД
- Поиск путей модернизации, направленной на повышение эффективности работы БД
- Подготовка предложений по модернизации применяемых программно-аппаратных средств

Прогнозирование и оценка рисков сбоев в работе БД

- Анализ частоты сбоев различных типов в работе БД
- Поиск информации о сбоях и действиях по их устранению в различных источниках (в том числе в Интернете)
- Прогнозирование и оценка рисков сбоев в работе БД

Разработка автоматических процедур для горячего резервирования БД

- Первоначальная установка БД горячего резервирования
- Мониторинг БД горячего резервирования в прикладной системе
- Настройка и оптимизация работы пользователей БД горячего резервирования

Выполнение процедур по вводу в рабочий режим ресурсов горячей замены

- Установка обновлений ПО на узлах системы горячего резервирования БД
- Настройка автоматического ввода в рабочий режим БД горячего резерва в случае использования автоматики
- Переключение на БД горячего резервирования в случае необходимости

Подготовка отчетов о функционировании БД

- Сбор информации о работе БД
- Заполнение отчетных форм о состоянии и функционировании БД

2.3.1.4 Обеспечение информационной безопасности на уровне БД

Разработка политики информационной безопасности на уровне БД

- Анализ возможных угроз для безопасности данных
- Выбор основных средств поддержки информационной безопасности на уровне БД

Контроль соблюдения регламентов по обеспечению безопасности на уровне БД

- Выявление действий, нарушающих регламент обеспечения безопасности на уровне БД
- Корректировка действий при отклонении от регламента обеспечения безопасности на уровне БД

- Устранение последствий некорректных действий, ведущих к снижению информационной безопасности на уровне БД

Оптимизация работы систем безопасности с целью уменьшения нагрузки на работу БД

- Определение возможностей оптимизации работы систем безопасности с целью уменьшения нагрузки на работу БД
- Выбор наиболее эффективных путей снижения нагрузки при обеспечении заданного уровня безопасности данных на уровне БД

Разработка регламентов и аудит системы безопасности данных

- Выбор критериев оценки результатов аудита данных на уровне БД
- Разработка методик аудита системы безопасности данных на уровне БД
- Аудит системы безопасности и оценка ее эффективности

Подготовка отчетов о состоянии и эффективности системы безопасности на уровне БД

- Определение показателей и критериев эффективности системы безопасности, их расчет и анализ
- Оценка уровня и состояния системы безопасности данных на уровне БД

Разработка автоматизированных процедур выявления попыток несанкционированного доступа к данным

- Анализ возможностей программирования процедур для выявления попыток несанкционированного доступа к данным
- Применение средств программирования для разработки автоматизированных процедур выявления попыток несанкционированного доступа к данным

2.3.1.5 Управление развитием БД

Анализ системных проблем обработки информации на уровне БД, подготовка предложений по перспективному развитию БД

- Сбор и анализ нереализованных потребностей пользователей БД
- Исследование рынка перспективных БД, их принципиальных возможностей
- Подготовка плана реализации принятых решений по перспективному развитию БД

Разработка регламентов обновления версий программного обеспечения БД

- Анализ основных этапов обновления версий программного обеспечения БД
- Разработка и описание типовых процессов по обновлению версий БД
- Подготовка регламентирующих документов по обновлению версий БД

Разработка регламентов по миграции БД на новые платформы и новые версии ПО

- Анализ основных этапов миграции БД на новые платформы и новые версии ПО
- Разработка и описание типовых процессов миграции БД на новые платформы и новые версии ПО
- Подготовка регламентирующих документов по проведению миграции БД

Изучение, освоение и внедрение в практику администрирования новых технологий работы с БД

- Мониторинг новых информационных технологий в области БД, появляющихся на рынке
- Освоение и внедрение в практику администрирования новых технологий работы с БД

Контроль обновления версий БД

- Планирование этапов и анализ результатов выполнения каждого этапа обновления версий БД
- Планирование, проведение и анализ результатов проверки функционирования БД после обновления

Контроль миграции БД на новые платформы и новые версии ПО

- Планирование этапов миграции БД
- Анализ результатов тестирования работы БД после миграции
- Восстановление БД и корректировка действий при обнаружении ошибок миграции

2.3.2 Эксплуатация СП

2.3.2.1 Обеспечение функционирования СП

Установка и настройка ПО для обеспечения работы пользователей с СП

- Подготовка к инсталляции ПО СП, настройка ОС, сопутствующего ПО.
- Инсталляция ПО СП
- Настройка ПО СП
- Контроль результатов настройки ПО СП

Мониторинг работы СП, сбор статистической информации о работе СП

- Мониторинг работы СП, в том числе различными автоматизированными средствами
- Выбор основных статистических показателей работы СП
- Анализ полученных статистических данных, формирование выводов об эффективности работы СП
- Обнаружение отклонений от штатного режима работы СП
- Анализ отклонений от штатного режима работы СП и их устранение

Обслуживание СП

- Очистка логов СП, в том числе и автоматизированными средствами
- Формирование дампов памяти и дампов потоков (при аварии)
- Взаимодействие со службами технической поддержки и поставщиков ПО с целью локализации и устранения сбоев (при аварии)
- Перезапуск СП по рекомендации службы технической поддержки (при аварии или приближении к ней)

Протоколирование событий, возникающих в процессе работы СП

- Фиксация отклонений от штатной работы СП
- Ведение журнала учета отклонений от штатной работы СП
- Информирование сотрудников, отвечающих за устранение отклонений от штатной работы СП
- Подготовка отчетов о состоянии функционирования СП

2.3.2.2 Предотвращение потери сервиса СП

Резервное копирование СП

- Запуск процедуры резервного копирования

- Мониторинг выполнения процедуры резервного копирования
- Контроль завершения процедуры резервного копирования

Восстановление СП

- Запуск процедуры восстановления СП
- Мониторинг выполнения процедуры восстановления СП
- Контроль завершения процедуры восстановления СП

2.3.3 Эксплуатация клиентской системы

2.3.3.1 Установка и настройка клиентского ПО

- Проверка соответствия ресурсов ОС и клиентского ПО
- Инсталляция и обновление клиентского ПО
- Настройка клиентского ПО
- Контроль результатов установки, обновления и настройки клиентского ПО

2.3.4 Эксплуатация сетевой системы

2.3.4.1 Мониторинг и сбор статистической информации о работе сети

- Мониторинг работы сети, в том числе различными автоматизированными средствами
- Выбор основных статистических показателей работы сети
- Анализ полученных статистических данных, формирование выводов об эффективности работы сети
- Обнаружение отклонений от штатного режима работы сети
- Анализ отклонений от штатного режима работы сети и их устранение

2.3.4.2 Протоколирование событий, сетевых коллизий, возникающих в процессе работы

- Фиксация отклонений от штатной работы сети
- Ведение журнала учета отклонений от штатной работы сети
- Информирование сотрудников, отвечающих за устранение отклонений от штатной работы сети
- Подготовка отчетов о состоянии функционирования сети

2.3.4.3 Оптимизация компонентов вычислительной сети

- Анализ компонентов сети и возможностей по управлению их конфигурацией
- Выбор критериев оценки при изменении конфигурации компонентов сети, взаимодействующих компонентов КТС
- Оптимизация компонентов сети, взаимодействующих компонентов КТС, контроль произошедших изменений в работе сети

2.4 Общий порядок развертывания и обслуживания компонентов решения WFM CC

2.4.1 Типовые понятия

Дамп БД – состоит из описания структуры БД и/или содержащихся в ней данных, обычно в виде команд SQL. Используется для резервного копирования/восстановления данных.

Апдейт (update) БД – представляет из себя изменения внутренней структуры таблиц и объектов БД, добавление новых особенностей в функционал ПО.

Патч (patch) БД – представляет собой набор исправлений, выявленных в ходе тестирования, внедрения и опытной эксплуатации.

Дистрибутив сервера приложений – представляет собой установочный файл, выполняющий распаковку файлов сервера приложений и его конфигурирование.

Учётная запись – это совокупность имени пользователя и пароля, которые необходимо ввести при запуске программы. Каждая учётная запись сопоставлена соответствующему пользователю и включает в себя набор функциональных модулей и опций, назначенных этим модулям.

2.4.2 Типовые действия при обновлении ПО

Любые действия перед внесением изменений в ПО должны предваряться его резервной копией. Типовые действия при обновлении ПО включают в себя следующие этапы:

- оповещение пользователей о проводимых работах
- отключение пользовательских сессий
- создание резервных копий ПО
- установка ПО
- запуск ПО
- проверка работоспособности функционала, установленного ПО и анализ логов (при их наличии)
- возврат в обычный режим работы

При необходимости особых действий, инструкции по обновлению высылаются вместе с архивами ПО.

2.4.2.1 Обновление БД

- Остановка СП и Сервисов
- Выполнение резервной копии БД
- Выполнение обновлений БД (скрипты обновлений, dbmaintain)
- Запуск СП и Сервисов

2.4.2.2 Обновление СП и Сервисов

Обновление СП

- Остановка СП
- Резервное копирование каталога с СП
- Удаление каталога с СП (при переходе с версии на версию)
- Подготовка дистрибутива (распаковать и выложить в каталог, из которого будет происходить установка) и специфичных конфигураций СП (убедиться в том, что параметры указаны правильно)
- Установка СП
- Запуск СП

Обновление Сервисов

Механизм обновления сервисов реализуется средствами docker и docker-compose

- Остановка docker-контейнера
- Загрузка нового образа docker-контейнера (tar-файл) в локальный репозиторий docker[36]
- Настройка конфигурационного файла с параметрами запуска контейнера[37]
- Запуск docker-контейнера
- Удаление нового установочного образа (tar-файл)
- Удаление старого docker-контейнера
- Удаление старого образа docker-контейнера из локального репозитория docker

2.4.2.3 Обновление клиентского рабочего места

В зависимости от количества рабочих мест, это может быть обновление каждого рабочего места по отдельности или всех сразу при помощи групповых доменных политик[38] включает в себя удаление каталога с клиентским ПО и его установку заново.

2.4.3 Регулярные процедуры по обслуживанию компонентов решения WFM CC

Очистка логов

При работе всех компонентов решения WFM CC формируется значительное количество логов, пропорционально количеству пользователей, нагрузки и режима работы ПО: обычный или отладочный (debug).

В связи с этим, необходимо своевременно удалять (архивировать и удалять) логи, следя за тем, чтобы дисковое пространство не переполнялось. Переполнение дискового пространства приводит к отказу функционирования соответствующих компонентов решения WFM CC

Подробнее процедура очистки логов описана в п. *3.4.3 Архивация журналов операций*

Резервное копирование БД

Изложено в *п. 3.4.1.1 Резервное копирование БД*

Резервное копирование/восстановление СП WFM CC

При работе с СП возможно

- Создание резервной копии без остановки сервиса путем создания копии каталога установки СП.
- Восстановление СП возможно копированием в каталог установки сохраненной копии каталога СП. Требуется предварительная остановка СП.

Перезапуск СП WFM CC

- По рекомендации технической поддержки, при подходе к критическими значениям потребления ресурсов (CPU, RAM) - производить перезапуск СП.

Резервное копирование/восстановление Сервисов[39]

---

[36] репозиторий, расположенный на хосте, где будет запущен сервис
[37] docker-compose.yml
[38] Как правило, клиентское рабочее место работает под ОС Windows
[39] Перечень сервисов
- Сервис Личный кабинет WFM CC
- Сервис Мобильный API WFM CC
- Сервис планирования

Сервисы запускаются в docker-контейнерах, поэтому достаточно иметь резервную копию образа (tar-файл) для разворачивания docker-контейнера из которого в случае необходимости произвести переустановку сервиса, а также конфигурационные файлы[40].

Также возможно сохранение образа для docker-контейнера из локального репозитория docker-контейнеров.

Перезагрузка компонентов системы мониторинга Zabbix

- По рекомендации технической поддержки, при отсутствии метрик или реакции от компонентов системы мониторинга – произвести перезапуск компонентов Zabbix.

Инструкция по запуску и остановке компонентов приведена в *п. 3.2.6.5 Запуск и остановка компонентов Zabbix.*

2.4.4 Развертывание средств мониторинга

БД:

- настроить внутренние инструменты мониторинга: pgAdmin, sql-запросы;
- установить утилиту NMON и настроить сбор отчетов о состоянии ОС на хосте БД.
- установить стандартные утилиты ОС: top, mpstat, vmstat, iostat, и тому подобные - для сбора данных о ресурсах ОС.

СП и Сервисы

- для наблюдения за ресурсами СП и Сервисов в реальном времени следует использовать Admin Console или внешние утилиты мониторинга (JVisualVM, JConcole, Command Line Interface), для чего необходимо:

1. создать УЗ администратора СП и Сервисов для подключения утилитами мониторинга;
2. развернуть у себя на локальной хосте одну из внешних утилиты мониторинга (например, JVisualVM).
   - установить утилиту NMON и настроить сбор отчетов о состоянии ОС на хосте СП.
   - установить стандартные утилиты ОС: top, mpstat, vmstat, iostat, и тому подобные - для сбора данных о ресурсах ОС.

- Система мониторинга Zabbix[41] версии 4.2 необходима для мониторинга ресурсов ОС, СП и Сервисов, а также состояния БД, для чего необходимо:

- на хостах, подлежащих мониторингу, установить Zabbix-агенты[42], которые собирают метрики как с самих хостов, так и с сервисов, развернутых на хостах, пересылают метрики на Zabbix-прокси[43], откуда они пересылаются на Zabbix-сервер[44], где они

- Сервис отчетов
- Сервис уведомлений
- Сервис интеграций

[40] Обычно конфигурационные файлы расположены в каталоге /argus/services
- /argus/services/.env
- /argus/services/db.json
- /argus/services/docker-compose.yml

[41] Подробнее см. п.3.2.5 Система мониторинга и п. 2.1.12 Система мониторинга

[42] Установлен в сети Заказчика

[43] Установлен в сети Заказчика

[44] Установлен в сети Заказчика

отображаются, анализируются и исходя из настроек высылаются в виде оповещений о превышении пороговых значений (пороговые значения указаны в *п. 3.4.2. Мониторинг показателей*).

- развернуть Zabbix-прокси для пересылки данных от агентов на Zabbix-сервер.
- развернуть Zabbix JavaGateway для мониторинга показаний JVM СП и Сервисов.
- обратиться в техническую поддержку Исполнителя, чтобы получить:
    1. УЗ для просмотра данных собираемых системой мониторинга Zabbix.
    2. Jar-файл Zabbix JavaGateway с необходимыми настройками для мониторинга СП и Сервисов
- Создать адрес электронный почты, на который будут приходить оповещения о критичных показаниях систем подлежащих мониторингу.

Создать адрес электронный почты, на который будут приходить оповещения о критичных показаниях систем подлежащих мониторингу.

Организовать с помощью планировщика задач ОС архивирование/удаление устаревших логов компонентов Zabbix и утилит мониторинга.

2.4.5 Типовые действия при аварии

При аварии необходимо:

- Зафиксировать время аварии, возникающую при этом ошибку.
- Собрать все артефакты (дампы потоков, дампы памяти, скриншоты систем мониторинга, лог-файлы) – т.е. сведения о состоянии системы на момент аварии, которые позволяют выявить причину аварии и предпринять действия по предотвращению ее в будущем.
- Связаться со службой технической поддержки, предоставить все собранные артефакты и предпринять совместные контраварийные действия по локализации возникшей проблемы.

В первую очередь собирается оперативная (текущая) информация о системе, после – ретроспективная (историческая) информация.

БД

- активные сессии БД: время ожидания, тип ожидания, SQL-запросы
- дерево блокировок сессий
- состояние ОС (утилитами ОС)
- ошибки в postgresql_<date>.log
- данные систем мониторинга, скриншоты
- логи ОС (при необходимости)

СП

- состояние ОС (утилитами ОС)
- логи ОС (при необходимости)
- логи (со всех узлов СП если их несколько)
- состояние СП утилитой jvisualVM (скриншоты)
- данные систем мониторинга, скриншоты
- дампы памяти (heap), дампы потоков (threads) СП

Пример снятия дампа памяти (heap):

```
cd /Data/jboss_prod/bin./runjboss.sh heap-dump
```

Пример снятия дампа потоков (thread):

```
cd /Data/jboss_prod/bin./runjboss.sh thread-dump >> thread-dump_15-01-2016_15-23
```

Дампы потоков рекомендуется делать несколько раз с интервалом 3-5 мин. Файлы дампов будут созданы в каталоге ***bin*** СП.

Поскольку логи и дампы занимают значительный объем, рекомендуется архивировать их перед отправкой утилитами tar и gzip.

Пример:

```
cd /Data/jboss_prod/standalonetar -cvf log.tar loggzip log.tar
```

После архивации всех артефактов и отправки их Исполнителю, на хосте следует удалить уже неактуальные архивы tar и gz, а также дампы памяти и потоков.

3. Руководство по сервисному обслуживанию решения WFM CC

3.1 Настройка программной среды для развёртывания серверного ПО решения WFM CC

Перед проведением работ по развертыванию компонентов решения WFM CC, на каждой из ВМ должны быть проведены следующие предварительные\подготовительные действия:

- предоставлен доступ в интернет;
- обеспечена сетевая связанность между ВМ;
- открыты необходимые порты между ВМ согласно ТА;
- установлены утилиты и пакеты для диагностики состояния ОС: top, htop, vmstat, iostat, iotop, netstat, tcpdump, telnet, ping, mc;
- установлены Docker и Docker Compose на всех ВМ;
- установлены Oracle JDK версии 8 обновления 77 (8u77) на всех ВМ.

3.1.1 Сервер БД WFM CC

3.1.1.1 Организация каталогов

Рекомендуемая структура каталогов для размещения ПО БД WFM CC:

/argus	Каталог[45] содержит в себе БД и дополнительное окружение для поддержания его работы.
/argus/distr	Каталог содержит в себе дистрибутивы и установочные пакеты БД
/argus/nmon	Каталог, содержащий отчеты nmon о производительности системы (формат: ***<сетевое_имя_узла>_ггммдд_0101.nmon***) и их архивы[46] (формат: ***<сетевое_имя_узла>_ггммдд_0101.nmon.gz***).
/argus/scripts	Вспомогательные скрипты.
/argus/tmp	Каталог для временных файлов СП

Пример создания каталога

mkdir /argus
chown argus:argus /argus -R

3.1.1.2 Синхронизация времени

В ОС должна быть установлена служба времени (ntpd или chronyd), обеспечивающая синхронизацию системного времени с сервера времени.

[45] Для размещения рекомендуется выделять отдельный раздел диска.
[46] После архивации отчеты удаляются.

3.1.2. СП WFM CC

3.1.2.1 Учетная запись

В ОС должна быть создана учетная запись ***argus*** для установки и запуска как СП, так и любого другого дополнительного ПО (например, JDK). Учетная запись должна обладать правами на запись в каталог СП (например, ***/argus/jboss_prod***) и его подкаталоги.

Создание новой учетной записи в ОС Linux производится утилитой ***adduser***.

Пример

```
adduser argus
```

С более подробной инструкцией по созданию учетной записи через командную строку можно ознакомиться на странице: https://access.redhat.com/documentation/en-us/red_hat_enterprise_linux/7/html/system_administrators_guide/ch-managing_users_and_groups

3.1.2.2 Java-машина

Для стабильной работы СП WFM CC необходим предустановленный пакет Oracle JDK версии 8 обновления 77 (8u77).
Java версии 8 доступна для скачивания со страницы:

http://www.oracle.com/technetwork/java/javase/downloads/index.html.

Загрузка JDK 8u77 из архива сайта Oracle требует наличия зарегистрированной учетной записи.

Рекомендуется устанавливать Java-машину в каталог ***/argus/jdk/jdk1.8.0_77*** как JDK (java development kit), а не как JRE (java runtime environment), т.к. JDK предоставляет дополнительные инструменты для диагностики приложения.

Необходимо указать в ***.bash_profile*** (или ***.profile***, в зависимости от того, какой файл используется в ОС) пользователя ОС, запускающего установочный файл СП или сам процесс СП: переменную окружения ***JAVA_HOME***, в которой требуется прописать путь до JDK

Например:

export JAVA_HOME=/argus/jdk/jdk1.8.0_77

в переменной окружения ***PATH*** указать путь до ***bin*** каталога ***JDK***

Например

export PATH=/argus/jdk/jdk1.8.0_77/bin:$PATH

Желательно для установки брать последнюю Java-машину, в которой учитывается отсутствие перехода на зимнее время в Российской Федерации

или же имеющаяся Java-машина должна быть обновлена с помощью ***Java Time Zone Updater***. Дополнительную информацию см.http://www.oracle.com/technetwork/java/javase/tzupdater-readme-136440.html

3.1.2.3 Данные часового пояса

Хост должен быть в UTC-timezone.

Проверка таймзон, установленных на хосте

```
timedatectl list-timezones
```

Установка таймзоны UTC

```
timedatectl set-timezone UTC
```

Перед установкой СП необходимо убедиться, что в JDK установлены актуальные данные часового пояса.

3.1.2.4 Локаль и кодировка операционной системы

Операционная система должна поддерживать кодировку UTF-8 и в ней должны быть установлены форматы времени, даты и чисел (***locale***) в соответствии с используемыми пользователями языковыми стандартами (при установке в Российской Федерации нужна локаль ***ru_RU.UTF-8***).

В операционных системах Linux проверка локали осуществляется командой ***locale***. Ниже показан вывод команды на правильно настроенном сервере:

```
$ locale
LANG=ru_RU.UTF-8
LC_CTYPE="ru_RU.UTF-8"
LC_NUMERIC="ru_RU.UTF-8"
LC_TIME="ru_RU.UTF-8"
LC_COLLATE="ru_RU.UTF-8"
LC_MONETARY="ru_RU.UTF-8"
LC_MESSAGES="ru_RU.UTF-8"
LC_PAPER="ru_RU.UTF-8"
LC_NAME="ru_RU.UTF-8"
LC_ADDRESS="ru_RU.UTF-8"
LC_TELEPHONE="ru_RU.UTF-8"
LC_MEASUREMENT="ru_RU.UTF-8"
LC_IDENTIFICATION="ru_RU.UTF-8"
LC_ALL=
```

3.1.2.5 Синхронизация времени

Для корректной работы СП WFM CC требуется, чтобы была обеспечена синхронизация времени ОС СП WFM CC и ОС, на которой установлена БД WFM CC.

В ОС должна быть установлена служба времени (ntpd или chronyd), обеспечивающая синхронизацию системного времени с сервера времени.

3.1.2.6 Максимальное число открытых файлов и сокетов

Необходимо увеличить ограничение на максимальное число открытых файлов и сокетов по сравнению со значениями по умолчанию. Сервер удерживает открытыми множество файлов. Сервер приложений открывает сокеты, устанавливая исходящие сетевые соединения к БД WFM CC и к СП других компонентов решения WFM CC, с которыми поддерживается взаимодействие. Сервер приложений открывает сокеты, принимая входящие соединения от браузеров пользователей. Требуемое максимальное число открытых файлов и сокетов рассчитывается по формуле:

$$max_{открытых\ сокетов\ и\ файлов} = max_{конкурентных\ пользователей} * 20$$

Linux

Параметры настройки ядра.

Посмотреть текущие настройки уровня ОС:

```
cat /proc/sys/fs/file-max
```

Установить значение параметра уровня ОС в файле:

```
cat /proc/sys/fs/file-max
```

Значение:

```
sysctl -w fs.file-max=102400
```

После изменения параметров – активировать изменения в ядре без перезагрузки хоста:

```
sysctl -p /etc/sysctl.conf
```

Посмотреть настройки для текущего пользователя (УЗ) ОС:

```
ulimit -n
```

Установить значение для УЗ в файле:

```
/etc/security/limits.conf
```

Значение:

```
argus soft nofile  100000
argus hard nofile 100000
```

3.1.2.7 Максимальное число запущенных процессов

Необходимо увеличить ограничение на максимальное число запущенных процессов по сравнению со значениями по умолчанию. При работе сервера без дополнительных http-портов, количество потоков сервера может находиться в пределах 200-1000, а добавление очередного дополнительного http-порта влечет создание пула на 256 потоков. Таким образом, в ОС у пользователя, под которым запускается сервер приложений, следует увеличить максимальное число запущенных процессов (*max-user-processes*). Значение параметра рассчитывается по формуле:

***max-user-processes* ≥ 1000 + *additional-ports-count* * 256**

где *additional-ports-count* - количество дополнительных http-портов (*argus.io.additional-http-ports*)

Посмотреть настройки для текущего пользователя (УЗ) ОС:

```
ulimit -a | grep processes
```

Установить значение для УЗ в файле:

```
/etc/security/limits.conf
```

Значения:

```
argus soft nproc 4000
argus hard nproc 4000
```

3.1.2.8 Дополнительные рекомендуемые настройки ОС

Linux

Необходимо увеличить значение ***socket send/receive buffer*** по сравнению со значениями по умолчанию в файле:

```
/etc/sysctl.conf
```

Значения:

```
default socket receive buffer:              sysctl -w net.core.rmem_default=262144
default socket send buffer:                  sysctl -w net.core.wmem_default=262144
max socket receive buffer:                  sysctl -w net.core.rmem_max=262144
max socket send buffer size:               sysctl -w net.core.wmem_max=262144
```

После изменения параметров – активировать изменения в ядре без перезагрузки хоста:

```
sysctl -p /etc/sysctl.conf
```

3.1.2.9 Организация каталогов

Рекомендуемая структура каталогов для размещения ПО СП WFM CC:

/argus	Каталог содержит в себе СП и дополнительное окружение для поддержания его работы. Для размещения рекомендуется выделять отдельный раздел диска.
/argus/distr	Каталог содержит в себе дистрибутивы и пакеты установок СП и дополнительного ПО. Например: /argus/distr/1450/argus-dist-3.14.0.16954-customer.jar /argus/distr/1450/ customer.prod.argus-app-01.properties /argus/distr/jdk_8u77_linux_x64.gz

/argus/jboss_arch	Каталог хранит в себе резервные копии СП (формат копии: ддммгггг/jboss_prod). Например: /argus/jboss_arch/16012016/jboss_prod
/argus/jboss_prod	Каталог установки СП
/argus/jdk	Каталог с установленной JDK.
/argus/jdk/jdk1.8.0_77	JDK версии 8 обновления 77. Требуется для работы СП
/argus/nmon	Каталог, содержащий отчеты nmon о производительности системы (формат: ***<сетевое_имя_узла>_ггммдд_0101.nmon***) и их архивы[47] (формат: ***<сетевое_имя_узла>_ггммдд_0101.nmon.gz***).
/argus/scripts	Вспомогательные скрипты.
/argus/tmp	Каталог для временных файлов СП

Пример создания каталога

mkdir /argus
chown argus:argus /argus -R

3.1.2.10 Имя хоста

Имя хоста не должно содержать символ подчеркивания.

Проверка:

```
uname -n
```

3.1.2.11 Требования к установленным шрифтам

Для корректной работы сервера приложений, на ОС должен быть установлен пакет **fontconfig** с **TrueType-шрифтами**.

3.1.3 Сервис Личный кабинет WFM CC

3.1.3.1 Docker и Docker Compose

Для установки ПО Docker необходима операционная система CentOS с версией ядра не ниже 3.10.0-229.el7.x86_64.

Для компонентов, поставляемых контейнере Docker, на хосте предварительно должны быть установлено ПО

- Docker не ниже 19.03.12
- Docker Compose не ниже 1.29.2

[47] После архивации отчеты удаляются.

Команды проверки наличия Docker и Docker Compose:

```
docker -v
```

```
docker-compose -v
```

Если соответствующее ПО установлено, то будет выдано сообщение о версии и номере сборки.

3.1.3.2 Учетная запись

В ОС должна быть создана учетная запись argus для установки и запуска контейнера Docker. Учетная запись должна входить в группу **docker** с **uid/gid 1099**

Создание новой учетной записи в ОС Linux производится утилитой **adduser**.

Пример

```
adduser argus
```

```
usermod -aG docker argus
groupmod -g 1099 argus
usermod -u 1099 -g 1099 argus
```

Проверка создания УЗ

```
id argus
```

#Вывод в консоли результата проверки id argus

uid=1099(argus) gid=1099(argus) groups=1099(argus),995(docker)

С более подробной инструкцией по созданию учетной записи через командную строку можно ознакомиться на странице: https://access.redhat.com/documentation/en-us/red_hat_enterprise_linux/7/html/system_administrators_guide/ch-managing_users_and_groups

3.1.3.3 Организация каталогов

Рекомендуемая структура каталогов на хосте:

/argus	Каталог[48] содержит в себе дистрибутивы, конфигурационные файлы, ПО в виде образов Docker и дополнительное окружение для поддержания работы сервиса.
/argus/distr	Каталог содержит в себе дистрибутивы поставляемого ПО в виде образов Docker (tar-файлы)
/argus/nmon	Каталог, содержащий отчеты nmon о производительности системы (формат: ***<сетевое_имя_узла>_ггммдд_0101.nmon***) и их архивы[49][2] (формат: ***<сетевое_имя_узла>_ггммдд_0101.nmon.gz***).
/argus/scripts	Вспомогательные скрипты.

Пример создания каталога

[48] Для размещения рекомендуется выделять отдельный раздел диска.
[49] После архивации отчеты удаляются.

```
mkdir /argus
```

```
chown argus:argus /argus -R
```

3.1.3.4 Синхронизация времени

В ОС должна быть установлена служба времени (ntpd или chronyd), обеспечивающая синхронизацию системного времени с сервера времени.

3.1.4 Сервис Мобильный API WFM CC

3.1.4.1 Docker и Docker Compose

Для установки ПО Docker необходима операционная система CentOS с версией ядра не ниже 3.10.0-229.el7.x86_64.

Для компонентов, поставляемых контейнере Docker, на хосте предварительно должны быть установлено ПО

- Docker не ниже 19.03.12
- Docker Compose не ниже 1.29.2

Команды проверки наличия Docker и Docker Compose:

```
docker -v
```

```
docker-compose -v
```

Если соответствующее ПО установлено, то будет выдано сообщение о версии и номере сборки.

3.1.4.2 Учетная запись

В ОС должна быть создана учетная запись argus для установки и запуска контейнера Docker. Учетная запись должна входить в группу **docker** с **uid/gid 1099**

Создание новой учетной записи в ОС Linux производится утилитой **adduser**.

Пример

```
adduser argus
```

```
usermod -aG docker argus
groupmod -g 1099 argus
usermod -u 1099 -g 1099 argus
```

chown argus:argus /argus -R

Проверка создания УЗ

```
id argus
```

#Вывод консоли после проверки id argus

uid=1099(argus) gid=1099(argus) groups=1099(argus),995(docker)

С более подробной инструкцией по созданию учетной записи через командную строку можно ознакомиться на странице: https://access.redhat.com/documentation/en-us/red_hat_enterprise_linux/7/html/system_administrators_guide/ch-managing_users_and_groups

3.1.4.3 Организация каталогов

Рекомендуемая структура каталогов на хосте:

/argus	Каталог[50] содержит в себе дистрибутивы, конфигурационные файлы, ПО в виде образов Docker и дополнительное окружение для поддержания работы сервиса.
/argus/distr	Каталог содержит в себе дистрибутивы поставляемого ПО в виде образов Docker (tar-файлы)
/argus/nmon	Каталог, содержащий отчеты nmon о производительности системы (формат: ***<сетевое_имя_узла>_ггммдд_0101.nmon***) и их архивы[51][2] (формат: ***<сетевое_имя_узла>_ггммдд_0101.nmon.gz***).
/argus/scripts	Вспомогательные скрипты.

Пример создания каталога

```
mkdir /argus
```

```
chown argus:argus /argus -R
```

3.1.4.4 Синхронизация времени

В ОС должна быть установлена служба времени (ntpd или chronyd), обеспечивающая синхронизацию системного времени с сервера времени.

3.1.5 Сервис планирования

3.1.5.1 Docker и Docker Compose

Для установки ПО Docker необходима операционная система CentOS с версией ядра не ниже 3.10.0-229.el7.x86_64.

Для компонентов, поставляемых контейнере Docker, на хосте предварительно должны быть установлено ПО

- Docker не ниже 19.03.12
- Docker Compose не ниже 1.29.2

Команды проверки наличия Docker и Docker Compose:

```
docker -v
```

```
docker-compose -v
```

Если соответствующее ПО установлено, то будет выдано сообщение о версии и номере сборки.

[50] Для размещения рекомендуется выделять отдельный раздел диска.
[51] После архивации отчеты удаляются.

3.1.5.2 Учетная запись

В ОС должна быть создана учетная запись argus для установки и запуска контейнера Docker. Учетная запись должна входить в группу **docker** с **uid/gid 1099**

Создание новой учетной записи в ОС Linux производится утилитой **useradd**.

Пример

```
adduser argus
```

```
usermod -aG docker argus
groupmod -g 1099 argus
usermod -u 1099 -g 1099 argus
```

chown argus:argus /argus -R

Проверка создания УЗ

```
id argus
```

#Вывод консоли после проверки id argus

uid=1099(argus) gid=1099(argus) groups=1099(argus),995(docker)

С более подробной инструкцией по созданию учетной записи через командную строку можно ознакомиться на странице: https://access.redhat.com/documentation/en-us/red_hat_enterprise_linux/7/html/system_administrators_guide/ch-managing_users_and_groups

3.1.5.3 Организация каталогов

Рекомендуемая структура каталогов на хосте:

/argus	Каталог[52] содержит в себе дистрибутивы, конфигурационные файлы, ПО в виде образов Docker и дополнительное окружение для поддержания работы сервиса.
/argus/distr	Каталог содержит в себе дистрибутивы поставляемого ПО в виде образов Docker (tar-файлы)
/argus/nmon	Каталог, содержащий отчеты nmon о производительности системы (формат: ***<сетевое_имя_узла>_ггммдд_0101.nmon***) и их архивы[53] (формат: ***<сетевое_имя_узла>_ггммдд_0101.nmon.gz***).
/argus/scripts	Вспомогательные скрипты.

Пример создания каталога

```
mkdir /argus
```

```
chown argus:argus /argus -R
```

[52] Для размещения рекомендуется выделять отдельный раздел диска.
[53] После архивации отчеты удаляются.

3.1.5.4 Синхронизация времени

В ОС должна быть установлена служба времени (ntpd или chronyd), обеспечивающая синхронизацию системного времени с сервера времени.

3.1.6 Сервис отчетов

3.1.6.1 Docker и Docker Compose

Для установки ПО Docker необходима операционная система CentOS с версией ядра не ниже 3.10.0-229.el7.x86_64.

Для компонентов, поставляемых контейнере Docker, на хосте предварительно должны быть установлено ПО

- Docker не ниже 19.03.12
- Docker Compose не ниже 1.29.2

Команды проверки наличия Docker и Docker Compose:

```
docker -v
```

```
docker-compose -v
```

Если соответствующее ПО установлено, то будет выдано сообщение о версии и номере сборки.

3.1.6.2 Учетная запись

В ОС должна быть создана учетная запись argus для установки и запуска контейнера Docker. Учетная запись должна входить в группу **docker** с **uid/gid 1099**

Создание новой учетной записи в ОС Linux производится утилитой **useradd**.

Пример

```
adduser argus
```

```
usermod -aG docker argus
groupmod -g 1099 argus
usermod -u 1099 -g 1099 argus
```

chown argus:argus /argus -R

Проверка создания УЗ

```
id argus
```

#Вывод консоли после проверки id argus

uid=1099(argus) gid=1099(argus) groups=1099(argus),995(docker)

С более подробной инструкцией по созданию учетной записи через командную строку можно ознакомиться на странице: https://access.redhat.com/documentation/en-us/red_hat_enterprise_linux/7/html/system_administrators_guide/ch-managing_users_and_groups

3.1.6.3 Организация каталогов

Рекомендуемая структура каталогов на хосте:

/argus	Каталог[54] содержит в себе дистрибутивы, конфигурационные файлы, ПО в виде образов Docker и дополнительное окружение для поддержания работы сервиса.
/argus/distr	Каталог содержит в себе дистрибутивы поставляемого ПО в виде образов Docker (tar-файлы)
/argus/nmon	Каталог, содержащий отчеты nmon о производительности системы (формат: ***<сетевое_имя_узла>_ггммдд_0101.nmon***) и их архивы[55](формат: ***<сетевое_имя_узла>_ггммдд_0101.nmon.gz***).
/argus/scripts	Вспомогательные скрипты.

Пример создания каталога

```
mkdir /argus
```

```
chown argus:argus /argus -R
```

3.1.6.4 Синхронизация времени

В ОС должна быть установлена служба времени (ntpd или chronyd), обеспечивающая синхронизацию системного времени с сервера времени.

3.1.7 Сервис уведомлений

3.1.7.1 Docker и Docker Compose

Для установки ПО Docker необходима операционная система CentOS с версией ядра не ниже 3.10.0-229.el7.x86_64.

Для компонентов, поставляемых контейнере Docker, на хосте предварительно должны быть установлено ПО

- Docker не ниже 19.03.12
- Docker Compose не ниже 1.29.2

Команды проверки наличия Docker и Docker Compose:

```
docker -v
```

```
docker-compose -v
```

Если соответствующее ПО установлено, то будет выдано сообщение о версии и номере сборки.

3.1.7.2 Учетная запись

В ОС должна быть создана учетная запись argus для установки и запуска контейнера Docker. Учетная запись должна входить в группу **docker** с **uid/gid 1099**

Создание новой учетной записи в ОС Linux производится утилитой **useradd**.

Пример

[54] Для размещения рекомендуется выделять отдельный раздел диска.
[55] После архивации отчеты удаляются.

```
adduser argus
```

```
usermod -aG docker argus
groupmod -g 1099 argus
usermod -u 1099 -g 1099 argus
```

chown argus:argus /argus -R

Проверка создания УЗ

```
id argus
```

#Вывод консоли после проверки id argus

uid=1099(argus) gid=1099(argus) groups=1099(argus),995(docker)

С более подробной инструкцией по созданию учетной записи через командную строку можно ознакомиться на странице: https://access.redhat.com/documentation/en-us/red_hat_enterprise_linux/7/html/system_administrators_guide/ch-managing_users_and_groups

3.1.7.3 Организация каталогов

Рекомендуемая структура каталогов на хосте:

/argus	Каталог[56] содержит в себе дистрибутивы, конфигурационные файлы, ПО в виде образов Docker и дополнительное окружение для поддержания работы сервиса.
/argus/distr	Каталог содержит в себе дистрибутивы поставляемого ПО в виде образов Docker (tar-файлы)
/argus/nmon	Каталог, содержащий отчеты nmon о производительности системы (формат: ***<сетевое_имя_узла>_ггммдд_0101.nmon***) и их архивы[57] (формат: ***<сетевое_имя_узла>_ггммдд_0101.nmon.gz***).
/argus/scripts	Вспомогательные скрипты.

Пример создания каталога

```
mkdir /argus
```

```
chown argus:argus /argus -R
```

3.1.7.4 Синхронизация времени

В ОС должна быть установлена служба времени (ntpd или chronyd), обеспечивающая синхронизацию системного времени с сервера времени.

[56] Для размещения рекомендуется выделять отдельный раздел диска.
[57] После архивации отчеты удаляются.

3.1.8 Сервис интеграций

3.1.8.1 Java-машина

для работы приложения требуется OpenJDK версии 1.8[58] [1]

для ее установки необходимо выполнить:

```
yum install java-1.8.0-openjdk.x86_64
```

Проверить версию java:

```
java -version
```

Пример вывода:

```
openjdk version "1.8.0_232"
OpenJDK Runtime Environment (build 1.8.0_232-b09)
OpenJDK 64-Bit Server VM (build 25.232-b09, mixed mode)
```

3.1.8.2 Учетная запись

В ОС должна быть создана учетная запись **argus** для установки и управления ПО

Создание новой учетной записи в ОС Linux производится утилитой **useradd**.

пример

Создать пользователя argus

```
useradd argus
```

Задать пароль пользователю argus

```
passwd argus
```

С более подробной инструкцией по созданию учетной записи через командную строку можно ознакомиться на странице: https://access.redhat.com/documentation/en-us/red_hat_enterprise_linux/7/html/system_administrators_guide/ch-managing_users_and_groups

3.1.8.3 Организация каталогов

Рекомендуемая структура каталогов на хосте:

/argus	Каталог[59] содержит в себе дистрибутивы, конфигурационные файлы, резервные копии дистрибутивов и вспомогательные скрипты и утилиты
/argus/integration	Каталог содержит в себе приложение
/argus/distr	Каталог содержит в себе дистрибутивы поставляемого ПО в виде архивов

[58] Текущая версия в репозитории 1.8.0_232

[59] Для размещения рекомендуется выделять отдельный раздел диска.

/argus/backup	Каталог содержит в себе резервные копии, формируемые перед обновлением ПО
/argus/nmon	Каталог, содержащий отчеты nmon о производительности системы (формат: ***<сетевое_имя_узла>_ггммдд_0101.nmon***) и их архивы[60] (формат: ***<сетевое_имя_узла>_ггммдд_0101.nmon.gz***).
/argus/scripts	Вспомогательные скрипты.

Пример создания каталога

```
mkdir -p /argus/integration
```

```
chown argus:argus /argus -R
```

3.1.8.4 Настройка автозапуска

Создать файл **/etc/systemd/system/integration.service** содержащий

```
[Unit]
Description=integration
After=syslog.target


[Service]
User=argus
WorkingDirectory=/argus/integration
ExecStart=/argus/integration/integration-0.0.46-SNAPSHOT.jar
SuccessExitStatus=143
TimeoutStopSec=10
Restart=on-failure
RestartSec=5
OOMScoreAdjust=-1000


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

Выполнить

```
systemctl daemon-reload
systemctl enable integration.service
```

3.1.8.5 Синхронизация времени

В ОС должна быть установлена служба времени (ntpd или chronyd), обеспечивающая синхронизацию системного времени с сервера времени.

3.1.9 Сервис мониторинга

3.1.9.1 Docker и Docker Compose

Для установки ПО Docker необходима операционная система CentOS с версией ядра не ниже 3.10.0-229.el7.x86_64.

[60] После архивации отчеты удаляются.

Для компонентов, поставляемых контейнере Docker, на хосте предварительно должны быть установлено ПО

- Docker не ниже 19.03.12
- Docker Compose не ниже 1.29.2

Команды проверки наличия Docker и Docker Compose:

```
docker -v
docker-compose -v
```

Если соответствующее ПО установлено, то будет выдано сообщение о версии и номере сборки.

3.1.9.2 Учетная запись

В ОС должна быть создана учетная запись argus для установки и запуска контейнера Docker. Учетная запись должна входить в группу **docker с uid/gid 1099**

Создание новой учетной записи в ОС Linux производится утилитой useradd.

Пример

```
adduser argus
```

```
usermod -aG docker argus
groupmod -g 1099 argus
usermod -u 1099 -g 1099 argus
```

chown argus:argus /argus -R

Проверка создания УЗ

```
id argus
```

#Вывод консоли после проверки id argus

```
uid=1099(argus) gid=1099(argus) groups=1099(argus),995(docker)
```

С более подробной инструкцией по созданию учетной записи через командную строку можно ознакомиться на странице: https://access.redhat.com/documentation/en-us/red_hat_enterprise_linux/7/html/system_administrators_guide/ch-managing_users_and_groups

3.1.9.3 Организация каталогов

Рекомендуемая структура каталогов на хосте:

/argus	Каталог[61] содержит в себе дистрибутивы, конфигурационные файлы, ПО в виде образов Docker и дополнительное окружение для поддержания работы сервиса.
/argus/images	Каталог содержит в себе дистрибутивы поставляемого ПО в виде образов Docker (tar-файлы)
/argus/nmon	Каталог, содержащий отчеты nmon о производительности системы (формат: <сетевое_имя_узла>_ггммдд_0101.nmon) и их архивы[62] (формат: <сетевое_имя_узла>_ггммдд_0101.nmon.gz).
/argus/scripts	Вспомогательные скрипты.

[61] Для размещения рекомендуется выделять отдельный раздел диска.
[62] После архивации отчеты удаляются.

/argus/services	Каталог содержит конфигурационные файлы docker-compose.yml и .env
/argus/monitoring-data-endpoint/logs	Каталог содержит логи контейнера monitoring-data-endpoint
/argus/monitoring-data-endpoint/logs/gcstats	Каталог содержит логи сборщика мусора (GC stats) для контейнера monitoring-data-endpoint

Пример создания каталога

```
mkdir /argus
```

```
chown argus:argus /argus -R
```

3.1.9.4 Синхронизация времени

В ОС должна быть установлена служба времени (ntpd или chronyd), обеспечивающая синхронизацию системного времени с сервера времени.

3.1.9.5 Настройка часового пояса

Часовой пояс (timezone) должен быть одинаков для всех ВМ решения WFM CC

Пример настройки:

```
timedatectl set-timezone Europe/Moscow
```

Текущая таймзона:

```
timedatectl
```

Доступные таймзоны:

```
timedatectl list-timezones
```

3.1.10 Балансировщик СП

ПО решения WFM CC работает с балансировщиком, который применяется для балансировки нагрузки при обращении к сервисам и СП, а также используется как ***reverse proxy*** при обращении к сервисам и СП по HTTPS.

Требования к настройкам балансировщика изложены в п. 2.1.10.3 Требования к балансировщику нагрузки сервисов WFM CC

Используемые программные балансировщики при работе ПО Аргус:

- На базе веб-сервера ***apache*** httpd (https://httpd.apache.org/download.cgi)
  - с установленным ***mod_jk*** (http://tomcat.apache.org/download-connectors.cgi);
  - с установленным ***mod_cluster*** (http://mod-cluster.jboss.org/downloads);
  - с включенным ***mod_proxy_balancer*** (обычно поставляется с httpd).
- На базе программного балансировщика ***HAProxy***

Перед установкой и настройкой ***HAproxy*** необходимо установить репозиторий epel:

```
yum install epel-release
```

Используемые аппаратные балансировщики при работе ПО Аргус[63]:

- CISCO ACE 10
- Citrix NetScaler

3.1.10.1 Синхронизация времени

В ОС должна быть установлена служба времени (ntpd или chronyd), обеспечивающая синхронизацию системного времени с сервера времени.

3.1.11 Балансировщик БД

Компоненты балансировщика БД устанавливаются из стандартных пакетов.

Должен быть предоставлен доступ в интернет к репозиториям для обновления и установки пакетов

Подробнее об установке см.пп.

- 3.2.4.1.1 Установка Keepalived
- 3.2.4.2.1 Установка Haproxy
- 3.2.4.3.1 Установка Etcd
- 3.2.4.4.1 Установка Patroni

3.1.11.1 Синхронизация времени

В ОС должна быть установлена служба времени (ntpd или chronyd), обеспечивающая синхронизацию системного времени с сервера времени.

3.1.12 Средства мониторинга

3.1.12.1 Настройка программной среды для мониторинга СУБД

Настройка программной среды СУБД приведена в п. 3.1.1 Сервер БД WFM CC

3.1.12.2 Настройка программной среды для мониторинга СП

На хостах, с которых планируется удаленый мониторинг ресурсов СП, требуется подготовить программную среду для работы утилит:

- *Java Visual VM (**JVisualVM**) - фреймворк диагностики, позволяющий в реальном времени оценивать, а также сохранять в форме отчетов информацию о состоянии потоков сервера, создаваемой нагрузке на ОС, параметрах JMX-бинов, обнаруживать блокировки потоков и др. Позволяет одновременно работать со множеством серверов. Плагины к **JVisualVM** существенно расширяют возможности фреймворка, добавляя, например, функционал для работы с jmx-бинами, просмотра подробных данных по сборке мусора и пр. **JViusalVM** не входит в состав сервера приложений, но включается в некоторые JVM. Также расширенный вариант **JVisualVM** доступен на сайте разработки, http://visualvm.java.net/index.html. Там же содержится подробная документация по фреймворку и инструкции по разработке плагинов.*
- ***JConsole*** является одним из наиболее мощных диагностических инструментов, позволяющих в реальном времени получать информацию об используемых ресурсах ОС, созданных внутри процесса JVM потоках, загруженных классах, получить или изменить состояние JMX-бинов, а

[63] Список аппаратных балансировщиков не ограничен лишь приведенными моделями.

также многие другие возможности. ***JConsole*** не входит в состав поставки сервера приложений, но включается во многие JVM. Подробнее работа с ***JConsole*** освещена в документации к Java: http://docs.oracle.com/javase/8/docs/technotes/guides/management/jconsole.html.

Для подключения к СП через ***JConsole***, на хосте пользователя должен быть установлен сервер не менее, чем WildFly 10.1.0 Final и пакет инструментов JDK не младше версии 1.8.0_77.

- Command Line Interface (**CLI**) - интерфейс командной строки, позволяющий управлять СП. Для успешного использования CLI требуется установленный WildFly 10.1.0 Final на хосте пользователя и JDK не младше версии 1.8.0_77.

На хостах СП, необходимо подготовить программную среду для работы утилит мониторинга ресурсов уровня ОС:

- NMON (сокращение от Nigel's Monitor) — инструмент администратора, предназначенный для анализа и мониторинга производительности Linux-систем. NMON можно скачать: http://nmon.sourceforge.net/pmwiki.php?n=Site.Download
- Стандартные утилиты мониторинга ресурсов уровня ОС (top, mpstat, vmstat, iostat).

Для Red Hat Enterprise Linux 6 или выше утилиты top, mpstat, vmstat и iostat включены в поставку. Для других ОС следует обеспечить их наличие.

Проверить наличие утилит можно вызовами команд через терминал, соответствующими именам утилит. Например:

```
# iostat
Linux 3.16.0-4-amd64 (jboss3) 03/22/2017 _x86_64_ (2 CPU)

avg-cpu: %user %nice %system %iowait %steal %idle
0.71 0.00 0.16 0.18 0.14 98.81

Device: tps kB_read/s kB_wrtn/s kB_read kB_wrtn
xvda 2.32 4.90 81.31 3068081 50884172
```

Утилиты top и vmstat входят в пакет **procps**, утилиты mpstat и iostat входят в пакет **sysstat**. Установив необходимый пакет можно получить советующий набор утилит.

Установка пакета procps и sysstat через утилиту **yum**:

```
yum -y install procps sysstat
```

или **apt-get**:

```
apt-get install procps sysstat
```

Zabbix Agent. Настройка программной среды освещена в *п.3.1.11.3 Настройка программной среды системы мониторинга Zabbix*

3.1.12.3 Настройка программной среды системы мониторинга Zabbix

На хостах, на которых будет выполнятся установка пакетов Zabbix Proxy, Zabbix Java Gateway или Zabbix Agent версии 3.0.x необходимо обеспечить:

- доступ в интернет;
- настроить репозиторий Zabbix.

Инструкция по настройке репозитория приведена в Zabbix Documentation 3.0: раздел 3. Установка из пакетов. п. 1. Установка репозитория: https://www.zabbix.com/documentation/3.0/ru/manual/installation/install_from_packages/repository_installation

- Для Zabbix Proxy и Zabbix Java Gateway на хосте обновить PHP до версии 5.4 или 5.5. Инструкция для обновления PHP на CentOS 5 и 6: http://www.shayanderson.com/linux/centos-5-or-centos-6-upgrade-php-to-php-54-or-php-55.htm

Установить зависимости:

```
wget http://dl.fedoraproject.org/pub/epel/7/x86_64/f/fping-3.10-4.el7.x86_64.rpm
rpm -Uhv fping-3.10-4.el7.x86_64.rpm
```

- Для процессов компонентов Zabbix требуется создать непривилегированного пользователя.

Если на хосте нет возможности обеспечить доступ в интернет, необходимо отдельно скачать пакеты компонентов Zabbix из репозитория и обеспечить наличие нужного пакета перед установкой советующего компонента системы мониторинга Zabbix.

Ссылка на репозиторий Zabbix и имена пакетов приведены в *п. 3.2.11.3.2 Установка компонентов Zabbix на хост без доступа к интернету*.

3.2 Установка, настройка и обновление серверного ПО решения WFM CC

3.2.1 БД WFM CC

3.2.1.1 Базовая установка СУБД PostgreSQL

Инструкции по установке и настройке СУБД PostgreSQL 10 описаны на официальном сайте - https://www.postgresql.org/docs/10/tutorial-install.html

```
yum -y install postgresql10 postgresql10-server postgresql10-contrib.x86_64
jsquery_10.x86_64
```

3.2.1.2 Создание системного пользователя БД

Перед первой установкой необходимо создать системного пользователя БД *argus_sys*

```
CREATEROLE argus_sys WITH LOGIN CREATEDB CREATEROLE PASSWORD '<пароль>';
```

3.2.1.3 Создание и настройка БД

Создать БД	CREATE DATABASE <имя базы>;
Сделать владельцем БД пользователя ***argus_sys***	ALTER DATABASE <имя базы> OWNER TO argus_sys;

Выдать необходимые привилегии пользователю ***argus_sys***	GRANT ALL PRIVILEGES ON DATABASE <имя базы> TO argus_sys
Определить ***search path***	ALTER DATABASE <имя базы> SET search_path = pg_catalog, public, system; ALTER ROLE **argus_sys** SET search_path = pg_catalog, public, system;
Создать схему ***dbm*** в БД, в которую будет происходить заливка данных	\c <имя базы> CREATE SCHEMA IF NOT EXISTS dbm AUTHORIZATION ***argus_sys***;

Пример создания пользователя БД (п.3.2.1.2); БД, привилегий и схем (п.3.2.1.3); расширений БД (п.3.2.1.4)

```
CREATE ROLE argus_sys WITH LOGIN CREATEDB CREATEROLE PASSWORD '***';
CREATE DATABASE prod OWNER argus_sys;

ALTER DATABASE prod SET search_path = pg_catalog, public, system;
ALTER ROLE argus_sys SET search_path = pg_catalog, public, system;

\c prod
CREATE SCHEMA IF NOT EXISTS dbm AUTHORIZATION argus_sys;

CREATE EXTENSION IF NOT EXISTS jsquery SCHEMA public;
CREATE EXTENSION IF NOT EXISTS btree_gin SCHEMA public;
CREATE EXTENSION IF NOT EXISTS pg_trgm SCHEMA public;
CREATE EXTENSION IF NOT EXISTS btree_gist SCHEMA public;
CREATE EXTENSION IF NOT EXISTS lo SCHEMA public;
```

3.2.1.4 Расширения необходимые для работы БД

В БД, созданной в п. 3.2.1.3 необходимо установить расширения PostgreSQL:

```
\c <имя базы>

CREATE EXTENSION IF NOT EXISTS jsquery SCHEMA public;
CREATE EXTENSION IF NOT EXISTS btree_gin SCHEMA public;
CREATE EXTENSION IF NOT EXISTS pg_trgm SCHEMA public;
CREATE EXTENSION IF NOT EXISTS btree_gist SCHEMA public;
CREATE EXTENSION IF NOT EXISTS lo SCHEMA public;
```

Если соответствующего расширения нет в системе, то его необходимо предварительно установить, например для СУБД PostgreSQL10 :

```
yum install jsquery_10
```

3.2.1.5 Обновление БД

Получить файл *update-database-<version>.zip*

Распаковать полученный архив *update-database-<version>.zip*

```
cd /argus && unzip update-database-<version>.zip
```

Задать параметры доступа к базе в файле **dbmaintain.properties**

Пример dbmaintain.properties

```
database.dialect=postgresql
database.driverClassName=org.postgresql.Driver
database.password=***
database.url=jdbc\:postgresql\://192.168.100.10\:5432/prod
database.userName=argus_sys
databases.names=prod
dbMaintainer.allowOutOfSequenceExecutionOfPatches=true
dbMaintainer.script.ignoreCarriageReturnsWhenCalculatingCheckSum=true
dbMaintainer.script.locations=data/update-all-ver_0.1.0.jar
```

Выполнить проверку

```
./dbmaintain.sh checkScriptUpdates
```

Если проверка показала наличие скриптов для выполнения, то выполнить обновление

```
./dbmaintain.sh updateDatabase
```

Если вызов будет завершен строкой **The database has been updated successfully** значит обновление успешно установлено.
В ином случае необходимо сообщить о проблеме специалистам поддержки НТЦ АРГУС, отправить лог файл и ждать особых рекомендации по дальнейшему сценарию действий и (или) восстановлению работоспособности системы.

3.2.2 СП WFM CC

3.2.2.1 Проверка настройки программной среды

Необходимо убедиться, что требования из п. 3.1.2 выполнены.

3.2.2.2 Распаковка архива пакета установки

Распаковать архив пакета установки на хосте СП, используя утилиту ***unzip***:

```
unzip [имя_архива_пакета_установки].zip
```

Архив должен быть распакован именно на хосте СП, а не на локальной машине администратора, В противном случае, передача распакованного архива по сети (с помощью протокола SCP/FTP или им подобного) – приведет к проблеме с русскими буквами в именах файлов.
Архив будет распакован в текущий каталог. При извлечении файлов в консоль вместо русских могут выводиться знаки вопроса. Это нормальная ситуация.

3.2.2.3 Сохранение резервной копии ранее установленного СП и его восстановление

Если установка сервера производится с целью обновления установленного ранее сервера, следует создать резервную копию установленного ранее сервера.

Для создания резервной копии сервера приложений достаточно создать копию каталога установки сервера. Останавливать сервер приложений при этом не требуется. Восстановить сервер можно копированием в каталог установки сохраненной копии каталога, предварительно остановив сервер.

3.2.2.4 Установка СП

Получить файл argus-dist-<версия>-ccwfm.jar

Для установки СП необходимо выполнить команду:

```
java -jar argus-dist-<версия>-ccwfm.jar -options prod.properties
```

3.2.2.5 Запуск СП

Перед запуском необходимо убедиться, что СП еще не запущен.

Проверить текущий статус установленного СП можно с помощью параметра ***status*** скрипта **runjboss.sh**, для чего перейти в каталог установки СП ***INSTALL_PATH/bin*** выполнить команду:

```
./runjboss.sh status
```

Результатом выполнения команды будет:

wildfly started (pid 1535) - сервер запущен.

или

wildfly not started - сервер не запущен.

Для того, чтобы запустить СП необходимо перейти в каталог **<INSTALL_PATH>*/bin*** и выполнить команду:

```
cd <INSTALL_PATH>/bin;./runjboss.sh start
```

Результатом выполнения команды будет:

```
Starting wildfly in default mode (standalone)...
```

Примечание:

Запустить СП под непривелигированной учетной записью (например, argus), если текущий пользователь root, можно выполнив команду:

```
sudo -u argus ./runjboss.sh start
```

Запуск СП занимает несколько минут. После завершения фазы запуска СП пользователям становится доступен веб-интерфейс, до завершения процесса загрузки большого объема данных из БД в кэш СП, если кэши еще ни разу не загружались.

Сервер запустится когда в файле *<INSTALL_PATH>/standalone/log/last_boot_errors.log* будет написано «**Сервер запущен**»

Если в файле **last_boot_errors.log** есть ошибки – то необходимо сообщить о проблеме специалистам поддержки НТЦ АРГУС, отправить все лог файлы (всю папку logs) и ждать особых рекомендации по дальнейшему сценарию действий и (или) восстановлению работоспособности системы.

Примечание:

Так как процесс обновления кэшей СП завершается вне фазы запуска СП, то в ***last_boot_errors.log*** могут быть не видны ошибки синхронизации кэшей СП.

3.2.2.6 Остановка СП

Для остановки сервера следует перейти в подкаталог **bin** каталога установки и выполнить команду ***runjboss.sh*** с одним из нижеуказанных параметров:

stop - нормальная остановка сервера[64].

stop kill - не дожидаясь остановки сервера, завершает его работу.

Ход остановки протоколируется в лог-файл:

КаталогУстановки/standalone/log/server.log.

По завершении остановки в него выводится сообщение:

```
16:48:14,397 INFO [as] (MSC service thread 1-2) JBAS015950: WildFly 10.1.0.Final
"Tweek" stopped in 1155ms
```

При успешном выполнении операции остановки СП, на экран будут выведены сообщения:

```
Stopping wildfly:Done.
```

Если СП не удается остановить штатными средствами, в ОС Linux для остановки процесса СП можно использовать команду kill:

```
kill -9 pid
```

где pid - идентификатор процесса сервера приложений в ОС.

3.2.2.7 Файл конфигурации

Создать конфигурационный файл для запуска **prod.properties** и указать в нем все необходимые параметры

Пример конфигурационного файла

```
# Базовые настройки

INSTALL_PATH=/argus/jboss_prod/
argus.app.memory.max-size=8192
```

[64] Остановка сервера командой ***runjboss.sh stop*** может занимать несколько минут.

```
argus.app.debug-mode.enabled=false
jboss.bind.address.management=0.0.0.0
argus.app.admin.user=developer
argus.app.admin.pass=***
argus.java.home.path=/usr/java/jdk1.8.0_202-amd64
argus.app.security-mode.enabled=false
argus.db.address=192.168.100.10
argus.db.name=demodb
argus.db.port=5432
argus.db.user=argus_sys
argus.db.pass=***
jboss.bind.address=192.168.100.20
jboss.socket.binding.port-offset=0

# Доступ к каждому из сервисов с которым взаимодействует СП WFM CC согласно ТА

ccwfm.notification.service.enabled=true
ccwfm.notification.service.url=http://192.168.100.25:9000


ccwfm.reportservice.url=http://192.168.100.24:9000
ccwfm.reportservice.url.callback=http://192.168.100.20:8080/ccwfm/reportresult


ccwfm.planningservice.url=http://192.168.100.23:9000
ccwfm.planningservice.url.callback=http://192.168.100.20:8080/ccwfm/planningresult
```

3.2.2.8 Настройка УЗ администратора СП

УЗ администратора предоставляет доступ к средствам администрирования СП. Она создается двумя способами:

- По значениям настроек во время установки СП:

```
argus.app.admin.userargus.app.admin.pass
```

- С помощью скрипта ***add-user.sh***, находящийся в подкаталоге ***bin*** каталога установленного СП.

Пример:

Перейдя в подкаталог **bin** установленного СП, выполнить:

```
./add_user.sh
```

Будет выведено сообщение о выбираемом типе создаваемого пользователя (см. рис. 3.2.2.12_1). Следует выбрать тип ***Management User***, предоставляющий привилегии управления сервером приложений, указав значение: ***a***

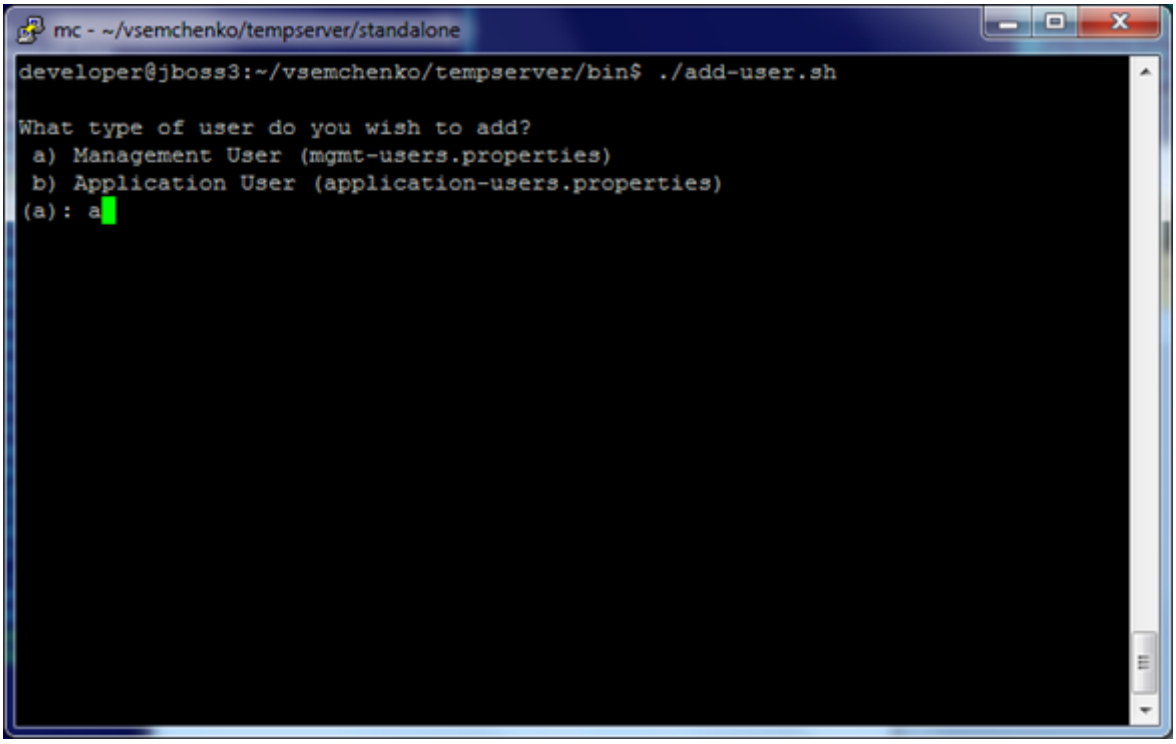


Рисунок 3.2.2.8_1 - Тип пользователя

Далее ввести имя учетной записи (см. рис. 3.2.2.12_2), например: ***app-admin***

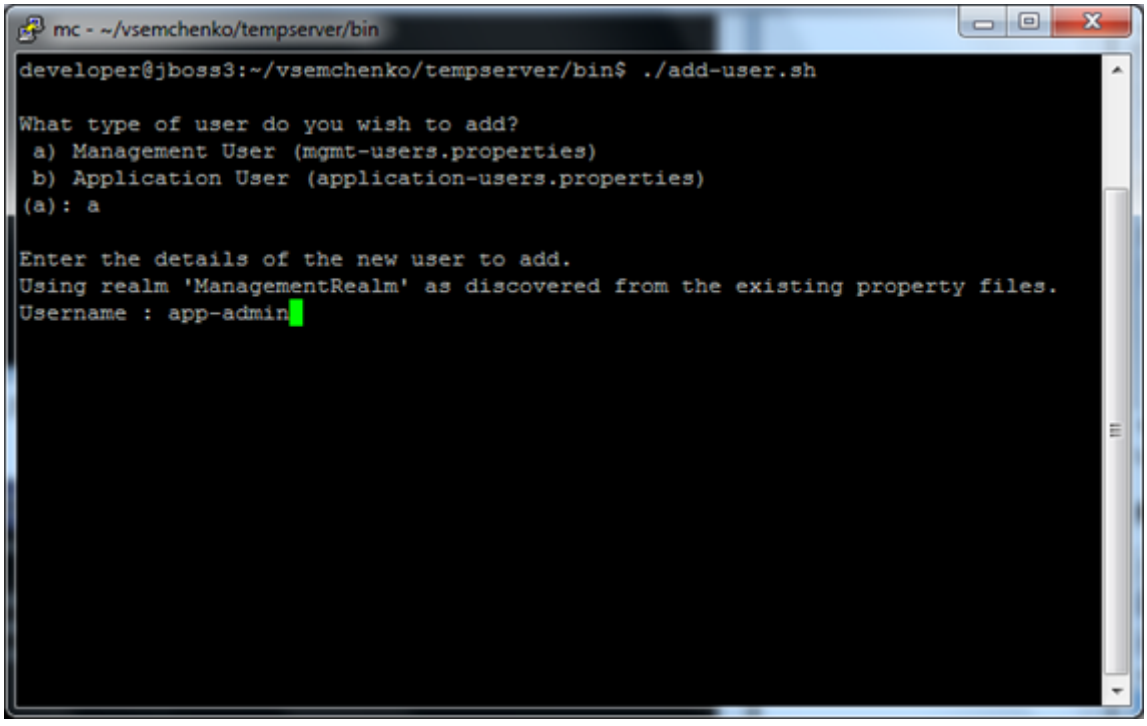


Рисунок 3.2.2.8_2 - Имя УЗ

Далее на экран выведутся рекомендации к паролю (см. рис. 3.2.2.8_3):

- *Пароль не должен содержать следующих значений: root, admin, administrator.*
- *Пароль должен содержать по крайней мере 8 символов, один буквенный символ, одну цифру и один не алфавитно-цифровой символ.*
- *Пароль должен отличаться от имени пользователя.*

Следует вводить пароль, придерживаясь данных рекомендаций, например: ***my-password***

На вопрос*: Are you sure you want to use the password entered yes/no?*

Следует выбрать положительный ответ: ***y***

После чего повторно ввести пароль: ***my-password***

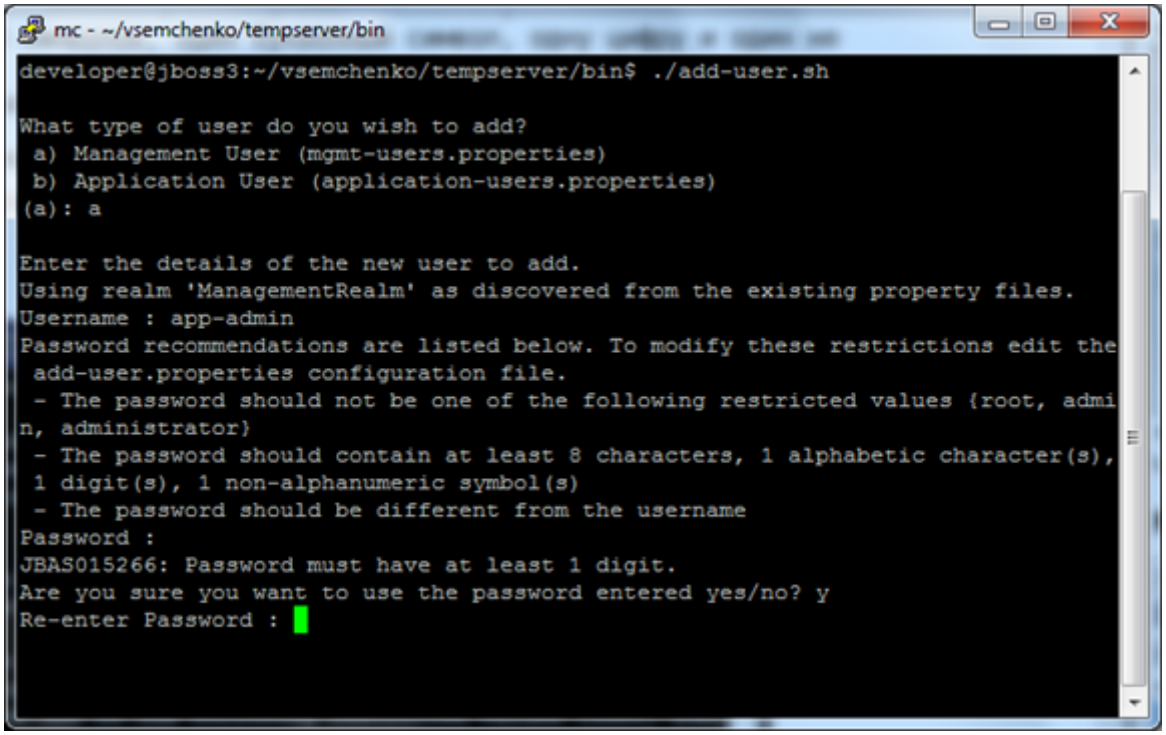


Рисунок 3.2.2.8_3 - Пароль УЗ

Для окончания настройки создаваемой учетной записи (см. рис. 3.2.2.12_4) необходимо ответить на ряд вопросов:

- What groups do you want this user to belong to? (Please enter a comma separated list, or leave blank for none)[ ]:

Следует нажать ***Enter*** .

- About to add user 'app-admin' for realm 'ManagementRealm'

Is this correct yes/no?

Требуется подтвердить выполняемую операцию: ***y***

- Is this new user going to be used for one AS process to connect to another AS process?

e.g. for a slave host controller connecting to the master or for a Remoting connection for server to server EJB calls.

yes/no?

Следует ввести: ***n***

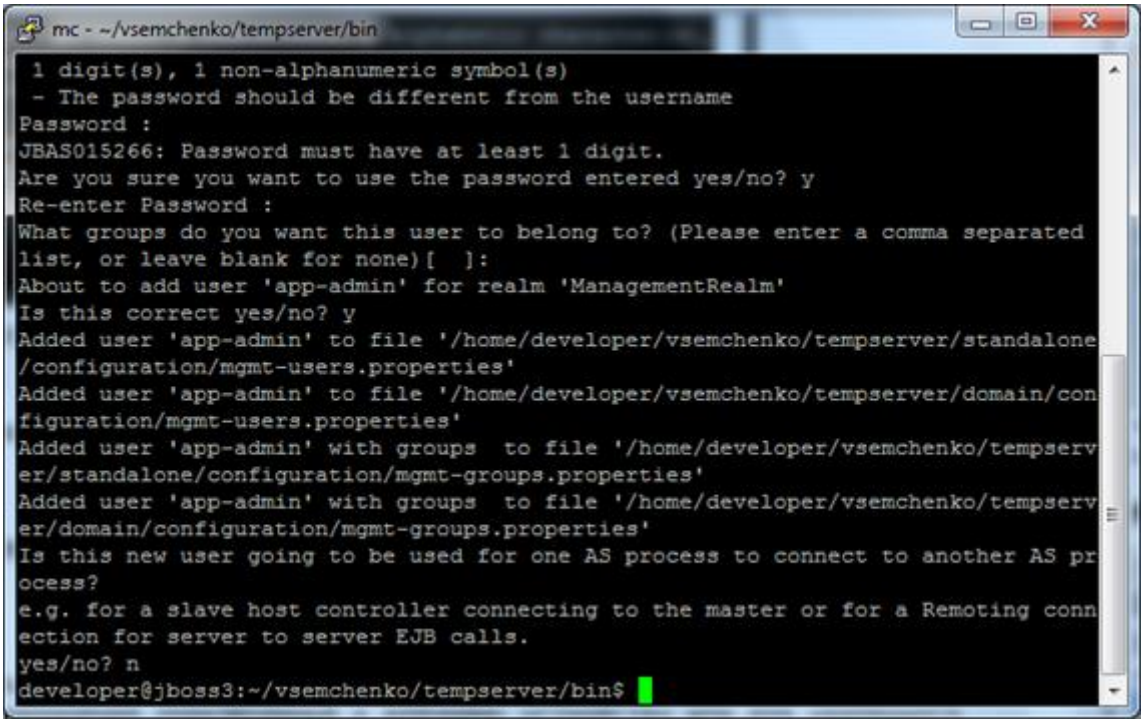


Рисунок 3.2.2.8_4 - Окончательная настройка УЗ

Для проверки успешности создания УЗ администратора СП,

Для подтверждения успешного создания учетной записи, необходимо зайти через web-браузер на веб-интерфейс управления СП ***http://IP-адрес:Порт/***, где:

IP-адрес – IP-адрес СП

Порт – порт, на котором сервер ожидает management-http подключений.

Значение порта определяется по формуле: ***9990 + port-offset***,

где ***port-offset*** - смещение портов относительно значений по умолчанию.

Отобразится приглашение на ввод имени и пароля созданной УЗ администратора СП (см. рис.3.2.2.8_5).

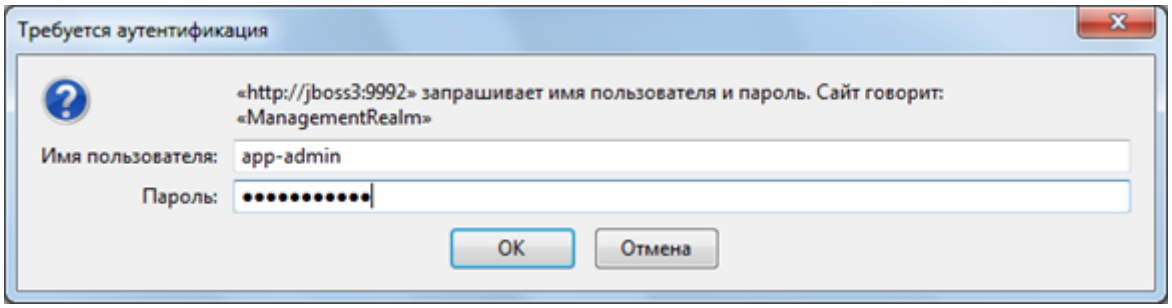


Рисунок 3.2.2.8_5 - Ввод логина и пароля УЗ администратора СП

После ввода логина и пароля УЗ администратора СП, будет отображена главная веб-страница управления СП (см. рис.3.2.2.8_6).

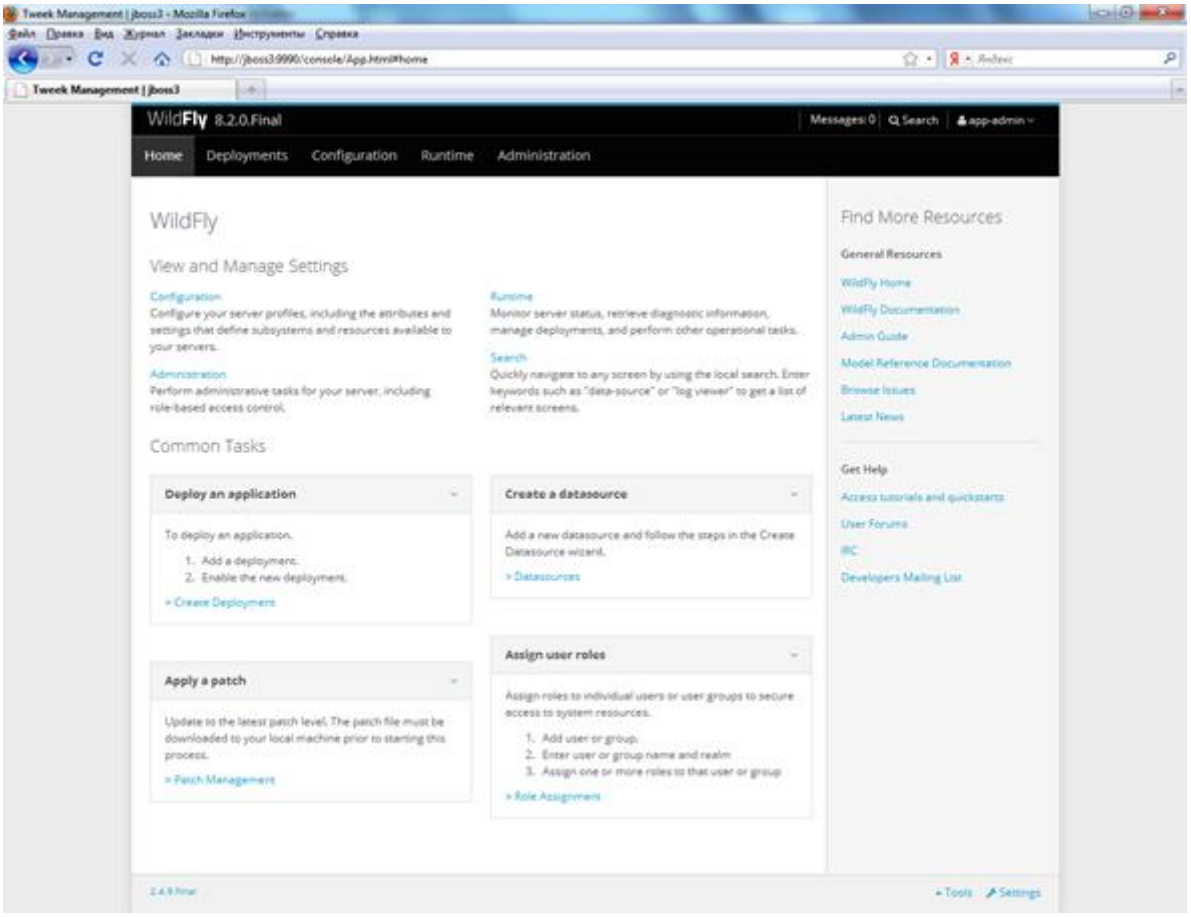


Рисунок 3.2.2.8_6 - Главная веб-страница управления СП

3.2.3 Сервис Личный кабинет WFM CC

3.2.3.1 Проверка настройки программной среды

Необходимо убедиться, что требования из п. 3.1.3 выполнены.

3.2.3.2 Установка обновлений сервиса Личный кабинет WFM CC

При установке обновлений сервиса Личный кабинет WFM CC необходимо придерживаться следующего порядка действий

1. остановить контейнер.
2. скачать образ
3. обновить образ.
4. внести изменения в конфигурационные файлы[65].
5. запустить контейнер.
6. если пп.1-5 выполнены успешно, то удалить старый контейнер и образ старого контейнера.

Примеры команд[66]

информация по текущим контейнерам на хосте

```
docker ps -a
```

[65] .env; docker-compose.yml

[66] Все необходимые подробности по командам можно посмотреть в оригинальной документации по docker: https://docs.docker.com/engine/reference/commandline/docker/

остановить одиночный контейнер

```
docker stop <container_id>
```

или все контейнеры, описанные в docker-compose.yml

```
docker-compose stop
```

скачать образ из удаленного репозитория Аргус в локальный репозиторий

```
docker pull gitlab:4567/laboratorium/personal-area/front:dev-<version>

docker pull gitlab:4567/laboratorium/mobile-api:dev-<version>
```

В случае, если доступа к удалённому репозиторию Аргус нет, то на стороне Аргус необходимо скачать образ из удаленного репозитория Аргус в локальный репозиторий

```
docker pull gitlab:4567/laboratorium/personal-area/front:dev-<version>

docker pull gitlab:4567/laboratorium/mobile-api:dev-<version>
```

Экспортировать образ

```
docker image save -o personal-area-<version>.tar ggitlab:4567/laboratorium/personal-area/front:dev-<version>

docker image save -o mobile-api-lk<version>.tar gitlab:4567/laboratorium/mobile-api:dev-<version>
```

После чего передать архив с образом контейнера (personal-area-<version>.tar и mobile-api-lk<version>.tar) заказчику

На стороне заказчика необходимо загрузить образ в локальный репозиторий docker

```
docker load -i personal-area-<version>.tar

docker load -i mobile-api-lk<version>.tar
```

Внести изменения в конфигурационные файлы

.env

```
HOST_IP=192.168.47.3
RMI_PORT=9067
JAVA_OPTS=-Dcom.sun.management.jmxremote -Dcom.sun.management.jmxremote.port=9067 -
Dcom.sun.management.jmxremote.rmi.port=9067 -
Dcom.sun.management.jmxremote.local.only=false -
Dcom.sun.management.jmxremote.authenticate=false -
Dcom.sun.management.jmxremote.ssl=false -Djava.rmi.server.hostname=192.168.47.3 -
XX:+HeapDumpOnOutOfMemoryError -XX:HeapDumpPath=/argus/logs/ -
XX:+PrintCommandLineFlags -XX:MinRAMPercentage=10.0 -XX:MaxRAMPercentage=90.0
DB_ADDR=192.168.47.5:5432
```

```
DB_NAME=demodb
TZ=Europe/Moscow
MAIN_API_URL= http://192.168.47.3:9060
CCWFM_URL=http://192.168.47.2:8080
```

docker-compose.yml

```
version: "3"
services:
personal-area:
container_name: personal-area
image: gitlab:4567/laboratorium/personal-area/front:dev
ports:
- "9050:8080"
restart: always
environment:
- MAIN_API_URL
- TZ

volumes:
- personal-area-logs:/argus/logs
logging:
driver: "json-file"
options:
max-size: "200m"
max-file: "10"

mobile-api-lk:
container_name: mobile-api-lk
image: gitlab:4567/laboratorium/mobile-api:dev
ports:
- "9060:8080"
- "9067:9067"
restart: always
environment:
- DB_ADDR
- DB_NAME
- HOST_IP
- RMI_PORT
- JAVA_OPTS
- GW_MODE=webapp
- CCWFM_URL
- TZ

volumes:
- mobile-api-lk-logs:/argus/logs
logging:
driver: "json-file"
options:
max-size: "200m"
max-file: "10"

volumes:
personal-area-logs:
driver: local
driver_opts:
o: bind
```

```
type: none
device: /argus/personal-area/logs
mobile-api-lk-logs:
driver: local
driver_opts:
o: bind
type: none
device: /argus/mobile-api-lk/logs
```

Запустить одиночный контейнер[67]

```
docker start <container_id>
```

или запустить все контейнеры, описанные в ***docker-compose.yml***

```
docker-compose up -d
```

Проверить успешность запуска контейнера

```
docker ps -a

docker logs -f personal-area

docker logs -f mobile-api-lk
```

Удалить контейнер

```
docker rm <container_id>
```

Информация по текущим образам в репозитории

```
docker images
```

Удалить образ контейнера.

```
docker rmi <image_id>
```

3.2.3.3 Запуск сервиса

Запустить все контейнеры, описанные в ***docker-compose.yml***

```
docker-compose up -d
```

Проверить успешность запуска контейнера

```
docker ps -a
```

[67] Каждый контейнер для ***personal-area*** и ***mobile-api-lk***

```
docker logs -f lk-service
```

3.2.3.4 Остановка сервиса

Информация по текущим контейнерам на хосте

```
docker ps -a
```

Остановить одиночный контейнер

```
docker stop <container_id>
```

Или все контейнеры, описанные в ***docker-compose.yml***

```
docker-compose stop
```

3.2.3.5 Файлы конфигурации

.env

Пример файла конфигурации

```
HOST_IP=192.168.47.3
DB_ADDR=192.168.47.5:5432
DB_NAME=demodb
TZ=Europe/Moscow
MAIN_API_URL= http://192.168.47.3:9060
CCWFM_URL=http://192.168.47.2:8080
```

docker-compose.yml

Пример файла конфигурации

```
version: "3"
services:
  personal-area:
    container_name: personal-area
    image: gitlab:4567/laboratorium/personal-area/front:dev
    ports:
      - "9050:8080"
    restart: always
    environment:
      - MAIN_API_URL
      - TZ
    volumes:
      - personal-area-logs:/argus/logs
    logging:
      driver: "json-file"
      options:
        max-size: "200m"
        max-file: "10"
  mobile-api-lk:
    container_name: mobile-api-lk
    image: gitlab:4567/laboratorium/mobile-api:dev
    ports:
```

```
      - "9060:8080"
      - "9067:9067"
    restart: always
    environment:
      - DB_ADDR
      - DB_NAME
      - HOST_IP
      - RMI_PORT
      - JAVA_OPTS
      - GW_MODE=webapp
      - CCWFM_URL
      - TZ
    volumes:
      - mobile-api-lk-logs:/argus/logs
    logging:
      driver: "json-file"
      options:
        max-size: "200m"
        max-file: "10"
volumes:
  personal-area-logs:
    driver: local
    driver_opts:
      o: bind
      type: none
      device: /argus/personal-area/logs
  mobile-api-lk-logs:
    driver: local
    driver_opts:
      o: bind
      type: none
      device: /argus/mobile-api-lk/logs
```

3.2.4 Сервис Мобильный API WFM CC

3.2.4.1 Проверка настройки программной среды

Необходимо убедиться, что требования из п. 3.1.4 выполнены.

3.2.4.2 Установка обновлений и настройка сервиса Мобильный API WFM CC

Настройка сервиса осуществляется путем внесения изменений в конфигурационные файлы[68] с последующим перезапуском сервиса.

При установке обновлений сервиса Мобильный API WFM CC необходимо придерживаться следующего порядка действий

1. остановить контейнер
2. скачать образ
3. обновить образ
4. внести изменения в конфигурационные файлы
5. запустить контейнер
6. если пп.1-5 выполнены успешно, то удалить старый контейнер и образ старого контейнера

[68] .env; docker-compose.yml

примеры команд[69]

Информация по текущим контейнерам на хосте

```
docker ps -a
```

Остановить одиночный контейнер

```
docker stop <container_id>
```

Или все контейнеры, описанные в docker-compose.yml

```
docker-compose stop
```

Скачать образ из удаленного репозитория Аргус в локальный репозиторий

```
docker pull gitlab:4567/laboratorium/mobile-api:release-<version>
```

В случае, если доступа к удалённому репозиторию Аргус нет, то на стороне Аргус необходимо скачать образ из удаленного репозитория Аргус в локальный репозиторий

```
docker pull gitlab:4567/laboratorium/mobile-api:release-<version>
```

Экспортировать образ

```
docker image save -o lk-<version>.tar gitlab:4567/laboratorium/mobile-api:release-
<version>
```

После чего передать архив с образом контейнера (mobile-api-<version>.tar) заказчику

на стороне заказчика необходимо загрузить образ в локальный репозиторий docker

```
docker load -i mobile-api-<version>.tar
```

Внести изменения в конфигурационные файлы

.env

```
HOST_IP=192.168.47.4
RMI_PORT=9017
JAVA_OPTS=-Dcom.sun.management.jmxremote -Dcom.sun.management.jmxremote.port=9017 -
Dcom.sun.management.jmxremote.rmi.port=9017 -
Dcom.sun.management.jmxremote.local.only=false -
Dcom.sun.management.jmxremote.authenticate=false -
Dcom.sun.management.jmxremote.ssl=false -Djava.rmi.server.hostname=192.168.47.4 -
XX:+HeapDumpOnOutOfMemoryError -XX:HeapDumpPath=/argus/logs/ -
XX:+PrintCommandLineFlags -XX:MinRAMPercentage=10.0 -XX:MaxRAMPercentage=90.0
DB_ADDR=192.168.47.5:5432
DB_NAME=demodb
TZ=Europe/Moscow
CCWFM_URL=http://192.168.47.2:8080
```

docker-compose.yml

```
version: "3"
services:
mobile-api:
container_name: mobile-api
```

[69] Все необходимые подробности по командам можно посмотреть в оригинальной документации по docker: https://docs.docker.com/engine/reference/commandline/docker/

```
image: gitlab:4567/laboratorium/mobile-api:release-1.3.0
ports:
- "9010:8080"
- "9017:9017"
restart: always
environment:
- DB_ADDR
- DB_NAME
- HOST_IP
- RMI_PORT
- JAVA_OPTS
- GW_MODE=mobile
- CCWFM_URL
- TZ

volumes:
- mobile-api-logs:/argus/logs
logging:
driver: "json-file"
options:
max-size: "200m"
max-file: "10"

volumes:
mobile-api-logs:
driver: local
driver_opts:
o: bind
type: none
device: /argus/mobile-api/logs
```

Запустить все контейнеры, описанные в ***docker-compose.yml***

```
docker-compose up -d
```

Проверить успешность запуска контейнера

```
docker ps -a

docker logs -f mobile-api
```

Удалить контейнер

```
docker rm <container_id>
```

Информация по текущим образам в репозитории

```
docker images
```

Удалить образ контейнера.

```
docker rmi <image_id>
```

3.2.4.3 Запуск сервиса

Запустить все контейнеры, описанные в ***docker-compose.yml***

```
docker-compose up -d
```

проверить успешность запуска контейнера

```
docker ps -a

docker logs -f mobile-api
```

3.2.4.4 Остановка сервиса

Информация по текущим контейнерам на хосте

```
docker ps -a
```

Остановить одиночный контейнер

```
docker stop <container_id>
```

Или все контейнеры, описанные в ***docker-compose.yml***

```
docker-compose stop
```

3.2.4.5 Файлы конфигурации

.env

Пример файла конфигурации

```
HOST_IP=192.168.47.4
DB_ADDR=192.168.47.5:5432
DB_NAME=demodb
TZ=Europe/Moscow
CCWFM_URL=http://192.168.47.2:8080
```

docker-compose.yml

Пример файла конфигурации

```
version: "3"
services:
  mobile-api:
    container_name: mobile-api
    image: gitlab:4567/laboratorium/mobile-api:release-1.3.0
    ports:
      - "9010:8080"
      - "9017:9017"
    restart: always
    environment:
      - DB_ADDR
      - DB_NAME
      - HOST_IP
      - RMI_PORT=9017
      - JAVA_OPTS=-Dcom.sun.management.jmxremote -
Dcom.sun.management.jmxremote.port=9017 -
Dcom.sun.management.jmxremote.rmi.port=9017 -
Dcom.sun.management.jmxremote.local.only=false -
Dcom.sun.management.jmxremote.authenticate=false -
```

```
Dcom.sun.management.jmxremote.ssl=false -Djava.rmi.server.hostname=192.168.47.4
-XX:+HeapDumpOnOutOfMemoryError -XX:HeapDumpPath=/argus/logs/ -
XX:+PrintCommandLineFlags -XX:MinRAMPercentage=10.0 -XX:MaxRAMPercentage=90.0
      - GW_MODE=mobile
      - CCWFM_URL
      - TZ
    volumes:
      - mobile-api-logs:/argus/logs
    logging:
      driver: "json-file"
      options:
        max-size: "200m"
        max-file: "10"
volumes:
  mobile-api-logs:
    driver: local
    driver_opts:
      o: bind
      type: none
      device: /argus/mobile-api/logs
```

3.2.4.6 Доступ к сервису Мобильный API по HTTPS

Сервис Мобильный API работает по протоколу HTTP, поэтому для работы с сервисом Мобильный API по протоколу HTTPS

необходим реверс-прокси [70]для конвертации трафика из HTTPS в HTTPS.

На стороне реверс-прокси необходимо наличие сертификата и ключа - файлов, находящихся в каталоге, указанном в конфигурационном файле /etc/nginx/nginx.conf

например:

ssl_certificate /etc/nginx/ssl/**nginx.crt**;

ssl_certificate_key /etc/nginx/ssl/**nginx.key**;

Замена старого сертификата и ключа на новые осуществляется подменой соответствующих файлов в каталоге, указанном в конфигурационном файле /etc/nginx/nginx.conf

После замены файлов необходимо перезапустить сервис nginx

3.2.5 Сервис планирования

Для работы сервиса планирования необходима работа двух сервисов, каждый из которых запускается в своем контейнере docker:

- шлюз (gateway) - ***planning-gw***
- собственно, сервис планирования - ***planning-service***

[70] веб-сервер nginx

3.2.5.1 Проверка настройки программной среды

Необходимо убедиться, что требования из п. 3.1.5 выполнены.

3.2.5.2 Установка обновлений сервиса планирования

При установке обновлений сервиса планирования необходимо придерживаться следующего порядка действий

1. остановить контейнер.
2. скачать образ
3. обновить образ.
4. внести изменения в конфигурационные файлы[71].
5. запустить контейнер.
6. если пп.1-5 выполнены успешно, то удалить старый контейнер и образ старого контейнера.

Примеры команд[72]

Информация по текущим контейнерам на хосте

```
docker ps -a
```

Остановить одиночный контейнер[73]

```
docker stop <container_id>
```

Или все контейнеры, описанные в docker-compose.yml

```
docker-compose stop
```

Скачать образ из удаленного репозитория Аргус в локальный репозиторий

```
docker pull gitlab:4567/laboratorium/planning-gw:release-<version>

docker pull gitlab:4567/laboratorium/planning-service:release-<version>
```

В случае, если доступа к удалённому репозиторию Аргус нет, то на стороне Аргус необходимо скачать образ из удаленного репозитория Аргус в локальный репозиторий

```
docker pull gitlab:4567/laboratorium/planning-gw:release-<version>

docker pull gitlab:4567/laboratorium/planning-service:release-<version>
```

Экспортировать образ

```
docker image save -o planning-gw-<version>.tar gitlab:4567/laboratorium/planning-gw:release-<version>

docker image save -o planning-service-<version>.tar gitlab:4567/laboratorium/planning-service:release-<version>
```

После чего передать архив с образом контейнера (planning-gw-<version>.tar и planning-service-<version>.tar) заказчику

---

[71] .env; docker-compose.yml

[72] Все необходимые подробности по командам можно посмотреть в оригинальной документации по docker: https://docs.docker.com/engine/reference/commandline/docker/

[73] Каждый контейнер для ***planning-gw*** и ***planning-service***

На стороне заказчика необходимо загрузить образ в локальный репозиторий docker

```
docker load -i planning-gw-<version>.tar

docker load -i planning-service-<version>.tar
```

Вести изменения в конфигурационные файлы

.env

```
HOST_IP=192.168.47.8
DB_ADDR=192.168.47.5:5432
DB_NAME=demodbOPERATING_SCHEDULE_SECONDS_SPENT_LIMIT=144000TIMETABLE_SECONDS_SPENT_L
IMIT=144000

TZ=Europe/Moscow
```

docker-compose.yml

```
version: "3"
services:
planning-gw:
container_name: planning-gw
image: gitlab:4567/laboratorium/planning-service/gateway:release-1.4.0
ports:
- "9030:8080"
- "9037:9037"
restart: always
environment:
- DB_ADDR
- DB_NAME
- HOST_IP
- RMI_PORT=9037
- JAVA_OPTS=-Dcom.sun.management.jmxremote -Dcom.sun.management.jmxremote.port=9037
-Dcom.sun.management.jmxremote.rmi.port=9037 -
Dcom.sun.management.jmxremote.local.only=false -
Dcom.sun.management.jmxremote.authenticate=false -
Dcom.sun.management.jmxremote.ssl=false -Djava.rmi.server.hostname=192.168.47.8 -
XX:+HeapDumpOnOutOfMemoryError -XX:HeapDumpPath=/argus/logs/ -
XX:+PrintCommandLineFlags -XX:MinRAMPercentage=10.0 -XX:MaxRAMPercentage=90.0
- TZ
volumes:
- planning-gw-logs:/argus/logs
logging:
driver: "json-file"
options:
max-size: "200m"
max-file: "10"

planning-service:
container_name: planning-service
image: gitlab:4567/laboratorium/planning-service/service:release-1.4.0
ports:
- "9047:9047"
restart: always
environment:
- DB_ADDR
- DB_NAME
- HOST_IP
- RMI_PORT=9047
- JAVA_OPTS=-Dcom.sun.management.jmxremote -Dcom.sun.management.jmxremote.port=9047
-Dcom.sun.management.jmxremote.rmi.port=9047 -
Dcom.sun.management.jmxremote.local.only=false -
```

```
Dcom.sun.management.jmxremote.authenticate=false -
Dcom.sun.management.jmxremote.ssl=false -Djava.rmi.server.hostname=192.168.47.8 -
XX:+HeapDumpOnOutOfMemoryError -XX:HeapDumpPath=/argus/logs/ -
XX:+PrintCommandLineFlags -XX:MinRAMPercentage=10.0 -XX:MaxRAMPercentage=90.0-
OPERATING_SCHEDULE_SECONDS_SPENT_LIMIT- TIMETABLE_SECONDS_SPENT_LIMIT- TZ

volumes:
- planning-service-logs:/argus/logs
logging:
driver: "json-file"
options:
max-size: "200m"
max-file: "10"

volumes:
planning-gw-logs:
driver: local
driver_opts:
o: bind
type: none
device: /argus/planning-gw/logs
planning-service-logs:
driver: local
driver_opts:
o: bind
type: none
device: /argus/planning-service/logs
```

Запустить одиночный контейнер[74]

```
docker start <container_id>
```

Или все контейнеры, описанные в docker-compose.yml

```
docker-compose up -d
```

Проверить успешность запуска контейнера

```
docker ps -a

docker logs -f planning-gw

docker logs -f planning-service
```

Удалить контейнер

```
docker rm <container_id>
```

Информация по текущим образам в репозитории

```
docker images
```

Удалить образ контейнера

```
docker rmi <image_id>
```

[74] Каждый контейнер для ***planning-gw*** и ***planning-service***

3.2.5.3 Запуск сервиса

Запустить одиночный контейнер[75]

```
docker start <container_id>
```

Или все контейнеры, описанные в ***docker-compose.yml***

```
docker-compose up -d
```

Проверить успешность запуска контейнера

```
docker ps -a

docker logs -f planning-gw

docker logs -f planning-service
```

3.2.5.4 Остановка сервиса

Информация по текущим контейнерам на хосте

```
docker ps -a
```

Остановить одиночный контейнер[76]

```
docker stop <container_id>
```

Или все контейнеры, описанные в ***docker-compose.yml***

```
docker-compose stop
```

3.2.5.5 Файлы конфигурации

.env

Пример файла конфигурации

```
HOST_IP=192.168.47.8
DB_ADDR=192.168.47.5:5432
DB_NAME=demodb
OPERATING_SCHEDULE_SECONDS_SPENT_LIMIT=144000
TIMETABLE_SECONDS_SPENT_LIMIT=144000
TZ=Europe/Moscow
```

docker-compose.yaml

Пример файла конфигурации

```
version: "3"
services:
```

[75] Каждый контейнер для ***planning-gw*** и ***planning-service***
[76] каждый контейнер для ***planning-gw*** и ***planning-service***

```
  planning-gw:
    container_name: planning-gw
    image: gitlab:4567/laboratorium/planning-service/gateway:release-1.4.0
    ports:
      - "9030:8080"
      - "9037:9037"
    restart: always
    environment:
      - DB_ADDR
      - DB_NAME
      - HOST_IP
      - RMI_PORT=9037
      - JAVA_OPTS=-Dcom.sun.management.jmxremote -
Dcom.sun.management.jmxremote.port=9037 -
Dcom.sun.management.jmxremote.rmi.port=9037 -
Dcom.sun.management.jmxremote.local.only=false -
Dcom.sun.management.jmxremote.authenticate=false -
Dcom.sun.management.jmxremote.ssl=false -Djava.rmi.server.hostname=192.168.47.8
-XX:+HeapDumpOnOutOfMemoryError -XX:HeapDumpPath=/argus/logs/ -
XX:+PrintCommandLineFlags -XX:MinRAMPercentage=10.0 -XX:MaxRAMPercentage=90.0
      - TZ
    volumes:
      - planning-gw-logs:/argus/logs
    logging:
      driver: "json-file"
      options:
        max-size: "200m"
        max-file: "10"
  planning-service:
    container_name: planning-service
    image: gitlab:4567/laboratorium/planning-service/service:release-1.4.0
    ports:
      - "9047:9047"
    restart: always
    environment:
      - DB_ADDR
      - DB_NAME
      - HOST_IP
      - RMI_PORT=9047
      - JAVA_OPTS=-Dcom.sun.management.jmxremote -
Dcom.sun.management.jmxremote.port=9047 -
Dcom.sun.management.jmxremote.rmi.port=9047 -
Dcom.sun.management.jmxremote.local.only=false -
Dcom.sun.management.jmxremote.authenticate=false -
Dcom.sun.management.jmxremote.ssl=false -Djava.rmi.server.hostname=192.168.47.8
-XX:+HeapDumpOnOutOfMemoryError -XX:HeapDumpPath=/argus/logs/ -
XX:+PrintCommandLineFlags -XX:MinRAMPercentage=10.0 -XX:MaxRAMPercentage=90.0
      - OPERATING_SCHEDULE_SECONDS_SPENT_LIMIT
      - TIMETABLE_SECONDS_SPENT_LIMIT
      - TZ
    volumes:
      - planning-service-logs:/argus/logs
    logging:
      driver: "json-file"
      options:
        max-size: "200m"
        max-file: "10"
volumes:
  planning-gw-logs:
    driver: local
    driver_opts:
```

```
      o: bind
      type: none
      device: /argus/planning-gw/logs
  planning-service-logs:
    driver: local
    driver_opts:
      o: bind
      type: none
      device: /argus/planning-service/logs
```

3.2.6 Сервис отчетов

3.2.6.1 Проверка настройки программной среды

Необходимо убедиться, что требования из п. 3.1.6 выполнены.

3.2.6.2 Установка обновлений сервиса отчетов

При установке обновлений сервиса отчетов необходимо придерживаться следующего порядка действий

1. остановить контейнер.
2. скачать образ
3. обновить образ.
4. внести изменения в конфигурационные файлы[77].
5. запустить контейнер.
6. если пп.1-5 выполнены успешно, то удалить старый контейнер и образ старого контейнера.

Примеры команд[78]

Информация по текущим контейнерам на хосте

```
docker ps -a
```

Остановить одиночный контейнер

```
docker stop <container_id>
```

Или все контейнеры, описанные в docker-compose.yml

```
docker-compose stop
```

Скачать образ из удаленного репозитория Аргус в локальный репозиторий

```
docker pull gitlab:4567/laboratorium/report-service:release-<version>
```

В случае, если доступа к удалённому репозиторию Аргус нет, то на стороне Аргус необходимо скачать образ из удаленного репозитория Аргус в локальный репозиторий

```
docker pull gitlab:4567/laboratorium/report-service:release-<version>
```

Экспортировать образ

---

[77] .env; docker-compose.yml; json

[78] Все необходимые подробности по командам можно посмотреть в оригинальной документации по docker: https://docs.docker.com/engine/reference/commandline/docker/

```
docker image save -o lk-<version>.tar gitlab:4567/laboratorium/report-
service:release-<version>
```

После чего передать архив с образом контейнера (mobile-api-<version>.tar) заказчику

На стороне заказчика необходимо загрузить образ в локальный репозиторий docker

```
docker load -i report-service-<version>.tar
```

Внести изменения в конфигурационные файлы

.env

```
HOST_IP=192.168.47.6
RMI_PORT=9007
JAVA_OPTS=-Dcom.sun.management.jmxremote -Dcom.sun.management.jmxremote.port=9007 -
Dcom.sun.management.jmxremote.rmi.port=9007 -
Dcom.sun.management.jmxremote.local.only=false -
Dcom.sun.management.jmxremote.authenticate=false -
Dcom.sun.management.jmxremote.ssl=false -Djava.rmi.server.hostname=192.168.47.6 -
XX:+HeapDumpOnOutOfMemoryError -XX:HeapDumpPath=/argus/logs/ -
XX:+PrintCommandLineFlags -XX:MinRAMPercentage=10.0 -XX:MaxRAMPercentage=90.0

SPRING_APPLICATION_JSON="{"argus":{"reports":{"datamarts":[{"name":"MAIN_DB","url":"
jdbc:postgresql://192.168.47.5:5432/demodb?currentSchema=system","main":true}]}}}"
REPORT_DB_ADDR=192.168.47.5:5432
REPORT_DB_NAME=demodb
TZ=Europe/Moscow
```

docker-compose.yml

```
version: "3"
services:
report-service:
container_name: report-service
image: gitlab:4567/laboratorium/report-service:release-1.8.0
ports:
- "9000:8080"
- "9007:9007"
restart: always
environment:
- REPORT_DB_ADDR
- REPORT_DB_NAME
- SPRING_APPLICATION_JSON
- SENTRY_DSN
- SENTRY_ENVIRONMENT
- HOST_IP
- RMI_PORT
- JAVA_OPTS
- TZ

volumes:
- reports-storage:/argus/reports
- reports-logs:/argus/logs
- reports-plugins:/argus/plugins
logging:
driver: "json-file"
options:
max-size: "200m"
max-file: "10"

volumes:
reports-storage:
```

```
driver: local
driver_opts:
o: bind
type: none
device: /argus/reports/storage
reports-logs:
driver: local
driver_opts:
o: bind
type: none
device: /argus/reports/logs
reports-plugins:
driver: local
driver_opts:
o: bind
type: none
device: /argus/reports/plugins
```

db.json[79]

```
{
"argus": {
"reports": {
"datamarts": [
{
"name": "MAIN_DB",
"url": "jdbc:postgresql://192.168.47.5:5432/demodb?currentSchema=system",
"main": true
}
]
}
}
}
```

Запустить одиночный контейнер

```
docker start <container_id>
```

Или все контейнеры, описанные в docker-compose.yml[80]

```
docker-compose up -d
```

Проверить успешность запуска контейнера

```
docker ps -a

docker logs -f report-service
```

Удалить контейнер

```
docker rm <container_id>
```

Информация по текущим образам в репозитории

```
docker images
```

Удалить образ контейнера.

[79] "main": true необходимо указать для подключения плагина

[80] возможно передавать параметры для запуска в командной строке: SPRING_APPLICATION_JSON=$(cat db.json) docker-compose up -d

```
docker rmi <image_id>
```

3.2.6.3 Запуск сервиса

Запустить все контейнеры, описанные в ***docker-compose.yml***

```
SPRING_APPLICATION_JSON=$(cat db.json) docker-compose up -d
```

Проверить успешность запуска контейнера

```
docker ps -a

docker logs -f report-service
```

3.2.6.4 Остановка сервиса

Информация по текущим контейнерам на хосте

```
docker ps -a
```

Остановить одиночный контейнер

```
docker stop <container_id>
```

Или все контейнеры, описанные в ***docker-compose.yml***

```
docker-compose stop
```

3.2.6.5 Файлы конфигурации

.env[81]

Пример файла конфигурации

```
HOST_IP=192.168.47.6
SPRING_APPLICATION_JSON={"argus":{"reports":{"datamarts":[{"name":"MAIN_DB","url
":"jdbc:postgresql://192.168.47.5:5432/demodb?currentSchema=system","main":true}
]}}}
REPORT_DB_ADDR=192.168.47.5:5432
REPORT_DB_NAME=demodb
TZ=Europe/Moscow
```

docker-compose.yml

Пример файла конфигурации

```
version: "3"
services:
  report-service:
    container_name: report-service
    image: gitlab:4567/laboratorium/report-service:release-1.8.0
    ports:
```

[81] "main": true необходимо указать для подключения плагина

```
      - "9000:8080"
      - "9007:9007"
    restart: always
    environment:
      - REPORT_DB_ADDR
      - REPORT_DB_NAME
      - SPRING_APPLICATION_JSON
      - SENTRY_DSN
      - SENTRY_ENVIRONMENT
      - HOST_IP
      - RMI_PORT=9007
      - JAVA_OPTS=-Dcom.sun.management.jmxremote -
Dcom.sun.management.jmxremote.port=9007 -
Dcom.sun.management.jmxremote.rmi.port=9007 -
Dcom.sun.management.jmxremote.local.only=false -
Dcom.sun.management.jmxremote.authenticate=false -
Dcom.sun.management.jmxremote.ssl=false -Djava.rmi.server.hostname=192.168.47.6
-XX:+HeapDumpOnOutOfMemoryError -XX:HeapDumpPath=/argus/logs/ -
XX:+PrintCommandLineFlags -XX:MinRAMPercentage=10.0 -XX:MaxRAMPercentage=90.0
      - TZ
    volumes:
      - reports-storage:/argus/reports
      - reports-logs:/argus/logs
      - reports-plugins:/app/plugins
    logging:
      driver: "json-file"
      options:
        max-size: "200m"
        max-file: "10"
volumes:
  reports-storage:
    driver: local
    driver_opts:
      o: bind
      type: none
      device: /argus/reports/storage
  reports-logs:
    driver: local
    driver_opts:
      o: bind
      type: none
      device: /argus/reports/logs
  reports-plugins:
    driver: local
    driver_opts:
      o: bind
      type: none
      device: /argus/reports/plugins
```

3.2.7 Сервис уведомлений

3.2.7.1 Проверка настройки программной среды

Необходимо убедиться, что требования из п. 3.1.7 выполнены.

3.2.7.2 Установка обновлений сервиса уведомлений

При установке обновлений сервиса Мобильный API WFM CC необходимо придерживаться следующего порядка действий

1. остановить контейнер.
2. скачать образ
3. обновить образ.
4. внести изменения в конфигурационные файлы[82]
5. запустить контейнер.
6. если пп.1-5 выполнены успешно, то удалить старый контейнер и образ старого контейнера.

Примеры команд[83]

Информация по текущим контейнерам на хосте

```
docker ps -a
```

Остановить одиночный контейнер

```
docker stop <container_id>
```

Или все контейнеры, описанные в docker-compose.yml

```
docker-compose stop
```

Скачать образ из удаленного репозитория Аргус в локальный репозиторий

```
docker pull gitlab:4567/laboratorium/notification-service:release-<version>
```

В случае, если доступа к удалённому репозиторию Аргус нет, то на стороне Аргус необходимо скачать образ из удаленного репозитория Аргус в локальный репозиторий

```
docker pull gitlab:4567/laboratorium/notification-service:release-<version>
```

Экспортировать образ

```
docker image save -o lk-<version>.tar gitlab:4567/laboratorium/notification-
service:release-<version>
```

После чего передать архив с образом контейнера (notification-service-<version>.tar) заказчику на стороне заказчика необходимо загрузить образ в локальный репозиторий docker

```
docker load -i notification-service-<version>.tar
```

Внести изменения в конфигурационные файлы

.env

```
HOST_IP=192.168.47.7
RMI_PORT=9027
JAVA_OPTS=-Dcom.sun.management.jmxremote -Dcom.sun.management.jmxremote.port=9027 -
Dcom.sun.management.jmxremote.rmi.port=9027 -
Dcom.sun.management.jmxremote.local.only=false -
Dcom.sun.management.jmxremote.authenticate=false -
Dcom.sun.management.jmxremote.ssl=false -Djava.rmi.server.hostname=192.168.47.7 -
```

[82] .env; docker-compose.yml

[83] все необходимые подробности по командам можно посмотреть в оригинальной документации по docker: https://docs.docker.com/engine/reference/commandline/docker/

```
XX:+HeapDumpOnOutOfMemoryError -XX:HeapDumpPath=/argus/logs/ -
XX:+PrintCommandLineFlags -XX:MinRAMPercentage=10.0 -XX:MaxRAMPercentage=90.0
NOTIFICATION_DB_ADDR=192.168.47.5:5432
NOTIFICATION_DB_NAME=demodb
TZ=Europe/Moscow
MAIL_FROM=wfmcc@argustelecom.ru
MAIL_SMTP_HOST=mail.argustelecom.ru
MAIL_SMTP_PORT=25
MAIL_SMTP_USER=user
MAIL_SMTP_PASS=pass
```

docker-compose.yml

```
version: "3"
services:
notification-service:
container_name: notification-service
image: gitlab:4567/laboratorium/notification-service:release-1.0.2
ports:
- "9020:8080"
- "9027:9027"
restart: always
environment:
- NOTIFICATION_DB_ADDR
- NOTIFICATION_DB_NAME
- HOST_IP
- RMI_PORT
- JAVA_OPTS
- TZ
- MAIL_FROM
- MAIL_SMTP_HOST
- MAIL_SMTP_PORT
- MAIL_SMTP_USER
- MAIL_SMTP_PASS
#- MAIL_ENABLED=false

volumes:
- notification-service-logs:/argus/logs
logging:
driver: "json-file"
options:
max-size: "200m"
max-file: "10"

volumes:
notification-service-logs:
driver: local
driver_opts:
o: bind
type: none
device: /argus/notification-service/logs
```

Запустить одиночный контейнер

```
docker start <container_id>
```

или все контейнеры, описанные в docker-compose.yml

```
docker-compose up -d
```

Проверить успешность запуска контейнера

```
docker ps -a
```

```
docker logs -f notification-service
```

Удалить контейнер

```
docker rm <container_id>
```

Информация по текущим образам в репозитории

```
docker images
```

Удалить образ контейнера.

```
docker rmi <image_id>
```

Для работы через прокси-сервер

1. в файле .env необходимо добавить

    1.1 переменные с адресом прокси-сервера

    HTTP_PROXY

    HTTPS_PROXY

    и хостами\диапазону сети исключенными из проксирования

    NO_PROXY

    1.2 к переменной JAVA_OPTS необходимо добавить параметры с адресом прокси-сервера и его портом

    -Dhttp.proxyHost

    -Dhttp.proxyPort

    -Dhttps.proxyHost

    -Dhttps.proxyPort.

2. в файл docker-compose.yml добавить параметры

HTTP_PROXY

HTTPS_PROXY

NO_PROXY

3.2.7.3 Запуск сервиса

Запустить одиночный контейнер

```
docker start <container_id>
```

Или все контейнеры, описанные в ***docker-compose.yml***

```
docker-compose up -d
```

Проверить успешность запуска контейнера

```
docker ps -a
```

```
docker logs -f notification-service
```

3.2.7.4 Остановка сервиса

Информация по текущим контейнерам на хосте

```
docker ps -a
```

Остановить одиночный контейнер

```
docker stop <container_id>
```

Или все контейнеры, описанные в ***docker-compose.yml***

```
docker-compose stop
```

3.2.7.5 Файлы конфигурации

.env

Пример файла конфигурации

```
HOST_IP=192.168.47.7
NOTIFICATION_DB_ADDR=192.168.47.5:5432
NOTIFICATION_DB_NAME=demodb
TZ=Europe/Moscow
MAIL_FROM=wfmcc@argustelecom.ru
MAIL_SMTP_HOST=mail.argustelecom.ru
MAIL_SMTP_PORT=25
MAIL_SMTP_USER=user
MAIL_SMTP_PASS=***
```

docker-compose.yml

Пример файла конфигурации

```
version: "3"
services:
  notification-service:
    container_name: notification-service
    image: gitlab:4567/laboratorium/notification-service:release-1.0.2
    ports:
      - "9020:8080"
      - "9027:9027"
    restart: always
    environment:
      - NOTIFICATION_DB_ADDR
      - NOTIFICATION_DB_NAME
      - HOST_IP
      - RMI_PORT=9027
      - JAVA_OPTS=-Dcom.sun.management.jmxremote -
Dcom.sun.management.jmxremote.port=9027 -
Dcom.sun.management.jmxremote.rmi.port=9027 -
Dcom.sun.management.jmxremote.local.only=false -
Dcom.sun.management.jmxremote.authenticate=false -
```

```
Dcom.sun.management.jmxremote.ssl=false -Djava.rmi.server.hostname=192.168.47.7
-XX:+HeapDumpOnOutOfMemoryError -XX:HeapDumpPath=/argus/logs/ -
XX:+PrintCommandLineFlags -XX:MinRAMPercentage=10.0 -XX:MaxRAMPercentage=90.0
      - TZ
      - MAIL_FROM
      - MAIL_SMTP_HOST
      - MAIL_SMTP_PORT
      - MAIL_SMTP_USER
      - MAIL_SMTP_PASS
      #- MAIL_ENABLED=false
    volumes:
      - notification-service-logs:/argus/logs
    logging:
      driver: "json-file"
      options:
        max-size: "200m"
        max-file: "10"
volumes:
  notification-service-logs:
    driver: local
    driver_opts:
      o: bind
      type: none
      device: /argus/notification-service/logs
```

для работы СУ через прокси-сервер

1. в файле .env необходимо добавить

    1.1 переменные с адресом прокси-сервера

    HTTP_PROXY

    HTTPS_PROXY

    и хостами\диапазону сети исключенными из проксирования

    NO_PROXY

    1.2 к переменной JAVA_OPTS необходимо добавить параметры с адресом прокси-сервера и его портом

    -Dhttp.proxyHost

    -Dhttp.proxyPort

    -Dhttps.proxyHost

    -Dhttps.proxyPort.

2. в файл docker-compose.yml добавить параметры

HTTP_PROXY

HTTPS_PROXY

NO_PROXY

3.2.7.6 Настройка почтовых уведомлений

Для настройки почтовых уведомлений необходимо указать необходимые параметры в файлах .env и docker-compose.yml

например

.env

MAIL_FROM=wfmcc@argustelecom.ru

MAIL_SMTP_HOST=mail.argustelecom.ru

MAIL_SMTP_PORT=25

MAIL_SMTP_USER=user

MAIL_SMTP_PASS=***

docker-compose.yml

- MAIL_FROM
- MAIL_SMTP_HOST
- MAIL_SMTP_PORT
- MAIL_SMTP_USER
- MAIL_SMTP_PASS

- MAIL_ENABLED=false

см. п. 3.2.7.5 Файлы конфигурации

Раскомментированный параметр MAIL_ENABLED=false означает отключение почтовых уведомлений

Чтобы изменения применились, необходимо перезапустить сервис

см. п. 3.2.7.4 Остановка сервиса и п. 3.2.7.3 Запуск сервиса

3.2.8 Сервис интеграций

3.2.8.1 Проверка настройки программной среды

Необходимо убедиться, что требования из п. 3.1.8 выполнены.

3.8.2.2 Распаковка архива пакета установки

Распаковать архив пакета установки на хосте СП, используя утилиту ***unzip***:[84]

```
unzip [имя_архива_пакета_установки].zip
```

Архив должен быть распакован в каталоге **/argus/integration**.

Владельцем приложения должна быть учетная запись ***argus***.

Пример:

```
chown argus:argus /argus/integration/integration-0.0.46-SNAPSHOT.jar
chmod 500 /argus/integration/integration-0.0.46-SNAPSHOT.jar
```

[84] Архив должен быть распакован именно на хосте СП, а не на локальной машине администратора, В противном случае, передача распакованного архива по сети (с помощью протокола SCP/FTP или им подобного) – приведет к проблеме с русскими буквами в именах файлов.

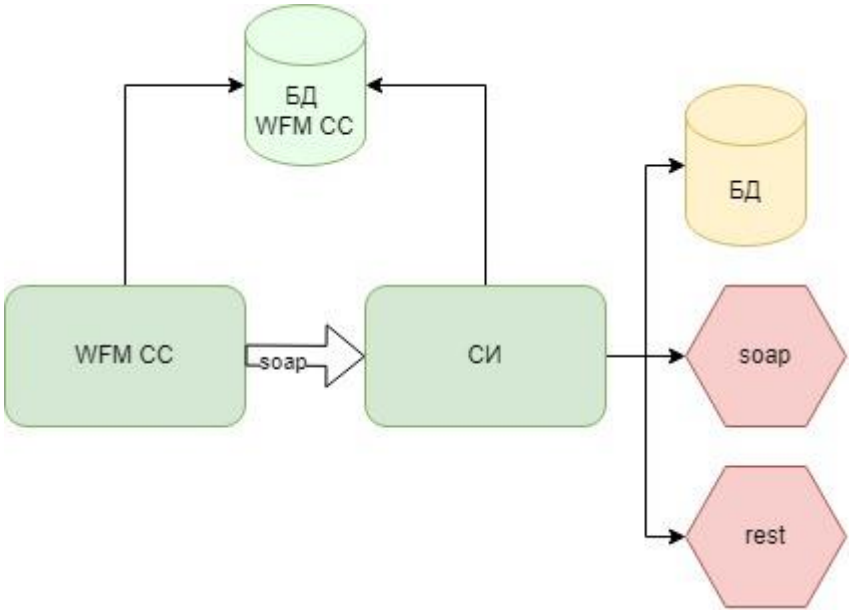
3.2.8.3 Настройка сервиса интеграций

Общая информация

СП WFM CC отправляет запрос к СИ.

СИ может запрашивать данные из БД Заказчика сам

СИ может делегировать получение и обработку данных в системы Заказчика. Формат обмена данными может быть либо soap, либо rest.



Справочник "Интеграционные системы"

Для подключения к СИ заполняется справочник "Интеграционные системы", в котором указывается

- название системы
- идентификатор системы (должен соответствовать значению, указанному в yaml файле)
- точки доступа для получения данных

Формат записи для точек доступа имеет следующий вид:

- **http://***<адрес СИ>:<порт СИ>***/services/***<название метода>***?wsdl**, где

<адрес СИ> - предоставляется Заказчиком, указывается в ТА (технической архитектуре)

<порт СИ> - предоставляется Заказчиком, указывается в ТА (технической архитектуре)

<название метода> - перечислены в таблице ниже:

Название метода	**Название столбца в таблице справочника "Интеграционные системы"**	**Пример**
personnel	Точка доступа для получения структуры персонала	http://192.168.111.222:8091/services/personnel?wsdl
historicData	Точка доступа для получения исторических данных работы КЦ	http://192.168.111.222:8091/services/historicData?wsdl

Название метода	Название столбца в таблице справочника "Интеграционные системы"	Пример
historicOperatorStatus	Точка доступа для получения исторических данных операторов	http://192.168.111.222:8091/services/historicOperatorStatus?wsdl
workTimeChats	Точка доступа для получения работы в чатах операторов	http://192.168.111.222:8091/services/workTimeChats?wsdl
monitoring	Точка доступа для получения данных мониторинга	http://192.168.111.222:8091/services/monitoring?wsdl

Yaml-файл

Общие настройки.

Параметры	Пример
<название системы>: enable: *false/true* system-id: *<значение>* (то же значние, что и в справочнике "Интеграционные системы)	genesys: enable: true system-id: GENESYS
server: port: *<значение>* (тот же порт, что и в справочнике "Интеграционные системы")	server: port: 8091

Если СИ запрашивает данные Заказчика сам, то в таком случае в yaml-файле указываются настройки к БД Заказчика.

datasource:

- url - адрес БД, к которой обращается СИ (предоставляется Заказчиком, указывается в ТА)
- username - название БД, к которой обращается СИ (предоставляется Заказчиком, указывается в ТА)
- password - пароль БД, к которой обращается СИ (предоставляется Заказчиком, указывается в ТА)

Если СИ делегирует получение и обработку данных Заказчику, то в таком случае, в зависимости от формата обмена данными (soap или rest), в yaml-файле настраиваются endpoint к Заказчику. Названия endpoint указываются и согласовываются в ТЗ.

В yaml-файле также указывается адрес и порт системы, к которой обращается СИ

- base-url: http://*<адрес>*:*<порт>*

Пример yaml-файла

application.yaml

```
integration:
naumen:
enable: true
system-id: NCC
base-url: http://192.168.100.30:8888/soap/ # URL к стороннему сервису
```

```
receive-timeout: 120000 # время ожидания ответа в мс
1c:
enable: true
system-id: 1c
base-url: http://192.168.100.40/customer/hs/wfm/ # URL к стороннему сервису
username: WFMSystem # имя пользователя 1c
password: *** # пароль
logging-requests: true
logging:
file:
max-history: 90
name: log/integration.log
max-size: 10MB
level:
ru.argustelecom.ccwfm.integration.systems.onec: debug
```

Внесения изменений в конфигурационный файл[85] необходимо перезапустить сервис интеграции

```
systemctl restart integration
```

3.2.8.4 Установка обновлений сервиса интеграций

Для проведения обновлений сервиса интеграций необходимо придерживаться следующего порядка действий

1. остановить сервис
2. сделать резервную копию исходного дистрибутива и конфигурационного файла
3. установить новый дистрибутив и выдать на него права для учетной записи argus
4. внести изменения в конфигурационный файл (при необходимости)
5. отредактировать файл автозапуска и обновить конфигурацию systemd
6. запустить сервис

Примеры команд

Остановить сервис интеграции

```
systemctl stop integration
```

Сделать резервную копию исходного дистрибутива и конфигурационного файла

```
mv /argus/integration/integration-0.0.46-SNAPSHOT.jar  /argus/backup/

cp /argus/distr/integration-0.0.47-SNAPSHOT.zip /argus/integration/integration-
0.0.47-SNAPSHOT.zip
```

Установить новый дистрибутив и выдать на него права для учетной записи argus

```
cd /argus/integration/

unzip integration-0.0.47-SNAPSHOT.zip
```

```
chown argus:argus /argus/integration/integration-0.0.47-SNAPSHOT.jar
chmod 500 /argus/integration/integration-0.0.47-SNAPSHOT.jar
```

Отредактировать файл автозапуска

[85] application.yaml

```
[Unit]
Description=integration
After=syslog.target


[Service]
User=argus
WorkingDirectory=/argus/integration
ExecStart=/argus/integration/integration-0.0.47-SNAPSHOT.jar
SuccessExitStatus=143
TimeoutStopSec=10
Restart=on-failure
RestartSec=5
OOMScoreAdjust=-1000


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

Обновить конфигурацию systemd

```
systemctl daemon-reload
```

Запустить сервис интеграции

```
systemctl start integration
```

3.2.8.5 Запуск сервиса

Запустить сервис интеграции

```
systemctl start integration
```

Проверить успешность запуска сервиса

```
systemctl status integration
```

3.2.8.6 Остановка сервиса

Остановить сервис интеграции

```
systemctl stop integration
```

Проверить успешность остановки сервиса

```
systemctl status integration
```

3.2.8.7 Файлы конфигурации

Пример конфигурационного yaml-файла

application.yaml

```
integration:
naumen:
enable: true
system-id: NCC
```

```
base-url: http://192.168.100.30:8888/soap/ # URL к стороннему сервису
receive-timeout: 120000 # время ожидания ответа в мс
1c:
enable: true
system-id: 1c
base-url: http://192.168.100.40/customer/hs/wfm/ # URL к стороннему сервису
username: WFMSystem # имя пользователя 1c
password: *** # пароль
logging-requests: true
logging:
file:
max-history: 90
name: log/integration.log
max-size: 10MB
level:
ru.argustelecom.ccwfm.integration.systems.onec: debug
```

3.2.9 Сервис мониторинга

3.2.9.1 Проверка настройки программной среды

Необходимо убедиться, что требования из п. 3.1.9 выполнены.

3.2.9.2 Установка обновлений сервиса мониторинга

При установке обновлений сервиса мониторинга необходимо придерживаться следующего порядка действий

1. остановить контейнер.
2. скачать образ
3. обновить образ.
4. внести изменения в конфигурационные файлы [1].
5. запустить контейнер.
6. если пп.1-5 выполнены успешно, то удалить старый образ контейнера.

Примеры команд [2]

информация по текущим контейнерам на хосте

```
docker ps -a
```

остановить одиночный контейнер

```
docker stop <container_id>
```

или все контейнеры, описанные в docker-compose.yml

```
docker-compose stop
```

скачать образ из удаленного репозитория Аргус в локальный репозиторий

```
docker pull gitlab:4567/laboratorium/monitoring-data-endpoint:WFMCC-<version>
```

в случае, если доступа к удалённому репозиторию Аргус нет, то на стороне Аргус необходимо скачать образ из удаленного репозитория Аргус в локальный репозиторий

```
docker pull gitlab:4567/laboratorium/monitoring-data-endpoint:WFMCC-<version>
```

экспортировать образ

```
docker image save -o lk-<version>.tar gitlab:4567/laboratorium/monitoring-data-endpoint:WFMCC-<version>
```

после чего передать архив с образом контейнера (monitoring-data-endpoint-<version>.tar) заказчику

на стороне заказчика необходимо загрузить образ в локальный репозиторий docker

```
docker load -i monitoring-data-endpoint-<version>.tar
```

внести изменения в конфигурационные файлы .env и docker-compose.yml (обновить версию сервиса, добавить/удалить/изменить параметры при необходимости)

Пример конфигов. Конфиг зависит от сборки сервиса мониторинга

.env

```
DB_ADDR=192.168.47.9:5432

DB_NAME=demodb

DB_SCHEMA=monitoring_service

DB_USERNAME=argus_sys

DB_PASS=passwd

STATUSES_REQ_URL=http://domen:8085/statuses/

PROCESSOR=default

MODE=active

SERVICE_ID=Protei

HOST_IP=192.168.47.7

TZ=Europe/Moscow
```

```
version: "3"
services:
  monitoring-data-endpoint:
    container_name: monitoring-data-endpoint
    image: gitlab:4567/laboratorium/monitoring-data-endpoint:dev
    ports:
      - "9060:8080"
      - "9067:9067"
    restart: always
    environment:
      - DB_ADDR
      - DB_NAME
      - DB_SCHEMA
      - DB_USERNAME
      - DB_PASS
      - HOST_IP
      - RMI_PORT=9067
      - JAVA_OPTS=-Dcom.sun.management.jmxremote -Dcom.sun.management.jmxremote.port=9067
-Dcom.sun.management.jmxremote.rmi.port=9067 -
Dcom.sun.management.jmxremote.local.only=false -
Dcom.sun.management.jmxremote.authenticate=false -Dcom.sun.management.jmxremote.ssl=false
-Djava.rmi.server.hostname=192.168.47.7 -XX:+HeapDumpOnOutOfMemoryError -
XX:HeapDumpPath=/argus/logs/ -XX:+PrintCommandLineFlags -XX:MinRAMPercentage=10.0 -
XX:MaxRAMPercentage=90.0 -
Xlog:gc*,gc+age=trace,safepoint:file=/argus/logs/gcstats.log.`date +%Y-%m-%d-%H-
%M`:tags,uptime,time,level:filecount=10,filesize=100M -XX:+UseParallelOldGC -
XX:+UseGCOverheadLimit -XX:GCTimeLimit=80 -XX:GCHeapFreeLimit=10
      - TZ
      - PROCESSOR
      - MODE
      - STATUSES_REQ_URL
      - SERVICE_ID
    volumes:
      - monitoring-data-endpoint-logs:/argus/logs
    logging:
      driver: "json-file"
      options:
        max-size: "200m"
```

```
        max-file: "10"


volumes:
  monitoring-data-endpoint-logs:
    driver: local
    driver_opts:
      o: bind
      type: none
      device: /argus/monitoring-data-endpoint/logs
```

запустить одиночный контейнер

```
docker start <container_id>
```

или все контейнеры, описанные в docker-compose.yml

```
docker-compose up -d
```

проверить успешность запуска контейнера

```
docker ps -a

docker logs -f monitoring-data-endpoint
```

информация по текущим образам в репозитории

```
docker images
```

удалить образ контейнера.

```
docker rmi <image_id>
```

удалить контейнер (при необходимости)

```
docker rm <container_id>
```

---

[1] .env; docker-compose.yml

[2] все необходимые подробности по командам можно посмотреть в оригинальной документации по docker: https://docs.docker.com/engine/reference/commandline/docker/

3.2.9.3 Запуск сервиса

запустить одиночный контейнер

```
docker start <container_id>
```

или все контейнеры, описанные в ***docker-compose.yml***

```
docker-compose up -d
```

проверить успешность запуска контейнера

```
docker ps -a

docker logs -f monitoring-data-endpoint
```

3.2.9.4 Остановка сервиса

информация по текущим контейнерам на хосте

```
docker ps -a
```

остановить одиночный контейнер

```
docker stop <container_id>
```

или все контейнеры, описанные в ***docker-compose.yml***

```
docker-compose stop
```

3.2.9.5 Файлы конфигурации

Пример конфигов. Конфиг зависит от сборки сервиса мониторинга

docker-compose.yml

```
version: "3"
services:
  monitoring-data-endpoint:
    container_name: monitoring-data-endpoint
    image: gitlab:4567/laboratorium/monitoring-data-endpoint:dev
    ports:
      - "9060:8080"
      - "9067:9067"
    restart: always
    environment:
      - DB_ADDR
      - DB_NAME
      - DB_SCHEMA
      - DB_USERNAME
      - DB_PASS
      - HOST_IP
      - RMI_PORT=9067
      - JAVA_OPTS=-Dcom.sun.management.jmxremote -Dcom.sun.management.jmxremote.port=9067
-Dcom.sun.management.jmxremote.rmi.port=9067 -
Dcom.sun.management.jmxremote.local.only=false -
Dcom.sun.management.jmxremote.authenticate=false -Dcom.sun.management.jmxremote.ssl=false
-Djava.rmi.server.hostname=192.168.47.7 -XX:+HeapDumpOnOutOfMemoryError -
XX:HeapDumpPath=/argus/logs/ -XX:+PrintCommandLineFlags -XX:MinRAMPercentage=10.0 -
XX:MaxRAMPercentage=90.0 -
Xlog:gc*,gc+age=trace,safepoint:file=/argus/logs/gcstats.log.`date +%Y-%m-%d-%H-
%M`:tags,uptime,time,level:filecount=10,filesize=100M -XX:+UseParallelOldGC -
XX:+UseGCOverheadLimit -XX:GCTimeLimit=80 -XX:GCHeapFreeLimit=10
      - TZ
      - PROCESSOR
      - MODE
      - STATUSES_REQ_URL
      - SERVICE_ID
    volumes:
      - monitoring-data-endpoint-logs:/argus/logs
    logging:
      driver: "json-file"
      options:
        max-size: "200m"
```

```
        max-file: "10"


volumes:
  monitoring-data-endpoint-logs:
    driver: local
    driver_opts:
      o: bind
      type: none
      device: /argus/monitoring-data-endpoint/logs
```

.env

```
DB_ADDR=192.168.47.9:5432
DB_NAME=demodb
DB_SCHEMA=monitoring_service
DB_USERNAME=argus_sys
DB_PASS=passwd
STATUSES_REQ_URL=http://domen:8085/statuses/
PROCESSOR=default
MODE=active
SERVICE_ID=Protei
HOST_IP=192.168.47.7
TZ=Europe/Moscow
```

3.2.10 Балансировщик СП

3.2.10.1 Веб-сервер apache (HTTPD)

Установка ПО на базе веб-сервера ***apache*** (HTTPD) производится из пакетов ОС Linux или же дистрибутив скачивается с сайта разработчика: https://httpd.apache.org/download.cgi.

В комплект поставки веб-сервера ***apache*** (HTTPD) входят модуль:

- ***mod_cache***

В случае использования дополнительных модулей – их нужно скачать отдельно:

- ***mod_jk*** с сайта http://tomcat.apache.org/download-connectors.cgi;
- ***mod_cluster***[86] с сайта http://mod-cluster.jboss.org/downloads.

[86] Mod_cluster, в отличие от других реализаций, позволяет на стороне балансировщика всегда иметь актуальную информацию не только о принципиальной доступности узла, но и детальную информацию о его техническом состоянии (CPU, потребление памяти, внутренних ресурсов СП и т.д).

3.2.10.1.1 Настройка кэширования статических ресурсов при помощи httpd + mod_cache

Кэширование применяется для получения и хранения статических ресурсов (рисунки, скрипты, страницы) на *front-end* сервере, к которому обращаются пользователи.

Целью кэширования является снижение нагрузки на *back-end* сервера (СП), повышение скорости ответа веб-страниц, уменьшение сетевого трафика.

Кэширование может быть организовано как на балансировщике, так и в виде веб-сервера, расположенного на отдельном хосте перед балансировщиком (или сервером приложений) и выполняющего только функции кэширования.

Настройки кэширования статических ресурсов прописываются в конфигурационном файле: ***cache-jk.conf***.

Файл конфигурации кэша ***cache-jk.conf*** подключается к основному конфигурационному файлу ***apache*** (HTTPD): ***conf/httpd.conf*** при помощи директивы ***Include***.

Необходимо учитывать, что путь к конфигурации указывается относительно атрибута веб-сервера: ***ServerRoot***.

Пример: ***Include conf/extra/cache-jk.conf***

Настройка кэша не требует дополнительной настройки со стороны СП.

Пример конфигурационных файлов для ***httpd + mod_cache*** поставляется в составе СП Аргус: ***INSTALL_PATH/loadbalancing/examples/mod_cache/httpd-2.2/conf/extra /cache-jk.conf***

и

INSTALL_PATH/loadbalancing/examples/mod_cache/httpd-2.4/conf/extra /cache-jk.conf

Листиг 3.2.9.1.1_1

Пример ***cache-jk.conf*** для ***httpd-2.2***

```
<IfModule mem_cache_module>
CacheEnable mem /argus/javax.faces.resource/
CacheEnable mem /javax.faces.resource/

CacheIgnoreCacheControl On

CacheDefaultExpire 28800
CacheMaxExpire 86400

CacheIgnoreHeaders Set-Cookie

CacheIgnoreNoLastMod On

CacheStoreNoStore On

CacheStorePrivate On
```

```
MCacheSize 10240

MCacheMaxObjectCount 5000

MCacheMinObjectSize 1

MCacheMaxObjectSize 100000

CacheIgnoreURLSessionIdentifiers argus_v
</IfModule>
```

В версии ***httpd-2.4*** изменен состав модулей, отвечающих за представление кэша.

Модуль ***mod_mem_cache*** был удалён из поставки ***httpd-2.4***.

Его функции переданы имплементациям ***mod_cache_socache***.

Подробнее с отличиями ***httpd-2.4*** от ***httpd-2.2*** можно ознакомиться: https://httpd.apache.org/docs/trunk/upgrading.html

3.2.10.1.2 Настройка балансировщика httpd + mod_jk

Балансировщик ***httpd + mod_jk*** должен быть настроен в соответствии с п. *2.1.10.3 Требования к балансировщику нагрузки СП*.

Пример конфигурационных файлов для ***httpd+mod_jk*** поставляется в составе СП Аргус.

INSTALL_PATH/tools/loadbalancing/examples/mod_jk/extra/

httpd.conf

В конфигурационном файле должны быть указаны загружаемые модули:

LoadModule jk_module modules/mod_jk.so

http://mod_jk.so/Листиг 3.2.9.1.2_1

Пример ***httpd-jk.conf***

```
<IfModule jk_module>
JkWorkersFile ./conf.d/workers.properties

JkLogFile "|/usr/sbin/rotatelogs /var/log/httpd/mod_jk.log 86400"

JkLogLevel info

JkShmFile /var/log/httpd/mod_jk.shm

LogLevel info
LogFormat "%t %a %{JK_WORKER_ROUTE}n:
%{JK_LB_LAST_NAME}n(%{JK_LB_LAST_STATE}n/%{JK_LB_LAST_BUSY}n) -
```

```
%{pid}P-%{tid}P %{JSESSIONID}C %r %s %B %D %{Referer}i \"%{User-Agent}i\""
jk_access_log


CustomLog "logs/jk_access_log" jk_access_log


JkWatchdogInterval 60


</IfModule>
```

Настройка логирования, указание местонахождения ***workers.properties***

Примеры конфигурационных файлов:

main-jk-host.conf

remoteadm-jk-host.conf

workers.properties

Листиг 3.2.9.1.2_2

Пример ***main-jk-host.conf***

```
Listen 80
Listen 443


<VirtualHost *:80>
<Location /jk-status>
JkMount jk-status
Order Allow,Deny
Allow from all
</Location>
<Location /jk-manager>
JkMount jk-manager
Order Allow,Deny
Allow from all
</Location>
</VirtualHost>


<VirtualHost *:443>
ServerAdmin webmaster@localhost
JkMount /argus/* main-balancer
</VirtualHost>
```

Листиг 3.2.9.1.2_3

Пример ***remoteadm-jk-host.conf***.

```
Listen 8443
<VirtualHost *:8443>
ServerAdmin webmaster@localhost
```

```
JkMount /* remotearm-balancer
</VirtualHost>
```

Листиг 3.2.9.1.2_4

Пример ***workers.properties***

```
worker.list=main-balancer,remotearm-balancer,jk-status,jk-manager

################################################
## STATUS WORKER
################################################
# Define two status worker:
# - jk-status for read-only use
# - jk-manager for read/write use
worker.jk-status.type=status
worker.jk-status.read_only=true
worker.jk-manager.type=status

################################################
## MAIN-BALANCER WORKER
################################################
worker.main-balancer.balance_workers=argusapp1_8009,argusapp2_8009
worker.main-balancer.reference=worker.balancer.template

################################################
## REMOTEARM-BALANCER WORKER
################################################
worker.remotearm-balancer.balance_workers=argusapp1_8119,argusapp2_8119
worker.remotearm-balancer.reference=worker.balancer.template

################################################
## BALANCER WORKER TEMPLATE
################################################
worker.balancer.template.type=lb
worker.balancer.template.method=B
worker.balancer.template.max_reply_timeouts=30
worker.balancer.template.recover_time=600
worker.balancer.template.error_escalation_time=0

################################################
## MAIN NODE WORKERS
```

```
################################################
worker.argusapp1_8009.reference=worker.template
worker.argusapp1_8009.host=192.168.100.180
worker.argusapp1_8009.port=8009
worker.argusapp1_8009.activation=A
worker.argusapp1_8009.route=192.168.100.180[0]

worker.argusapp2_8009.reference=worker.template
worker.argusapp2_8009.host=192.168.100.181
worker.argusapp2_8009.port=8009
worker.argusapp2_8009.activation=A
worker.argusapp2_8009.route=192.168.100.181[0]

################################################
## REMOTEARM NODE WORKERS
################################################
worker.argusapp1_8119.reference=worker.template
worker.argusapp1_8119.host=192.168.100.180
worker.argusapp1_8119.port=8119
worker.argusapp1_8119.activation=A
worker.argusapp1_8119.route=192.168.100.180[0]

worker.argusapp2_8119.reference=worker.template
worker.argusapp2_8119.host=192.168.100.181
worker.argusapp2_8119.port=8119
worker.argusapp2_8119.activation=A
worker.argusapp2_8119.route=192.168.100.181[0]

################################################
## NODE WORKER TEMPLATE
################################################
worker.template.type=ajp13
worker.template.socket_keepalive=true
worker.template.connection_pool_minsize=0
worker.template.connection_pool_timeout=600
worker.template.socket_connect_timeout=60000
worker.template.socket_timeout=1800
worker.template.reply_timeout=1500000
worker.template.ping_mode=CI
worker.template.connect_timeout=60000
worker.template.ping_timeout=60000
```

```
worker.template.connection_ping_interval=300
worker.template.retries=1
```

В качестве примера в конфигурационных файлах ***main-jk-host.conf*** и ***remoteadm-jk-host.conf*** указана настройка виртуальных хостов с доступом

- по порту 80 к ресурсам мониторинга состояния ***mod_jk***: ***jk-status***, ***jk-manager***
- по порту 443 к балансировщику через контекст ***/argus/**** для балансируемой группы ***main-balancer*** для доступа пользователей внутри КСПД.
- по порту 8443 к балансировщику через контекст **/** для балансируемой группы ***remotearm***

Для внешних пользователей, использующих мобильные устройства.

В файле ***workers.properties*** указаны

- название балансируемых групп

worker.list=main-balancer,remotearm

- состав балансируемых групп

worker.main-balancer.balance_workers=argusapp1_8009,argusapp2_8009

worker.remotearm-balancer.balance_workers=argusapp1_8119,argusapp2_8119

- тип балансировки, таймауты

worker.balancer.template.type=lb

worker.balancer.template.method=B

worker.balancer.template.max_reply_timeouts=30

worker.balancer.template.recover_time=600

worker.balancer.template.error_escalation_time=0

- настройки балансируемых групп: ip-адреса, порты, протоколы, таймауты

worker.argusapp1_8009.reference=worker.template

worker.argusapp1_8009.host=192.168.100.180

worker.argusapp1_8009.port=8009

worker.argusapp1_8009.activation=A

worker.argusapp1_8009.route=192.168.100.180[0]

worker.argusapp2_8009.reference=worker.template

worker.argusapp2_8009.host=192.168.100.181

worker.argusapp2_8009.port=8009

worker.argusapp2_8009.activation=A

worker.argusapp2_8009.route=192.168.100.181[0]

worker.argusapp1_8119.reference=worker.template

worker.argusapp1_8119.host=192.168.100.180

worker.argusapp1_8119.port=8119

worker.argusapp1_8119.activation=A

worker.argusapp1_8119.route=192.168.100.180[0]

worker.argusapp2_8119.reference=worker.template

worker.argusapp2_8119.host=192.168.100.181

worker.argusapp2_8119.port=8119
worker.argusapp2_8119.activation=A
worker.argusapp2_8119.route=192.168.100.181[0]

Для более детальной информации по настройке балансировщика ***httpd+mod_jk*** можно ознакомиться в документации, поставляемой в составе дистрибутива: *Руководство администратора по установке и резервированию reverse proxy для Сервера Приложений Аргус*.

3.2.10.1.3 Настройка балансировщика httpd + mod_cluster

Mod_cluster, в отличие от других реализаций, позволяет на стороне балансировщика всегда иметь

актуальную информацию не только о принципиальной доступности узла, но и детальную информацию о его техническом состоянии (cpu, потребление памяти, внутренние ресурсы сервера и т.д).

При работе с ***mod_cluster***

- поддерживается балансировка только одного порта, удаленный порт монтажника не поддерживается (до выхода версии СП WFLY-6803).
- настройка выполняется как на стороне балансировщика, так и на стороне СП

Настройки на стороне балансировщика.

Балансировщик ***httpd + mod_cluster*** должен быть настроен в соответствии с п. *2.1.10.3 Требования к балансировщику нагрузки СП*.

Пример конфигурационного файла для **httpd + mod_cluster** поставляется в составе СП Аргус: ***INSTALL_PATH/tools/loadbalancing/examples/mod_cluster/httpd.conf***.

Листиг 3.2.9.1.3_1

Пример ***httpd.conf***

```
ServerRoot "D:/temp/http-server/"
Listen *:7080
<IfModule !mpm_netware_module>
<IfModule !mpm_winnt_module>
User daemon
Group daemon
</IfModule>
</IfModule>
DocumentRoot "D:/temp/doc-root/"
<IfModule dir_module>
DirectoryIndex index.html
</IfModule>
ErrorLog "D:/temp/httpd-2.2/logs/error_log"
LogLevel warn
<IfModule log_config_module>
LogFormat "%h %l %u %t \"%r\" %>s %b \"%{Referer}i\" \"%{User-Agent}i\"" combined
```

```
LogFormat "%h %l %u %t \"%r\" %>s %b" common
<IfModule logio_module>
# You need to enable mod_logio.c to use %I and %O
LogFormat "%h %l %u %t \"%r\" %>s %b \"%{Referer}i\" \"%{User-Agent}i\" %I %O"
combinedio
</IfModule>
CustomLog "@rel_logfiledir@/access_log" common
</IfModule>
<IfModule alias_module>
ScriptAlias /cgi-bin/ "@exp_cgidir@/"
</IfModule>
<IfModule cgid_module>
</IfModule>
DefaultType text/plain
<IfModule mime_module>
TypesConfig @rel_sysconfdir@/mime.types
AddType application/x-compress .Z
AddType application/x-gzip .gz .tgz
</IfModule>
<IfModule ssl_module>
SSLRandomSeed startup builtin
SSLRandomSeed connect builtin
</IfModule>
LoadModule authz_host_module D:/temp/httpd-2.2/modules/mod_authz_host.so
LoadModule proxy_module D:/temp/httpd-2.2/modules/mod_proxy.so
LoadModule proxy_ajp_module D:/temp/httpd-2.2/modules/mod_proxy_ajp.so
LoadModule proxy_http_module D:/temp/httpd-2.2/modules/mod_proxy_http.so
LoadModule proxy_cluster_module D:/temp/httpd-2.2/modules/mod_proxy_cluster.so
LoadModule manager_module D:/temp/httpd-2.2/modules/mod_manager.so
LoadModule slotmem_module D:/temp/httpd-2.2/modules/mod_slotmem.so
LoadModule advertise_module D:/temp/httpd-2.2/modules/mod_advertise.so
<IfModule manager_module>
Listen *:7081
ManagerBalancerName argus-cluster
<VirtualHost *:7081>
<Directory />
Order deny,allow
Deny from none
Allow from all
</Directory>
KeepAliveTimeout 300
```

```
MaxKeepAliveRequests 0
#ServerAdvertise on http://@@MCMPIP@@:@@MCMPPORT@@
AdvertiseFrequency 5
#AdvertiseSecurityKey secret
#AdvertiseGroup @@ADVIP@@:23364
EnableMCPMReceive
<Location /mod_cluster_manager>
SetHandler mod_cluster-manager
Order deny,allow
Deny from none
Allow from all
</Location>
</VirtualHost>
</IfModule>
```

В конфигурационном файле должны быть указаны загружаемые модули:

LoadModule proxy_module modules/mod_proxy.

LoadModule proxy_ajp_module modules/mod_proxy_ajp.so

LoadModule slotmem_module modules/mod_slotmem.so

LoadModule manager_module modules/mod_manager.so

LoadModule proxy_cluster_module modules/mod_proxy_cluster.so

LoadModule advertise_module modules/mod_advertise.so

Должны быть определены следующие настройки:

Имя программного балансировщика: ***ManagerBalancerName argus-cluster***

Расположение веб-страницы управления***: <Location /mod_cluster_manager>***

Настройки на стороне Сервера приложений Аргус

Для взаимодействия узлов кластера с балансировщиком ***httpd + mod_cluster*** в файле конфигурации СП Аргус используются следующие настрой:

argus.modcluster.balancer.name - имя балансировщика, по которому его определяют узлы.

argus.modcluster.manager.address - адрес, где расположен ***mod_cluster_manager***. По данному адресу узел кластера сообщает информацию о своем состоянии.

argus.modcluster.manager.port - порт, на котором доступен ***mod_cluster_manager***.

Примечание:

Если значение ***argus.cluster.nodes*** останется пустым, а настройки ***argus.modcluster.**** заполнены, то во время установки СП Аргус будет получена ошибка:

Server configuration failed: Cluster not configured, but load-balancing enabled. Balancing wihout replication is dangerous and undesired!!!Server configuratioad-balancing enabled. Balancing wihout replication is dangerous and undesired!!!

При наличии данной ошибки – СП будет нерабочий.

Веб-интерфейс для управления ***mod_cluster*** доступен по url:

http://IP-адрес:порт/mod_cluster_manager, где

IP-адрес - значение параметра ***argus.modcluster.manager.address***,

Порт - значение параметра ***argus.modcluster.manager.port***.

Настройка ***Sticky Session***

Для настройки взаимодействия сервера приложений с **httpd + mod_cluster** необходимо отредактировать конфигурационный файл СП:

INSTALL_PATH/standalone/configuration/standalone.xml

или внести изменения в настройку **mod_cluster** через Admin Console СП

(вкладка Configuration -> поле Subsystems -> Web -> mod_cluster -> вкладка Sessions).

Изменение настроек ***mod_cluster*** должно выполняться на всех узлах кластера, и сами настройки должны быть идентичными, иначе полноценная работа кластера не гарантирована.

Листиг 3.2.9.1.3_2

Пример настройки ***sticky session*** в ***standalone.xml***

```
<subsystem xmlns="urn:jboss:domain:modcluster:1.2">
<mod-cluster-config proxy-list="192.168.100.75:7081" balancer=" argus-cluster" advertise="false"
connector="ajp" sticky-session="true" sticky-session-remove="false" sticky-session-force="false">
<dynamic-load-provider>
<load-metric type="cpu"/>
</dynamic-load-provider>
</mod-cluster-config>
</subsystem>
```

В теге ***mod-cluster-config*** указываются атрибуты:

Sticky Session - настройка для включения метода ***Sticky Session*** - метод балансировки нагрузки, при котором запросы сессии передаются на один и тот же узел кластера.

По умолчанию значение настройки ***true***.

Sticky Session Force - если значение **true**, то балансировщик всегда отправляет реквест сессии на

тот узел, с которым она связана, даже при возникновении сбоя на узле. Например, если на Сервере приложений Аргус-1 произошел сбой, то пользователь на посылаемый им реквест Серверу приложений Аргус-1 получит ошибку "сервис временно недоступен".

По умолчанию значение настройки **false**.

Sticky Session Remove - если значение **true**, при сбое на узле балансировщик отправляет реквесты

его пользовательских сессий на другие узлы кластера. Например, если на Сервере приложений Аргус-1 произошел сбой, реквесты текущих сессий перенаправляются балансировщиком на другой Сервер приложений Аргус.

По умолчанию значение настройки **false**.

3.2.10.1.4 Настройка безопасного соединения httpd + mod_ssl

Примеры конфигурации ***httpd + mod_proxy + mod_ssl*** можно найти в составе СП в каталоге ***tools/ssl/mod_proxy+mod_ssl***.

Переместить закрытый ключ и сертификат в каталог ***httpd***. Например, в ОС Linux:

- для сертификата каталог ***/etc/httpd/ssl.crt/***

- для приватного ключа каталог ***/etc/httpd/ssl.key/private/***

Пример настроек для включенного модуля ***mod_ssl*** в ***httpd***:

- Настройки кэша сеансов SSL (SSL Session Cache).

Место хранения кэша и его тип:

SSLSessionCache "shm:D:/httpd-2.2_/logs/ssl_scache(512000)"[87]

или

SSLSessionCache "shmcb:/var/log/httpd/ssl_scache(512000)"[88]

Время хранения кэша.

SSLSessionCacheTimeout 300

Семафор для внутреннего взаимного исключения операций. Используется значение по умолчанию.

SSLMutex default

Источник генератора псевдослучайных чисел (PRNG) для OpenSSL во время запуска и непосредственно перед установкой нового соединение SSL.

SSLRandomSeed startup builtin

или

SSLRandomSeed startup file:/dev/urandom

Для использования ***/dev/urandom*** в Linux должен быть обеспечены права доступа для УЗ, из-под которой запускается ***httpd***.

- Настройки для VirtualHost:

Настройка логов.

ErrorLog logs/ssl_error_log

TransferLog logs/ssl_access_log

LogLevel warn

Включение SSL.

SSLEngine onSSL

Включение поддерживаемых протоколов SSL. Все кроме SSL 2.

SSLProtocol all -SSLv2

Указать путь до сертификата.

SSLCertificateFile "/etc/ssl/certs/sert.crt"

Указать путь до приватного ключа.

SSLCertificateKeyFile "/etc/ssl/private/private.key"

---

[87] ОС Windows

[88] ОС Linux

Опции движка SSL, экспортируем переменные окружения SSL:

```
<Files ~ "\.(cgi|shtml|phtml|php3?)$">SSLOptions +StdEnvVars</Files><Directory "/var/www/cgi-bin">SSLOptions +StdEnvVars</Directory>
```

Настройка логов для реквестов:

```
CustomLog logs/ssl_request_log \"%t %h %{SSL_PROTOCOL}x %{SSL_CIPHER}x \"%r\" %b"
```

Перечень шифров OpenSSL используемые при согласовании ssl-соединения с клиентом.

SSLCipherSuite DEFAULT:!EXP:!SSLv2:!DES:!IDEA:!SEED:+3DES

3.2.10.2 Веб-сервер undertow

Веб-сервер ***undertow*** входит в состав СП WFM CC, как один из компонентов Wildfly и может использоваться качестве балансировщика с динамическим добавлением узлов кластера.

Для использования балансировщика на базе ***undertow*** достаточно установить СП WFM CC и убрать из приложения - файл ***INSTALL_PATH/standalone/deployments/argus-enterprise.ear***.

Балансировщик на базе ***undertow*** должен быть настроен в соответствии с должен быть настроен в соответствии с *п. 2.1.10.3 Требования к балансировщику нагрузки СП.*

Необходимые для ***undertow*** настройки указываются в конфигурационном файле: ***INSTALL_PATH/standalone/configuration/standalone.xml***

В тегах

```
<subsystem..>..</subsystem><socket-binding-group..>..</socket-binding-group>
```

Листинг 3.2.9.2_1

Пример фрагмента ***standalone.xml,*** поясняющего параметры настройки балансировщика в теге ***<subsystem>***

```
<subsystem xmlns="urn:jboss:domain:undertow:3.0" statistics-enabled="true">
<buffer-cache name="default"/>
<server name="default-server">
    <http-listener buffer-pool="view" max-post-size="104857600" name="default" no-request-timeout="1860000"
                read-timeout="1380000" record-request-start-time="true" request-parse-timeout="10000" socket-binding="http"
                tcp-keep-alive="true" write-timeout="1380000"/>
    <ajp-listener buffer-pool="view" max-post-size="104857600" name="ajp" no-request-timeout="1860000"
                read-timeout="1380000" record-request-start-time="true" request-parse-timeout="10000" socket-binding="ajp"
                tcp-keep-alive="true" write-timeout="1380000"/>
    <host alias="localhost" name="default-host">
        <location handler="welcome-content" name="/"/>
```

```
 <location name="/argus" handler="argus-handler"/>
            <filter-ref name="ie-disable-compatibility-header"/>
            <filter-ref name="gzip-
filter" predicate="regex[pattern='(?:application/javascript|text/css)(;.*)?',
                 value=%{o,Content-Type}, full-match=true]"/>
            <access-log pattern="%t %a %{i,X-Forwarded-For} %u [%I] %S %r %s
%{r,javax.servlet.error.status_code}
                %b %D %{i,Referer} "%{i,User-Agent}"" prefix="access." rotate="true"/>
      </host>
   </server>
   <server name="webui-mobile-server">
      <http-listener buffer-pool="view" max-cookies="10" max-header-size="10240" max-
headers="25" max-parameters="100"
            max-post-size="10485760" name="webui-mobile-default" no-request-
timeout="1860000" read-timeout="1380000"
            record-request-start-time="true" request-parse-timeout="10000" socket-binding="webui-
mobile-http"
            tcp-keep-alive="true" write-timeout="1380000"/>
      <ajp-listener buffer-pool="view" max-cookies="10" max-header-size="10240" max-
headers="25" max-parameters="100"
            max-post-size="65536" name="webui-mobile-ajp" no-request-timeout="1860000" read-
timeout="1380000"
            record-request-start-time="true" request-parse-timeout="10000" socket-binding="webui-
mobile-ajp"
            tcp-keep-alive="true" write-timeout="1380000"/>
      <host alias="localhost" name="default-host">
<location name="/" handler="mobile-handler"/>
            <filter-ref name="ie-disable-compatibility-header"/>
            <filter-ref name="gzip-filter"
            predicate="regex[pattern='(?:text/html|application/xhtml+xml|application/xml|application
/javascript|text/css)(;.*)?',
            value=%{o,Content-Type}, full-match=true]"/>
            <access-log pattern="%t %a %{i,X-Forwarded-For} %u [%I] %S %r %s
%{r,javax.servlet.error.status_code}
            %b %D %{i,Referer} "%{i,User-Agent}"" prefix="access_mobile." rotate="true"/>
      </host>
   </server>
   <servlet-container name="default">
      <jsp-config/>
   </servlet-container>
   <handlers>
      <file name="welcome-content" path="${jboss.home.dir}/welcome-content"/>
```

```
    <reverse-proxy name="argus-handler>
            <host name="host1" outbound-socket-binding="remote-default-
server1" scheme="ajp"
            path="/argus" instance-id="192_168_100_180[0]"/>
            <host name="host2" outbound-socket-binding="remote-default-
server2" scheme="ajp"
            path="/argus" instance-id="192_168_100_181[0]"/>
    </reverse-proxy>
    <reverse-proxy name="mobile-handler">
<host name="host1" outbound-socket-binding="remote-mobile-server1" scheme="ajp"
path="/ " instance-id="192_168_100_180[0]"/>
<host name="host2" outbound-socket-binding="remote-mobile-server2" scheme="ajp"
path="/ " instance-id="192_168_100_181[0]"/>
    </reverse-proxy>
  </handlers>
  <filters>
    <response-header header-name="X-UA-Compatible" header-value="IE=edge" name="ie-
disable-compatibility-header"/>
    <gzip name="gzip-filter"/>
  </filters>
</subsystem>
```

В главном теге ***subsystem undertow версии 3.0*** необходимо указать:

- для ***default-server*** в ***host*** указывается ***location***:

```
<location name="/argus" handler="argus-handler"/>
```

- для ***webui-mobile-server*** в ***host*** указывается ***location:***

```
<location name="/" handler="mobile-handler"/>
```

- в ***handlers*** указывается ***reverse-proxy***, отвечающий за балансировку контекста ***/argus*** основного порта:

```
<reverse-proxy name="argus-handler"/>
```

- для ***argus-handler*** указываются удаленные узлы, требующие балансировки:

```
<host name="host1" outbound-socket-binding="remote-default-
server1" scheme="ajp" path="/argus" instance-id="192_168_100_180[0]"/>
<host name="host2" outbound-socket-binding="remote-default-
server2" scheme="ajp" path="/argus" instance-id="192_168_100_181[0]"/>
```

- для ***argus-handler*** указываются удаленные узлы, требующие балансировки:

```
<host name="host1" outbound-socket-binding="remote-mobile-server1" scheme="ajp"
path="/ " instance-id="192_168_100_180[0]"/>
```

```
<host name="host2" outbound-socket-binding="remote-mobile-server2" scheme="ajp"
path="/ " instance-id="192_168_100_181[0]"/>
```

где:

name - имя узла (действует в пределах конфигурации);

outbound-socket-binding - название правила указывающего через какой порт подключаться к удаленного узлу (действует в пределах конфигурации);

scheme - схема подключения к удаленного узлу;

path - контекст удаленного узла отражаемый на балансировщике

instance-id - значение параметра jboss.node.name.

ВАЖНО! Когда используется балансироващик на базе undertow, на всех узлах, обслуживаемых балансировщиком, в jboss.node.name не должно содержаться символа "точка" (UNDERTOW-635).

Точки в значении этого параметра можно заменить на символ '_' в ***INSTALL_PATH/standalone/configuration/standalone.xml*** каждого узла.

Листинг 3.2.9.2_2

Пример фрагмента ***standalone.xml,*** поясняющего параметры настройки балансировщика в теге ***<socket-binding-group>***

```
<socket-binding-group        default-interface="public" name="standard-sockets" port-
offset="${jboss.socket.binding.port-offset:0}">
   <socket-binding                              interface="management" name="management-
http" port="${jboss.management.http.port:9990}"/>
   <socket-binding name="jgroups-tcp" port="7600"/>
   <socket-binding name="jgroups-tcp-fd" port="57600"/>
   <socket-binding name="http" port="${jboss.http.port:8080}"/>
   <socket-binding name="ajp" port="${jboss.ajp.port:8009}"/>
   <socket-binding name="https" port="${jboss.https.port:8443}"/>
   <socket-binding name="webui-mobile-http" port="8190"/>
   <socket-binding name="webui-mobile-ajp" port="8119"/>
   <socket-binding name="txn-recovery-environment" port="4712"/>
   <socket-binding name="txn-status-manager" port="4713"/>
   <outbound-socket-binding name="mail-smtp">
      <remote-destination
host="${argus.mail.smtp.host}" port="${argus.mail.smtp.port}"/>
   </outbound-socket-binding>
      <outbound-socket-binding name="remote-default-server1">
            <remote-destination host="192.168.100.180" port="8009">
  </outbound-socket-binding>
     <outbound-socket-binding name="remote-default-server2">
           <remote-destination host="192.168.100.181" port="8009">
  </outbound-socket-binding>
     <outbound-socket-binding name="remote-mobile-server1">
           <remote-destination host="192.168.100.180" port="8119">
     </outbound-socket-binding>
   <outbound-socket-binding name="remote-mobile-server2">
           <remote-destination host="192.168.100.181" port="8119">
   </outbound-socket-binding>
```

```
</socket-binding-group>
```

в ***outbound-socket-binding*** какие внешние порты должен использовать балансировщик для подключения к удаленным узлам

- для основного http порта указывается:

```
<outbound-socket-binding name="remote-default-server1">
<remote-destination host="192.168.100.180" port="8009">
</outbound-socket-binding>
<outbound-socket-binding name="remote-default-server2">
<remote-destination host="192.168.100.181" port="8009">
</outbound-socket-binding>
```

где в атрибуте host следует указывать IP-адрес удаленного узла

Дополнительную информацию по настройке ***undertow*** можно получить на сайте разработчика: https://docs.jboss.org/author/display/WFLY10/Using+Wildfly+as+a+Load+Balancer

3.2.10.3 Программный балансировщик HAProxy

3.2.10.3.1 Установка HAProxy

Проверить, что в ОС установлен репозиторий ***epel*** см. *п. 3.1.3 балансировщик СП*, после чего установить HAProxy:

```
yum install haproxy
```

3.2.10.3.2 Настройка HAProxy

Настройка HAProxy производится в файле конфигурации ***haproxy.cfg***, который находится в папке ***/etc/haproxy***

Листинг 3.2.9.3.2_1 Пример файла конфигурации ***haproxy.cfg***

```
#---------------------------------------------------------------------
# Global settings
#---------------------------------------------------------------------
global
# to have these messages end up in /var/log/haproxy.log you will
# need to:
#
# 1) configure syslog to accept network log events. This is done
# by adding the '-r' option to the SYSLOGD_OPTIONS in
# /etc/sysconfig/syslog
#
# 2) configure local2 events to go to the /var/log/haproxy.log
# file. A line like the following can be added to
# /etc/sysconfig/syslog
#
# local2.* /var/log/haproxy.log
#
log 127.0.0.1 local2 debug
chroot /var/lib/haproxy
pidfile /var/run/haproxy.pid
maxconn 4000
user haproxy
group haproxy
daemon
```

```
# turn on stats unix socket
stats socket /var/lib/haproxy/stats
#---------------------------------------------------------------------
# common defaults that all the 'listen' and 'backend' sections will
# use if not designated in their block
#---------------------------------------------------------------------
defaults
mode http
log global
option httplog
option dontlognull
retries 3
timeout http-request 10s
timeout queue 1m
timeout connect 10s
timeout client 1m
timeout server 1m
timeout http-keep-alive 10s
timeout check 10s
maxconn 3000
#---------------------------------------------------------------------
# frontend which proxys to the backends
#---------------------------------------------------------------------
frontend master-frontend
log 127.0.0.1 local6 debug
option httplog
bind *:8080
mode http
default_backend we-cluster
frontend config-frontend
bind *:8500
mode http
default_backend config-cluster
frontend segment1-frontend
log         127.0.0.1    local7 debug
option httplog
bind *:8081
mode http
default_backend segment1-cluster
#---------------------------------------------------------------------
# balancing between the various backends
#---------------------------------------------------------------------
backend we-cluster
balance leastconn
option httpchk GET /rest/servicecheck HTTP/1.0
server we1 192.168.101.163:8080 check
server we2 192.168.101.134:8080 check
server we3 192.168.101.46:8080 check
backend config-cluster
balance roundrobin
server conf1 192.168.101.163:8500 check
server conf2 192.168.101.134:8500 check
server conf3 192.168.101.46:8500 check
backend segment1-cluster
balance     leastconn
```

```
option httpchk GET /rest/servicecheck HTTP/1.0
server  te1 192.168.110.1:8500 check
server  te2 192.168.110.2:8500 check
server  te3 192.168.110.3:8500 check
listen stats
bind *:1936
stats enable
stats uri /
stats hide-version
stats auth qa:qa
                 listen siriusdb 0.0.0.0:5432
                 mode tcp
                  balance roundrobin
                  option pgsql-check user postgres
                  server master 192.168.101.171:5432 check
 server slave 192.168.101.173:5432 check backup
```

После внесения изменений в файл настроек – необходимо перезапустить балансировщик:

```
service haproxy restart
```

Для корректной работы HAProxy необходимо разрешить входящий tcp трафик, а именно в соответствии с файлом ***haproxy.cfg***, например

- по порту 8500 tcp

```
iptables -I INPUT 5 -p tcp -m state --state NEW -m tcp --dport 8500 -j ACCEPT
```

- по порту 8080 tcp

```
iptables -I INPUT 5 -p tcp -m state --state NEW -m tcp --dport 8080 -j ACCEPT
```

- по порту 1936 tcp

```
iptables -I INPUT 5 -p tcp -m state --state NEW -m tcp --dport 1936 -j ACCEPT
```

- по порту 5432 tcp

```
iptables -I INPUT 5 -p tcp -m state --state NEW -m tcp --dport 5432 -j ACCEPT
```

Для сохранения настроек:

```
iptables-save > /путь/к/файлу/файл
```

Для восстановления настроек:

```
cat /путь/к/файлу/файл | iptables-restore
```

3.2.11 Балансировщик БД

3.2.11.1 Keepalived

3.2.11.1.1 Установка Keepalived

Для установки Keepalived необходимо выполнить[89]:

[89] Требуются привилегии суперпользователя

```
yum install keepalived
```

3.2.11.1.2 Настройка Keepalived

1. Создать каталог конфигурации **/etc/keepalived/** для **Keepalived**:

```
mkdir /etc/keepalived/
```

2. Создать файл с именем **keepalived.conf** в каталоге конфигурации **/etc/keepalived/**:

```
nano /etc/keepalived/keepalived.conf
```

3. Отредактировать созданный файл конфигурации **keepalived.conf**[90]

```
cat /etc/keepalived/keepalived.conf
! Configuration File for keepalived

global_defs {
   router_id [уникальное названиe хоста keepalived, например PSQL-HA]
}

vrrp_script chk_haproxy {
     script "killall -0 haproxy"
     interval 2
     weight 2
}

vrrp_instance [название хоста, например VI_1] {
    state MASTER [начальное состояние при запуске, MASTER или BACKUP]
    interface eth0 [название интефейса, на котором будет работать VRRP]
    virtual_router_id 116 [уникальный идентификатор VRRP экземпляра, должен совпадать
на всех серверах]
    priority 114 [число от 0 до 255 - задает приоритет при выборе MASTER,
 сервер с большим приоритетом становится MASTER]
    advert_int 1
    authentication {
        auth_type PASS
        auth_pass AsDFDFD!@#123 [пароль можно задать новый]
    }
    virtual_ipaddress {
[ip-address-VIP] [блок виртуальных IP адресов, которые будут активны на сервере в
состоянии
MASTER. Должны совпадать на всех серверах внутри VRRP экземпляра.]
   }
    track_script {
        chk_haproxy
    }
}
```

4. Перезапустить сервис **keepalived**, чтобы изменения вступили в силу:

[90] Необходимо обратить внимание на необходимые корректировки, отмечены в [ ]!!! (Скобки необходимо удалить из файла конфигурации).

```
service keepalived restart
```

3.2.11.2 Haproxy

3.2.11.2.1 Установка Haproxy

Для установки Haproxy необходимо выполнить[91]:

1. Обновление списка пакетов:

```
yum update
```

2. Собственно, установку Haproxy:

```
yum install haproxy -y
```

3.2.11.2.2 Настройка Haproxy

1. Отредактировать файл конфигурации **/etc/haproxy/haproxy.cfg**[92]

```
global
    log /dev/log    local0
    log /dev/log    local1 notice
    chroot /var/lib/haproxy
    stats socket /run/haproxy/admin.sock mode 660 level admin
    stats timeout 30s
    user haproxy
    group haproxy
    daemon
    maxconn 2000

defaults
    log global
    mode tcp
    retries 2
    timeout client 30m
    timeout connect 4s
    timeout server 30m
    timeout check 5s

listen stats
    mode http
    bind *:7000
    stats enable
    stats uri /

# Connections to port 9999
listen PSQL_MASTER_9999
    bind *:5000
    option httpchk
    http-check expect status 200
    default-server inter 3s fall 3 rise 2 on-marked-down shutdown-sessions
    server [hostname]_5432 [ip-address-patroni1]:5432 maxconn 100 check port 8008
    server [hostname]_5432 [ip-address-patroni2]:5432 maxconn 100 check port 8008
```

[91] Требуются привилегии суперпользователя

[92] Необходимо обратить внимание на необходимые корректировки, отмечены в [ ]!!! (Скобки необходимо удалить из файла конфигурации).

```
```

2. Перезапустить сервис **haproxy**, чтобы изменения вступили в силу:

```
service haproxy restart
```

Если Haproxy не запускается, то необходимо проверить синтаксические ошибки:

```
/usr/sbin/haproxy -c -V -f /etc/haproxy/haproxy.cfg
```

3.2.11.3 Etcd

3.2.11.3.1 Установка Etcd

Для установки Etcd необходимо выполнить[93]:

1. Обновление списка пакетов:

```
yum update
```

2. Собственно, установку Etcd:

```
yum install etcd -y
```

Или же возможен альтернативный способ установки:

1. Создать временную директорию:

```
mkdir /tmp/etcd && cd /tmp/etcd
```

2. Установить пакет **wget**:

```
yum install wget -y
```

3. Скачать архив etcd:

```
curl -s https://api.github.com/repos/etcd-io/etcd/releases/latest \
  | grep browser_download_url \
  | grep linux-amd64 \
  | cut -d '"' -f 4 \
  | wget -qi -
```

4. Разавхивировать в директорию ***/usr/local/bin:***

```
tar xvf *.tar.gz
cd etcd-*/
sudo mv etcd* /usr/local/bin/
cd ~
rm -rf /tmp/etcd
```

5. Проверить версию **etcd** и **etcdctl**:

```
etcd –version
etcdctl --version
```

3.2.10.3.2 Настройка Etcd

1. Отредактировать файл конфигурации **/etc/etcd/etcd.conf**[94]

```
cat /etc/etcd/etcd.conf
```

[93] Требуются привилегии суперпользователя

[94] Необходимо обратить внимание на необходимые корректировки, отмечены в [ ]!!! (Скобки необходимо удалить из файла конфигурации).

```
#[Member]
#ETCD_CORS=""
ETCD_DATA_DIR="/var/lib/etcd/default.etcd"
#ETCD_WAL_DIR=""
ETCD_LISTEN_PEER_URLS="http://0.0.0.0:2380"
ETCD_LISTEN_CLIENT_URLS="http://0.0.0.0:2379"
#ETCD_MAX_SNAPSHOTS="5"
#ETCD_MAX_WALS="5"
ETCD_NAME="etcd2"
#ETCD_SNAPSHOT_COUNT="100000"
ETCD_HEARTBEAT_INTERVAL="100"
ETCD_ELECTION_TIMEOUT="1000"
#ETCD_QUOTA_BACKEND_BYTES="0"
#ETCD_MAX_REQUEST_BYTES="1572864"
#ETCD_GRPC_KEEPALIVE_MIN_TIME="5s"
#ETCD_GRPC_KEEPALIVE_INTERVAL="2h0m0s"
#ETCD_GRPC_KEEPALIVE_TIMEOUT="20s"
#
#[Clustering]
ETCD_INITIAL_ADVERTISE_PEER_URLS="http://[ip-address-patroni1]:2380"
ETCD_ADVERTISE_CLIENT_URLS="http://[ip-address-patroni2]:2379"
#ETCD_DISCOVERY=""
#ETCD_DISCOVERY_FALLBACK="proxy"
#ETCD_DISCOVERY_PROXY=""
#ETCD_DISCOVERY_SRV=""
ETCD_INITIAL_CLUSTER="etcd1=http://[ip-address-etcd1]:2380,etcd2=http://[ip-address-
etcd2]:2380,etcd3=http://[ip-address-etcd3]:2380"
ETCD_INITIAL_CLUSTER_TOKEN="etcd-cluster"
ETCD_INITIAL_CLUSTER_STATE="new"
#ETCD_STRICT_RECONFIG_CHECK="true"
#ETCD_ENABLE_V2="true"
#
#[Proxy]
#ETCD_PROXY="off"
#ETCD_PROXY_FAILURE_WAIT="5000"
#ETCD_PROXY_REFRESH_INTERVAL="30000"
#ETCD_PROXY_DIAL_TIMEOUT="1000"
#ETCD_PROXY_WRITE_TIMEOUT="5000"
#ETCD_PROXY_READ_TIMEOUT="0"
#
#[Security]
#ETCD_CERT_FILE=""
#ETCD_KEY_FILE=""
#ETCD_CLIENT_CERT_AUTH="false"
#ETCD_TRUSTED_CA_FILE=""
#ETCD_AUTO_TLS="false"
#ETCD_PEER_CERT_FILE=""
#ETCD_PEER_KEY_FILE=""
#ETCD_PEER_CLIENT_CERT_AUTH="false"
#ETCD_PEER_TRUSTED_CA_FILE=""
#ETCD_PEER_AUTO_TLS="false"
#
#[Logging]
#ETCD_DEBUG="false"
#ETCD_LOG_PACKAGE_LEVELS=""
#ETCD_LOG_OUTPUT="default"
#
#[Unsafe]
#ETCD_FORCE_NEW_CLUSTER="false"
#
#[Version]
#ETCD_VERSION="false"
#ETCD_AUTO_COMPACTION_RETENTION="0"
```

```
#
#[Profiling]
#ETCD_ENABLE_PPROF="false"
#ETCD_METRICS="basic"
#
#[Auth]
#ETCD_AUTH_TOKEN="simple"
```

2. Перезапустить сервис **etcd**, чтобы изменения вступили в силу:

```
service etcd restart
```

Проверить ноды и лидера кластера можно так:

```
etcdctl endpoint status --endpoints=$(etcdctl member list | grep -o '[^
]\+:2379' | paste -s -d,) -w table
```

Пример сервиса: /etc/systemd/system/etcd.service или /usr/lib/systemd/system/etcd.service

Аргумент: "--enable-v2=true" необходим при запуске etcd v3+ для использовании APIv2 по которому Patroni обращается к etcd.

```
[Unit]
Description=etcd key-value store
Documentation=https://github.com/etcd-io/etcd
After=network-online.target local-fs.target remote-fs.target time-sync.target
Wants=network-online.target local-fs.target remote-fs.target time-sync.target

[Service]
User=etcd
Type=notify
EnvironmentFile=/etc/etcd/etcd.conf
ExecStart=/usr/local/bin/etcd --enable-v2=true
Restart=always
RestartSec=10s
LimitNOFILE=40000

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

3.2.11.4 Patroni

3.2.11.4.1 Установка Patroni

Для установки Patroni необходимо выполнить[95]:

1. Остановить службу **Postgres**, чтобы **Patroni** мог управлять ею с этого момента::

```
systemctl stop postgresql
```

2. Patroni использует утилиты, которые устанавливаются вместе с **Postgres** и расположены в **/usr/pgsql-10/bin** (или /argus/pgsql-10/data если выполнен перенос директории). Необходимо создать символические ссылки в PATH чтобы Patroni мог найти утилиты:

```
ln -s /usr/pgsql-10/bin/* /usr/sbin/
```

3. Установить **python3** и **pip3**:

```
yum install python3 python-pip3 -y
```

[95] Требуются привилегии суперпользователя

4. Убедиться что установлена последняя версия **setuptools** пакета **python**:

```
pip3 install --upgrade setuptools
```

5. Использовать pip3 для установки **Patroni**:

```
pip3 install patroni
```

3.2.10.4.2 Настройка Patroni

1. Patroni может быть настроен с использованием файла YAML. Необходимо его создать и разместить в **/etc/patroni_01.yaml**. Далее необходимо внести корректировки в этот файл[96]

```
scope: postgres
namespace: /db/
name: postgresql0

restapi:
    listen: [ip-address-patroni]:8008
    connect_address: [ip-address-patroni]:8008
etcd:
    hosts: [ip-address-etcd]:2379, [ip-address-etcd]:2379, [ip-address-etcd]:2379
bootstrap:
    dcs:
        ttl: 30
        loop_wait: 10
        retry_timeout: 10
        maximum_lag_on_failover: 1048576
        postgresql:
            use_pg_rewind: true
            parameters:
              archive_mode: "on"
              archive_command: cp %p /var/lib/pgsql-10/archive/%f
(или /argus/pgsql-10/archive/%f если выполнен перенос директории)
              max_connections: 1000
              shared_buffers: 4GB
              effective_cache_size: 10GB
              maintenance_work_mem: 2GB
              checkpoint_completion_target: 0.9
              wal_buffers: 16MB
              default_statistics_target: 500
              random_page_cost: 1.1
              effective_io_concurrency: 200
              work_mem: 393kB
              min_wal_size: 1GB
              max_wal_size: 1GB
              max_worker_processes: 8
              dynamic_shared_memory_type: posix
              log_destination: 'csvlog'
              logging_collector: on
              log_directory: 'log'
              log_filename: 'postgresql-%Y-%m-%d_%H%M%S.log'
              log_truncate_on_rotation: on
              log_rotation_age: 1d
              log_rotation_size: 100MB
              log_timezone: 'Europe/Moscow'
```

[96] Необходимо обратить внимание на необходимые корректировки, отмечены в [ ]!!! (Скобки необходимо удалить из файла конфигурации).

```
                datestyle: 'iso, mdy'
                timezone: 'Europe/Moscow'
                lc_messages: 'en_US.UTF-8'
                lc_monetary: 'en_US.UTF-8'
                lc_numeric: 'en_US.UTF-8'
                lc_time: 'en_US.UTF-8'
                default_text_search_config: 'pg_catalog.english'

    initdb:
    - encoding: UTF8
    - data-checksums
    - auth-host: md5
    - auth-local: trust

    pg_hba:
    - host replication replica 127.0.0.1/32 md5
    - host replication replica [ip-address]/0 trust (адрес БД где развернута БД
postgresql)
    - host replication replica [ip-address]/0 trust (адрес БД где развернута
резервная БД postgresql)
    - host replication postgres [ip-address]/0 trust (адрес БД где развернута БД
postgresql)
    - host replication postgres [ip-address]/0 trust (адрес БД где развернута
резервная БД postgresql)
    - host replication replica all md5
    - host all postgres 127.0.0.1/32 md5

    users:
        admin:
            password: admin
            options:
                - createrole
                - createdb
postgresql:
    listen: [ip-address-postgresql]:5432
    connect_address: [ip-address-postgresql]:5432
    data_dir: /var/lib/pgsql-10/data (или /argus/pgsql-10/data если выполнен перенос
директории)
    pgpass: /tmp/pgpass
    authentication:
        replication:
            username: postgres
            password: postgres
        superuser:
            username: postgres
            password: postgres
    parameters:
        unix_socket_directories: '.'
tags:
    nofailover: false
    noloadbalance: false
    clonefrom: false
    nosync: false
```

2. УЗ **postgres** должна иметь доступ к указанной директории в **data_dir.** Если этого каталога не существует, необходимо создать его:

```
mkdir /var/lib/pgsql-10/data -p (или /argus/pgsql-10/data если выполнен перенос
директории)
```

3. УЗ **postgres** необходимо сделать владельцем данной директории:

```
chown postgres:postgres /var/lib/pgsql-10/data(или /argus/pgsql-10/data если выполнен перенос директории)
```

4. Установить ограничения для каталога, чтобы он был доступен только для УЗ **postgres:**:

```
chmod 700 /var/lib/pgsql-10/data(или /argus/pgsql-10/data если выполнен перенос директории)
```

5. Необходимо создать скрипт systemd, который позволит запускать, останавливать и выполнять проверку статуса **Patroni**. Для этого создать файл ***/etc/systemd/system/patroni.service*** со следующим содержимым:

```
[Unit]Description=Runners to orchestrate a high-availability
PostgreSQLAfter=syslog.target
network.target[Service]Type=simpleUser=postgresGroup=postgresExecStart=/usr/bin/patr
oni
/etc/patroni_01.yamlKillMode=processTimeoutSec=30Restart=no[Install]WantedBy=multi-
user.targ
```

6. Запустить **Patroni** и **Postgres**:

```
systemctl start patroni
```

7. Проверить статус **Patroni:**

```
systemctl status patroni
```

Если все настроено правильно, выходные данные первого узла (мастера) будут выглядеть так:

```
● patroni.service - Runners to orchestrate a high-availability PostgreSQLLoaded:
loaded (/etc/systemd/system/patroni.service; enabled; vendor preset: enabled)Active:
active (running) since Thu 2017-07-29 16:49:18 UTC; 8min agoMain PID: 13097
(patroni)..... INFO: Lock owner: postgresql0; I am postgresql0... INFO: no action.
i am the leader with the lock
```

При запуске последующих узлов лог будет выглядеть следующим образом:

```
INFO: no action. i am a secondary and i am following a leaderLock owner:
postgresql0; I am postgresql2
```

3.2.11 Средства мониторинга

3.2.11.1 Настройка мониторинга СУБД

Мониторинг СУБД осуществляется средствами системы мониторинга Zabbix. (см. п. 3.2.11.3 Установка и настройка системы мониторинга Zabbix)

Для мониторинга в Zabbix Server настраивается соответствующий шаблон.

Настройку шаблона осуществляют сотрудники Исполнителя.

3.2.11.2 Установка и настройка мониторинга СП

На хостах, с которых планируется удалённый мониторинг ресурсов СП, устанавливаются утилиты: JVisualVM , JConsole, CLI (Command Line Interface)

На хостах СП, необходимо установить и настроить следующие утилиты и сервисы:

- Утилита NMON.
- Zabbix Agent.

Установка изложена в пунктах:

- *3.2.11.3.1 Установка компонентов Zabbix на хост с доступом в интернет*
- *3.2.11.3.2 Установка компонентов Zabbix на хост без доступа в интернет.*

Настройка приведена в *п. 3.2.11.3.3 Настройка компонентов Zabbix*

3.2.11.3 Установка и настройка системы мониторинга Zabbix

3.2.11.3.1 Установка компонентов Zabbix на хост с доступом в интернет

Перед началом установки компонентов Zabbix на хост с доступом в интернет необходимо выполнить настройку программной среды. Настройка программной среды описана в *п. 3.1.11.3 Настройка программной среды системы мониторинга Zabbix*.

Для RHEL 6x, Oracle Linux 6x, CentOS 6x и других поддерживаемых ОС:

- Инструкция по установке Zabbix Agent из пакета приведена в Zabbix Documentation 3.0 раздел ***3. Установка из пакетов.*** *п. 4 Установка агента*: https://www.zabbix.com/documentation/3.0/ru/manual/installation/install_from_packages/agent_installation
- Установка Zabbix Java Gateway из пакета командой:
- Инструкция по установке Zabbix Proxy с поддержкой SQLite3 из пакета приведена в Zabbix Documentation 3.0 раздел ***3. Установка из пакетов.****п. 5 Установка прокси*: https://www.zabbix.com/documentation/3.0/ru/manual/installation/install_from_packages/proxy_installation

Установка Zabbix Java Gateway из пакета командой:

```
yum install zabbix-java-gateway
```

Установка Zabbix Proxy из пакета командой:

```
yum install zabbix-proxy-sqlite3
```

Установка Zabbix Agent из пакета командой:

```
yum install zabbix-agent
```

При наличии доступа в интернет, скачивается и устанавливаются пакет последний версии.

3.2.11.3.2 Установка компонентов Zabbix на хост без доступа в интернет

При отсутствии доступа в интернет для RHEL 6x, Oracle Linux 6x, CentOS 6x скачать пакеты из репозитория: http://repo.zabbix.com/zabbix/3.0/rhel/6/x86_64/

Устанавливаются пакеты через утилиту **yum** с указанием полной версии и имени пакта.

- Установка Zabbix Agent:

```
yum install zabbix-agent-3.0.4-1.el6.x86_64.rpm
```

- Установка Zabbix Proxy:

```
yum install zabbix-proxy-sqlite3-3.0.4-1.el6.x86_64.rpm
```

- Установка Zabbix Java Gateway

```
yum install zabbix-java-gateway-3.0.4-1.el6.x86_64.rpm
```

3.2.11.3.3 Настройка компонентов Zabbix

Настройка Zabbix Proxy и Zabbix Java Gateway и Zabbix Agent осуществляется согласно топологии, приведенной в *п. 2.1.12 Система мониторинга* см. *рис 2.1.12 Схема мониторинга*.

- Настройка Zabbix Agent

Внести изменения в конфигурационный файл /etc/zabbix/zabbix_agentd.conf в соответствии с приведенным примером в Листинге 3.2.11.3.3_1, задав значения настроек специфичных для хоста:

Server - IP-адресс Zabbix Proxy.

Hostname - сетевое имя хоста (проверяется командой uname -n).

Листинг 3.2.11.3.3_1

Пример конфигурационного файла /etc/zabbix/zabbix_agentd.conf:

```
PidFile=/var/run/zabbix/zabbix_agentd.pid
LogFile=/var/log/zabbix/zabbix_agentd.log
LogFileSize=0
Server=192.168.100.100
Hostname=argus.domain.ru
RefreshActiveChecks=60
Include=/etc/zabbix/zabbix_agentd.d/
```

- Настройка Zabbix Proxy

Внести изменения в конфигурационный файл /etc/zabbix/zabbix_proxy.conf в соответствии с приведенным примером в Листинге 3.2.11.3.3_2, задав значения настроек специфичных для хоста:

Server - IP-адресс Zabbix Server НТЦ Аргус.

Hostname - сетевое имя хоста (проверяется командой uname -n).

Листинг 3.2.11.3.3_2

Пример конфигурационного файла /etc/zabbix/zabbix_proxy.conf:

```
# 0 - активный прокси
ProxyMode=0
Server=192.168.100.100
Hostname=argus.domain.ru
LogFile=/var/log/zabbix/zabbix_proxy.log
LogFileSize=0
PidFile=/var/run/zabbix/zabbix_proxy.pid
DBName=/home/argus/db/zabbix_proxy.sqlite3
DBUser=zabbix
StartPingers=2
######### PROXY SPECIFIC PARAMETERS #############
# Used for monitoring availability of Proxy on server side.
HeartbeatFrequency=60
# How often proxy retrieves configuration data from Zabbix Server in seconds.
ConfigFrequency=600
# Proxy will send collected data to the Server every N seconds.(onle for proxy
active mode)
DataSenderFrequency=1
### Option: JavaGateway
JavaGateway=localhost
JavaGatewayPort=10052
StartJavaPollers=10
SNMPTrapperFile=/var/log/snmptrap/snmptrap.log
Timeout=4
```

```
ExternalScripts=/usr/lib/zabbix/externalscripts
LogSlowQueries=3000
```

- Настройка Zabbix Java Gateway

1. После установки Zabbix Java Gateway необходимо заменить **/usr/sbin/zabbix_java/bin/zabbix-java-gateway-3.0.x.jar** на jar-файл предостовляемый НТЦ Аргус. Заменяемый jar-файл содержится в себе изменения необходимые для мониторинга СП ПИРС.
2. Внести изменения в конфигурационный файл /etc/zabbix/zabbix_java_gateway.conf в соответствии с приведенным примером в Листинге 3.2.11.3.3_3

Листинг 3.2.11.3.3_3

Пример конфигурационного файла /etc/zabbix/zabbix_java_gateway.conf:

```
# Default: LISTEN_IP="0.0.0.0"    - IP address to listen on.
# Default: LISTEN_PORT=10052    - Port to listen on.
# Default: PID_FILE=        - If omitted, Zabbix Java Gateway is started as a
console application.
PID_FILE="/var/run/zabbix/zabbix_java_gateway.pid"
# Default: START_POLLERS=5    - Number of worker threads to start.
# Default: TIMEOUT=3    - How long to wait for network operations.
TIMEOUT=3
```

3. Скопировать библиотеки с СП WildFly 10 в /usr/sbin/zabbix_java/lib/

```
[developer@zabbixagent jboss3100]$ mkdir nmdir
[developer@zabbixagent
jboss3100]$ for i in $(cat needed_modules); do find ./modules/system/layers/base/ -
iname ${i}*.jar -exec cp {} ./nmdir/ \; ; done
[developer@zabbixagent jboss3100]$ ls -l nmdir
-rw-rw-r-- 1 developer developer  57193 мар 29 19:25 jboss-logging-3.1.4.GA.jar
-rw-rw-r-- 1 developer developer 314271 мар 29 19:25 jboss-logmanager-
1.5.2.Final.jar
-rw-rw-r-- 1 developer developer 218800 мар 29 19:25 jboss-marshalling-
1.4.9.Final.jar
-rw-rw-r-- 1 developer developer  82509 мар 29 19:25 jboss-marshalling-river-
1.4.9.Final.jar
-rw-rw-r-- 1 developer developer 270831 мар 29 19:25 jboss-remoting-4.0.6.Final.jar
-rw-rw-r-- 1 developer developer  89802 мар 29 19:25 jboss-sasl-1.0.4.Final.jar
-rw-rw-r-- 1 developer developer  16483 мар 29 19:25 jcl-over-slf4j-1.7.2.jbossorg-
1.jar
-rw-rw-r-- 1 developer developer   3979 мар 29 19:25 jul-to-slf4j-stub-
1.0.1.Final.jar
-rw-rw-r-- 1 developer developer 482869 мар 29 19:25 log4j-jboss-logmanager-
1.1.0.Final.jar
-rw-rw-r-- 1 developer developer 312164 мар 29 19:25 remoting-jmx-2.0.0.Final.jar
-rw-rw-r-- 1 developer developer  26281 мар 29 19:25 slf4j-api-1.7.2.jbossorg-1.jar
-rw-rw-r-- 1 developer developer 507134 мар 29 19:25 xnio-api-3.3.0.Final.jar
-rw-rw-r-- 1 developer developer  99145 мар 29 19:25 xnio-nio-3.3.0.Final.jar
```

где needed_modules - файл со списком необходимых библиотек, составляется вручную и содержит имена либ без версии:

```
jboss-logging
jboss-logmanager
jboss-marshalling
jboss-marshalling-river
jboss-remoting
jboss-sasl
jcl-over-slf4j
```

```
jul-to-slf4j-stub
log4j-jboss-logmanager
remoting-jmx
slf4j-api
xnio-api
xnio-nio
```

4. Библиотеки поместить на хост с установленным Zabbix Proxy в каталог /usr/sbin/zabbix_java/lib/

3.2.11.3.4 Запуск и остановка компонентов Zabbix

- Zabbix Agent

запускается командой:

```
/etc/init.d/zabbix-agent start
```

Останавливается командой:

```
/etc/init.d/zabbix-agent stop
```

- Zabbix Proxy

Запускается командой:

```
/etc/init.d/zabbix-proxy start
```

Останавливается командой:

```
/etc/init.d/zabbix-proxy stop
```

- Zabbix Java Gateway

запускается командой:

```
/etc/init.d/zabbix-java-gateway start
```

Останавливается командой

```
/etc/init.d/zabbix-java-gateway stop
```

3.3 Установка и настройка клиентского ПО решения WFM CC

3.3.1 Общие требования к настройке рабочих мест

Рабочие места персонала необходимо обеспечить персональным компьютером, подсоединенным к ЛВС.

С рабочих мест должна быть обеспечена IP-связанность с сервером БД и с каждым из сервисов решения WFM CC,

или же с балансировщиком в случае дублирования сервисов и работы их в кластерном режиме.

3.3.2 Требования к ПО Web-client

Таблица 3 - Требования к программной части

Название	Минимальные требования	Рекомендуемые требования

ОС	Операционная система, официально поддерживающая установку ниже описанных браузеров.	Операционная система, официально поддерживающая установку ниже описанных браузеров.
Браузер	Firefox 91+, Microsoft Edge 103+, Chrome 98+	Chrome, Firefox или Microsoft Edge последней версии. IE не рекомендуется

3.4 Необходимые регулярные процедуры

3.4.1 Резервное копирование

Частота резервного копирования и длительность хранения резервных копий производится согласно внутреннему регламенту заказчика.

Резервное копирование необходимо делать перед любым проведением технологических работ, связанных с обновлением/изменением серверного ПО.

3.4.1.1 Резервное копирование БД

Возможно создание следующих видов резервных копий:

- Логическая резервная копия
  Выполняется утилитами *pg_dump, pg_dumpall*
  Срок хранения логических резервных копий определяется внутренним регламентом заказчика
  Рекомендуется хранить не менее трех последних логических резервных копий.
- Физическая резервная копия
  Выполняется утилитой *pg_basebackup*
  Срок хранения физических резервных копий определяется внутренним регламентом заказчика
  Рекомендуется хранить не менее трех последних логических резервных копий.
- Резервное копирование ВМ
  В случае использования для работы БД системы виртуализации - следует делать делать копии ВМ[97]
  Срок хранения резервных копий ВМ определяется внутренним регламентом заказчика
  Рекомендуется хранить не менее трех последних логических резервных копий.

Все типы резервных копий следует регулярно проверять на консистентность и возможность восстановления

Помимо резервных копий следует сохранять файлы конфигурации экземпляров БД[98]

3.4.1.2 Резервное копирование СП

Резервное копирование СП зависит от выбранного способа развертывания ПО:

[97] Резервные копии ВМ необходимо делать обязательно в сочетании с логичискими и физическими бэкапами !

[98] Следует копировать после каждого внесения изменений

- в случае отдельного аппаратного сервера – это копирование каталога, в который инсталлирован СП, что составит примерно 1 GB[99]

- в случае виртуализированной системы[100]– это копия всей виртуальной машины

Резервное копирование не требует остановки сервиса.

Резервное копирование может осуществляться как средствами ОС, так и внешними системами.

3.4.1.3 Резервное копирование балансировщика нагрузки СП

- В случае виртуализированной системы – это копия всей виртуальной машины

Помимо резервных копий следует сохранять файлы конфигурации экземпляров балансировщика нагрузки[101]

3.4.1.4 Резервное копирование балансировщика нагрузки БД

- В случае виртуализированной системы – это копия всей виртуальной машины

Помимо резервных копий следует сохранять файлы конфигурации экземпляров балансировщика нагрузки[102]

3.4.2 Мониторинг показателей

Везде, где указана необходимость мониторинга, необходимо собирать исторические данные для диагностики и анализа аварийных ситуаций.

Мониторинг может осуществляться средствами ОС, СУБД (sql-запросы к системным представлениям), СП (Admin Console) и внешних утилит (JVisualVM, JConcole, Command Line Interface, NMON) и систем мониторинга (Zabbix).

Необходимо постоянно следить за показаниями мониторинга и анализировать их в соответствии с инструкциями.

При возникновении аварийной ситуации необходимо выполнять действия по сбору артефактов, анализу проблемы и её решению в соответствии с инструкциями приведенными в *п. 2.4.5 Типовые действия при аварии*

3.4.2.1 Мониторинг показателей БД

Мониторинг показателей БД осуществляется по предварительно сформированному шаблону на Zabbix-Server.

Параметры мониторинга и пороговые значения, при превышении которых происходит оповещение, выставляются индивидуально в процессе сопровождения и накопления информации о стабильной работе БД[103]

---

[99] Только бинарные и конфигурационные файлы без учета логов

[100] Работа сервера приложений возможна в среде виртуализации, протестирована и рекомендована к промышленной эксплуатации с использованием систем виртуализации: Linux Xen (Debian, xen 3.2.1), VMware vSphere 5.1 и выше.

[101] Следует копировать после каждого внесения изменений

[102] Следует копировать после каждого внесения изменений

[103] Список параметров приведен в Приложении *Параметры мониторинга сервера БД*

3.4.2.2 Мониторинг показателей СП

Мониторинг показателей СП осуществляется по предварительно сформированному шаблону на Zabbix-Server.

Параметры мониторинга и пороговые значения, при превышении которых происходит оповещение, выставляются индивидуально в процессе сопровождения и накопления информации о стабильной работе СП[104]

3.4.3 Архивация журналов операций

При работе компонентов решения WFM CC – в лог-файлах формируется большое количество отладочной информации.

Лог файлы необходимы для проведения контраварийных работ, поэтому должны храниться на хосте какое-то время (как правило это 3-5 дней).

Необходимо следить за местом на дисковом пространстве, чтобы формируемые лог-файлы не заполняли его полностью (что грозит остановкой сервиса), а также своевременно удалять лог-файлы по истечению оговоренного срока хранения.

Для экономии места на диске лог-файлы рекомендуется архивировать и удалять.

Лог-файлы формируются в следующих каталогах:

- БД ***<Каталог_установки_СП>/data/log***

Листинг 3.4.3.1

Пример скрипта архивации и удаления для БД:

```
#!/bin/bashlog_path=/argus/pgdata/logexcept_file_name=postgresql-$(date +"%Y-%m-
%d").logfind $log_path -mtime +2 -deletearray=(`find $log_path -name "*.log" -type
f`)for ((i=0; i < ${#array[@]}; i++))doif [ ${array[$i]} ==
$log_path/$except_file_name ]thencontinuefigzip ${array[$i]}done
```

- СП ***<Каталог_установки_СП>/standalone/log/***

Лог-файлы структурированы следующим образом

- Каталог **bugreports/** содержит архивы с отчетами об ошибках.

Файлы с именами вида "[пользователь]-[сессия]-[дата]-[время]-[корневое исключение]-[порядковый номер].zip" содержат контекстную информацию о событии, вызвавшем создание отчета об ошибке.

- ***gcstats.log*** - лог сборки мусора, сохраняемый опцией ***JVM Xloggc***. Лог-файл создается, когда включен промышленный режим эксплуатации СП.

- ***access_log.log**** - лог доступа, содержит сведения о поступивших HTTP-запросах.

- ***last_boot_errors.log*** - лог ошибок заполняется при запуске СП.

- ***response.log*** - лог-файл всегда пустой, вынужденно создается при старте сервера по техническим причинам.

- ***server.log**** - общий лог, содержит сообщения, важные в масштабе всего сервера: подробное описание запуска сервера, создание и завершение пользовательских сессий, произошедшие

[104] Список параметров приведен в *Приложении Параметры мониторинга СП*

ошибки, вызовы call-центра, превышение времени обработки HTTP-запросов, прочие сообщения с достаточно высоким уровнем логирования.

- ***webservices.log*** - лог-файл для протоколирования обращений по веб-сервисам к СП.

Логи ***access_log*** , ***server*** ротируются по суткам, то есть в файл с именем вида "****.log***" заносятся сообщения текущего дня, а в полночь к имени файла добавляется точка и дата в формате "***yyyy-mm-dd***". Последующие сообщения заносятся в новый файл с именем вида "****.log***".

В состав СП входит скрипт автоматизации удаления/архивирования устаревших лог-файлов. Скрипт доступен в подкаталоге ***<Каталог_установки_СП>/tools/unix*** и называется ***remove_old_logs.sh***.

Инструкция по применению скрипта присутствует в примечаниях к скрипту.

Листинг 3.4.3.2

Пример скрипта архивации и удаления для СП[105]:

```
#!/bin/bash#Скрипт архивации логов. Хранит архивы N днейexport
PATH=/usr/local/bin:/bin:/usr/bin:/usr/local/sbin:/usr/sbin:/sbin# путь до каталога
с логами Сервера Приложенийexport argus_logs=/argus/jboss_prod/standalone/log# путь
файлу, в который ведется лог работы скриптаexport
logfile=/argus/scripts/arch_logs.log# путь до каталога с бэкапом логовexport
bugreports=/argus/jboss_prod/standalone/log/bugreportsdate_now=$(date +%Y\-%m\-
%d);echo '=====================================================' >> $logfileecho ""
>> $logfileecho "Started at `date`" >> $logfileecho '' >> $logfilels -al $argus_logs >>
$logfileecho '' >> $logfile#ПеременныеOLD_LOGS=$(date -d "-1 day" +"%Y-%m-%d" )echo
$OLD_LOGSBUGREPORTS=$(date -d "-1 day" +"%Y.%m.%d")echo $BUGREPORTSOLD_ARCH=$(date -d
"-30 day" +"%Y.%m.%d")echo $OLD_ARCH#Создадим архивы старше 1 днyfind
/argus/jboss_prod/standalone/log -name "*$OLD_LOGS*" | xargs tar -cvzf
/argus/jboss_arch/$OLD_LOGS.logs.tar.gzfind
/argus/jboss_prod/standalone/log/bugreports -name "*$BUGREPORTS*" | xargs tar -cvzf
/argus/jboss_arch/$BUGREPORTS.logs.bugreports.tar.gz#Удалим все файлы старше 1 дня
из каталога логов СПfind /argus/jboss_prod/standalone/log -name "*$OLD_LOGS*" -exec rm
-rf {} \;find /argus/jboss_prod/standalone/log/bugreports -name "*$BUGREPORTS*" -exec
rm -rf {} \;#Удаление архивов, которые лежат дольше, чем кол-во дней хранения
логовfind /argus/jboss_arch -name "*$OLD_ARCH*" -exec rm -rf {} \;echo "Finished at
`date`" >> $logfileecho '=====================================================' >>
$logfile
```

Скрипты архивации и удаления можно регулярно запускать, прописав их в планировщик задач, например в ***cron*** (OS Linux).

Архивирование и удаление логов балансировщиков

Пример скрипта архивации и удаления для балансировщиков:

```
#!/bin/bashlog_path=/var/log/httpdfind $log_path -mtime +7 -deletearray=(`find
$log_path -name "error_log-*" -o -name "access_log-*"`)for ((i=0; i < ${#array[@]};
i++))dogzip ${array[$i]}done
```

3.4.4 Настройка NMON

NMON (сокращение от Nigel's Monitor) — инструмент администратора, предназначенный для анализа и мониторинга производительности Linux-систем.

[105] Необходимо адаптировать скрипт к конкретному хосту и ОС; Скрипт входит в дистрибутив СП. Находится относительно каталога установки СП: tools/unix/remove_old_logs.sh

NMON можно скачать: http://nmon.sourceforge.net/pmwiki.php?n=Site.Download

Скрипт регулярной подготовки отчетов

Запускается раз в сутки (прописывается в планировщик задач cron)

Пример скрипта[106]

```
#!/bin/bashexport NMONFS=/argus/nmoncd $NMONFS/argus/scripts/nmon_x86_64_rhel6 -f -m
$NMONFS -s 60 -c 1440/bin/find $NMONFS -name '*.nmon' -mmin +2880 | xargs
gzip/bin/find $NMONFS -name '*.nmon.gz' -mtime +365 | xargs rm
```

В скрипте NMON делает 1440(-c 1440) snapshots через 60(-s 60) секунд каждый. Сохраняет отчеты в каталог NMONFS=/argus/nmon. Через двое суток(-mmin +2880) архивирует отчеты. Через 365(-mtime +365) дней удаляет архивы отчетов.

Анализатор отчетов. Отчет.

Nmon analyser – инструмент для составления отчетов о производительности нескольких подсистем, таких как использование процессора (процессоров), использование оперативной памяти и файлов подкачки, статистику дисковых и сетевых операций (ввода/вывода), информацию о состоянии ядра и многое другое... Описание всех подсистемы, статистику по которым предоставляет Nmon analyser, можно найти в документации, которая поставляется в архиве с анализатором. Отчет представляет из себя Excel документ, на каждом листе которого отображается статистика по одной из подсистем.

Для того чтобы сформировать анализ отчета Nmon, необходимо загрузить Nmon analizer, нажать на кнопку "Analize nmon data", выбрать отчет Nmon.

Анализатор отчетов можно скачать:

https://www.ibm.com/developerworks/community/wikis/home?lang=en#!/wiki/Power+Systems/page/nmon_analyser

Скрипт может быть установлен удаленно силами исполнителя при помощи системы управления конфигурацией: Ansible. Для этого на хосте должен быть поднят ssh-сервер и установлен пакет python2.7 и выше.

3.4.5 Очистка каталога временных файлов СП

В каталоге временных файлов хранятся временные файлы, используемые при построении и выгрузке отчетов.

Указывается каталог в виде значения параметра СП **java.io.tmpdir**

Очищать его можно скриптом, прописанным в планировщике задач, например в ***cron*** (OS Linux).

Листинг 3.4.5

Пример скрипта удаления временных файлов для СП Сервер отчетов[107]:

```
#!/bin/bash#Скрипт удаления временных файлов java.io.tmpdir. Хранит файлы N
днейexport PATH=/usr/local/bin:/bin:/usr/bin:/usr/local/sbin:/usr/sbin:/sbinexport
argus_tmp=/argus/tmpexport logfile=/argus/scripts/cleartmp.logdate_now=$(date +%Y\-
%m\-%d);echo '====================================================' >> $logfileecho
"" >> $logfileecho "Started at `date`" >> $logfileecho '' >> $logfilels -al $argus_tmp
```

[106] Скрипт входит в дистрибутив СП. Находится относительно каталога установки СП: tools/unix/ nmon.sh

[107] Необходимо адаптировать скрипт к конкретной структуре каталогов хоста (см. п. 3.1.2.9 Организация каталогов) и ОС; Скрипт не входит в дистрибутив СП.

```
>> $logfileecho '' >> $logfile#Удаление временных файлов, которые лежат дольше, чем N
днейfind /argus/tmp -mtime +N -exec rm -rf {} \;echo "Finished at `date`" >>
$logfileecho '============================================================' >> $logfile
```

3.5 Мероприятия по обновлению ПО в случае перехода в другой часовой пояс

3.5.1 .Проверка наличия информации о новом часовом поясе

Перед тем как появятся патчи для БД и JVM, информация должна появиться на сайте https://www.iana.org/time-zones[108]

3.5.2 Обновление часовых поясов JDK

Для обновления данных часового пояса нужно перейти в каталог ***JDK_HOME/bin*** и выполнить команду:

```
java -jar tzupdater.jar -u
```

tzupdater.jar можно взять здесь:

https://www.oracle.com/technetwork/java/javase/downloads/tzupdater-download-513681.html

Проверить результататы обновления, для чего из каталога ***JDK_HOME/bin*** выполнить команду:

```
java TestTimeddmmyyyy[109]
```

3.5.3 Обновление библиотеки joda-time в составе СП

Получить из Аргус СП с обновленной библиотекой **joda-time**

[108] Изменения перечислены в файле News в архива (например tzdb-2018e.tar.lz)

[109] **TestTimeddmmyyyy** - файл проверки обновления таймзоны, подготовленный в Аргус

4. Справочники администратора

4.1 Справочник администратора БД

4.1.1 pgdump

Для того, чтобы снять и упаковать копию с продуктивной зоны необходимо выполнить команду на сервере БД (под УЗ postgres):

`cd /tmp && pg_dump -Fc prod > prod_07.06.2022.Fc && tar -czvf prod_07.06.2022.Fc.tar.gz prod_07.06.2022.Fc`	`prod - это название БД, которую мы извлекаем.`

Далее необходимо выгрузить копию, например через winscp, на целевой сервер БД, где ее необходимо распаковать:

```
cd /tmp && tar -xf prod_07.06.2022.Fc.tar.gz
```

После чего подключиться к целевой БД:

```
psql -h 127.0.0.1 -p 5432 -U postgres
```

Сохранить имеющуюся БД

```
alter database demodb rename to demodb_old;
```

Создать новую БД и загрузить в нее дамп

```
CREATE DATABASE demodb OWNER argus_sys;
psql demodb < /tmp/prod_07.06.2022.Fc
```

ВАЖНО ! каталог /tmp приведен для примера, поскольку есть на любой ОС типа Linux и в него есть возможность записи с любой УЗ

на самом деле размера /tmp может не хватить и в зависимости от размера дампа следует выбрать другой каталог в разделе ОС, в котором достаточно места для его хранения дампа и в котором есть возможность записи из-под УЗ postgres

более подробно с утилитой **pgdump** можно ознакомиться здесь: https://postgrespro.ru/docs/postgresql/10/app-pgdump

4.2 Справочник администратора СП и сервисов

4.2.1 heapdump и threaddump

Снятие дампов (heapdump и threaddump) выполняется при помощи утилит JDK: jcmd и jstack и имеет следующий формат выполнения:

`heapdump`	`jcmd <pid> GC.heap_dump <file-path>`
`threaddump`	`jstack <pid> > <file-path>`

обычно в в докер-контейнере выполняется единственный процесс с pid = 1

4.2.1.1 heapdump и threaddump СП WFM CC

Для того, чтобы снять heapdump СП WFM CC необходимо

- перейти в каталог установки СП WFM CC ***INSTALL_PATH/bin*** и выполнить команду:

```
./runjboss.sh heap-dump
```

дамп будет сформирован в текущем каталоге ***INSTALL_PATH/bin***

для того, чтобы снять threaddump СП WFM CC необходимо

- перейти в каталог установки СП WFM CC ***INSTALL_PATH/bin*** и выполнить команду:

```
./runjboss.sh thread-dump
```

дамп будет сформирован в текущем каталоге ***INSTALL_PATH/bin***

4.2.1.2 heapdump и threaddump Сервиса Личный кабинет WFM CC

Предполагается, что

1. при настройке docker-контейнера настроен проброс каталогов из контейнера в ОС (см. п.3.2.7.5 Файлы конфигурации) и формирование дампов выполняется именно в такой каталог (обычно внутри контейнера это /argus/logs, а со стороны ОС это /argus/mobile-api-lk/logs)
2. размер дискового пространства точки монтирования внешнего каталога ОС (обычно это /argusmobile-api-lk/logs) достаточен для сохранения в нем дампов

Для того, чтобы снять **heapdump** сервиса уведомлений необходимо

Перейти в контейнер

```
docker container exec -it имя_контейнера /bin/bash
```

И выполнить

```
export DATE=`date +%Y-%m-%d-%H_%M_%S`
jcmd 1 GC.heap_dump /argus/logs/heap_dump_$DATE
```

Для того, чтобы снять **threaddump** сервиса уведомлений необходимо

Перейти в контейнер

```
docker container exec -it имя_контейнера /bin/bash
```

И выполнить

```
export DATE=`date +%Y-%m-%d-%H_%M_%S`
jstack 1 > /argus/logs/threaddump_$DATE.txt
```

4.2.1.3 heapdump и threaddump Сервиса Мобильный API WFM CC

предполагается, что

1. при настройке docker-контейнера настроен проброс каталогов из контейнера в ОС (см. п.3.2.7.5 Файлы конфигурации) и формирование дампов выполняется именно в такой каталог (обычно внутри контейнера это /argus/logs, а со стороны ОС это /argus/mobile-api/logs)
2. размер дискового пространства точки монтирования внешнего каталога ОС (обычно это /argus/mobile-api/logs) достаточен для сохранения в нем дампов

для того, чтобы снять **heapdump** сервиса уведомлений необходимо

перейти в контейнер

```
docker container exec -it имя_контейнера /bin/bash
```

И выполнить

```
export DATE=`date +%Y-%m-%d-%H_%M_%S`
jcmd 1 GC.heap_dump /argus/logs/heap_dump_$DATE
```

Для того, чтобы снять **threaddump** сервиса уведомлений необходимо

Перейти в контейнер

```
docker container exec -it имя_контейнера /bin/bash
```

И выполнить

```
export DATE=`date +%Y-%m-%d-%H_%M_%S`
jstack 1 > /argus/logs/threaddump_$DATE.txt
```

4.2.1.4 heapdump и threaddump Сервиса отчетов

предполагается, что

1. при настройке docker-контейнера настроен проброс каталогов из контейнера в ОС (см. п.3.2.7.5 Файлы конфигурации) и формирование дампов выполняется именно в такой каталог (обычно внутри контейнера это /argus/logs, а со стороны ОС это /argus/reports/logs)
2. размер дискового пространства точки монтирования внешнего каталога ОС (обычно это /argus/reports/logs) достаточен для сохранения в нем дампов

Для того, чтобы снять **heapdump** сервиса уведомлений необходимо

Перейти в контейнер

```
docker container exec -it имя_контейнера /bin/bash
```

И выполнить

```
export DATE=`date +%Y-%m-%d-%H_%M_%S`
jcmd 1 GC.heap_dump /argus/logs/heap_dump_$DATE
```

Для того, чтобы снять **threaddump** сервиса уведомлений необходимо

перейти в контейнер

```
docker container exec -it имя_контейнера /bin/bash
```

И выполнить

```
export DATE=`date +%Y-%m-%d-%H_%M_%S`
jstack 1 > /argus/logs/threaddump_$DATE.txt
```

4.2.1.5 heapdump и threaddump Сервиса планирования

1. heapdump и threaddump Сервиса планирования (planning-gw)

Снятие дампов (heapdump и threaddump) Сервиса планирования (planning-gw) возможно через web-UI или из docker-контейнера

1.1 для того, чтобы снять heapdump и threaddump Сервиса планирования (planning-gw) через web-UI необходимо через web-browser ввести соответствующий URL в формате:

- http://ip-address:port/actuator/heapdump (после перехода по URL начнется загрузка хип дампа на сторону клиента)
- http://ip-address:port/actuator/threaddump (результирующий ответ отображается в формате json на веб-странице браузера)

Пример:

Снять heapdump: http://192.168.47.8:9030/actuator/heapdump

Снять threaddump: http://http://192.168.47.8:9030/actuator/threaddump

1.2 для того, чтобы снять heapdump и threaddump Сервиса планирования (planning-gw) из docker-контейнера

предполагается, что

- при настройке docker-контейнера настроен проброс каталогов из контейнера в ОС (см. п.3.2.7.5 Файлы конфигурации) и формирование дампов выполняется именно в такой каталог (обычно внутри контейнера это /argus/logs, а со стороны ОС это /argus/planning-gw/logs)
- размер дискового пространства точки монтирования внешнего каталога ОС (обычно это /argus/planning-gw/logs) достаточен для сохранения в нем дампов

Для того, чтобы снять **heapdump** Сервиса планирования (planning-gw) необходимо

перейти в docker-контейнер

```
docker container exec -it имя_контейнера /bin/bash
```

И выполнить

```
export DATE=`date +%Y-%m-%d-%H_%M_%S`
jcmd 1 GC.heap_dump /argus/logs/heap_dump_$DATE
```

Для того, чтобы снять **threaddump** Сервиса планирования (planning-gw) необходимо

перейти в docker-контейнер

```
docker container exec -it имя_контейнера /bin/bash
```

И выполнить

```
export DATE=`date +%Y-%m-%d-%H_%M_%S`
jstack 1 > /argus/logs/threaddump_$DATE.txt
```

2. heapdump и threaddump Сервиса планирования (planning-service)

для того, чтобы снять heapdump и threaddump Сервиса планирования (planning-service) из docker-контейнера

предполагается, что

- при настройке docker-контейнера настроен проброс каталогов из контейнера в ОС (см. п.3.2.7.5 Файлы конфигурации) и формирование дампов выполняется именно в такой каталог (обычно внутри контейнера это /argus/logs, а со стороны ОС это /argus/planning-service/logs)

- размер дискового пространства точки монтирования внешнего каталога ОС (обычно это /argus/planning-service/logs) достаточен для сохранения в нем дампов

Для того, чтобы снять **heapdump** Сервиса планирования (planning-service) необходимо

Перейти в docker-контейнер

```
docker container exec -it имя_контейнера /bin/bash
```

И выполнить

```
export DATE=`date +%Y-%m-%d-%H_%M_%S`
jcmd 1 GC.heap_dump /argus/logs/heap_dump_$DATE
```

Для того, чтобы снять **threaddump** Сервиса планирования (planning-gw) необходимо

перейти в docker-контейнер

```
docker container exec -it имя_контейнера /bin/bash
```

И выполнить

```
export DATE=`date +%Y-%m-%d-%H_%M_%S`
jstack 1 > /argus/logs/threaddump_$DATE.txt
```

4.2.1.6 heapdump и threaddump Сервиса уведомлений

предполагается, что

1. при настройке docker-контейнера настроен проброс каталогов из контейнера в ОС (см. п.3.2.7.5 Файлы конфигурации) и формирование дампов выполняется именно в такой каталог (обычно внутри контейнера это /argus/logs, а со стороны ОС это /argus/notification-service/logs)
2. размер дискового пространства точки монтирования внешнего каталога ОС (обычно это /argus/notification-service/logs) достаточен для сохранения в нем дампов

Для того, чтобы снять **heapdump** сервиса уведомлений необходимо

Перейти в контейнер

```
docker container exec -it имя_контейнера /bin/bash
```

И выполнить

```
export DATE=`date +%Y-%m-%d-%H_%M_%S`
jcmd 1 GC.heap_dump /argus/logs/heap_dump_$DATE
```

Для того, чтобы снять **threaddump** сервиса уведомлений необходимо

Перейти в контейнер

```
docker container exec -it имя_контейнера /bin/bash
```

И выполнить

```
export DATE=`date +%Y-%m-%d-%H_%M_%S`
jstack 1 > /argus/logs/threaddump_$DATE.txt
```

4.2.2 Лог-файлы

В каталоге /argus расширение .log имеют лог-файлы только за текущую дату, лог-файлы за другие даты упаковываются в архив средствами приложения и имеют расширение .gz

4.2.2.1 логи Сервера приложений

Для скачивания лог-файлов СП необходимо любым удобным способом, например по sftp, подключиться к хосту СП и скачать целиком каталог /argus/jboss_prod/standalone/log

4.2.2.2 логи Сервиса Личный Кабинет

Для скачивания лог-файлов Сервиса Личный Кабинет необходимо любым удобным способом, например по sftp, подключиться к хосту Сервиса Личный Кабинет и скачать целиком каталог /argus/mobile-api-lk/logs и /argus/personal-area/logs

4.2.2.3 логи Мобильного API

Для скачивания лог-файлов Мобильного API необходимо любым удобным способом, например по sftp, подключиться к хосту Мобильного API и скачать целиком каталог /argus/mobile-api/logs

4.2.2.4 логи Сервиса планирования

Для скачивания лог-файлов Сервиса планирования необходимо любым удобным способом, например по sftp, подключиться к хосту Сервиса планирования и скачать целиком каталоги /argus/planning-gw/logs и /argus/planning-service/logs

4.2.2.5 логи Сервиса отчетов

Для скачивания лог-файлов СО необходимо любым удобным способом, например по sftp, подключиться к хосту СО и скачать целиком каталог /argus/reports/logs

4.2.2.6 логи Сервиса уведомлений

Для скачивания лог-файлов СУ необходимо любым удобным способом, например по sftp, подключиться к хосту СУ и скачать целиком каталог /argus/notification-service/log

4.2.2.7 логи Сервиса интеграций

Для скачивания лог-файлов СИ необходимо любым удобным способом, например по sftp, подключиться к хосту СИ и скачать целиком каталог /argus/integration/log

Лист Регистрации Изменений

Документ	Дата правки	Исполнитель	Краткое описание внесенного изменения
1	21.10.2021	Трифонов А.А.	Базовая версия
2	11.05.2021	Трифонов А.А.	Добавлена глава 3.2.4.6 Доступ к сервису Мобильный API по HTTPS
3	20.05.2021	Трифонов А.А.	Добавлена глава 3.2.7.6 Настройка почтовых уведомлений
4	03.06.2022	Трифонов А.А.	3.2.3 Сервис Личный кабинет WFM CC 3.2.4 Сервис Мобильный API WFM CC 3.2.5 Сервис планирования 3.2.6 Сервис отчетов 3.2.7 Сервис уведомлений в JAVA_OPTS добавлен параметр -XX:MinRAMPercentage=10.0
5	06.06.2022	Трифонов А.А.	Описание настроек проксирования 3.2.7.2 Установка обновлений сервиса уведомлений 3.2.7.5 Файлы конфигурации
6	07.06.2022	Трифонов А.А.	3.2.8.3 Настройка сервиса интеграций дополнение описания
7	09.06.2022	Трифонов А.А.	Добавлена глава 4.1.1 pgdump
8	30.06.2022	Трифонов А.А.	Добавлена глава 4.2 Справочник администратора СП и сервисов
9	04.07.2022	Трифонов А.А	Добавлена глава 4.2.2 Лог-файлы

Список принятых сокращений

БД	База Данных
КТС	Комплекс Технических Средств
ЛВС	Локальная Вычислительная Сеть
ОС	Операционная Система
ПО	Программное Обеспечение
СИ	Сервис Интеграций
СО	Сервис Отчетов
СП	Сервер Приложений
СУ	Сервис уведомлений
ТА	Техническая Архитектура
ТЗ	Техническое Задание
WFM CC	Work Force Management Call Center

Приложения

Параметры мониторинга

Параметры мониторинга СП

Item Key

```
agent.hostname
agent.ping
agent.version
avg.servlet.page.response.time
increment-avg-servlet-page-response-time
jmx["java.lang:type=GarbageCollector,name=PS MarkSweep", "CollectionCount"]
jmx["java.lang:type=GarbageCollector,name=PS MarkSweep", "CollectionTime"]
jmx["java.lang:type=GarbageCollector,name=PS Scavenge", "CollectionCount"]
jmx["java.lang:type=GarbageCollector,name=PS Scavenge", "CollectionTime"]
jmx["java.lang:type=Memory","HeapMemoryUsage.used"]
jmx["java.lang:type=Memory","NonHeapMemoryUsage.used"]
jmx["java.lang:type=Threading", "ThreadCount"]
jmx["jboss.as.expr:data-
source=ArgusDS,subsystem=\"datasources\",statistics=\"pool\"", "AvailableCount"]
jmx["jboss.as.expr:deployment=ccwfm-app-
{$APP_VERSION}.ear,subsystem=\"undertow\",subdeployment=\"webui-
{$APP_VERSION}.war\"","activeSessions"]
jmx["jboss.as.expr:deployment=ccwfm-app-
{$APP_VERSION}.ear,subsystem=\"undertow\",subdeployment=\"webui-
{$APP_VERSION}.war\"","sessionsCreated"]
jmx["jboss.as.expr:subsystem=argus,request-
resource=RequestResource","pageRequestCount"]
jmx["jboss.as.expr:subsystem=argus,request-
resource=RequestResource","totalPageRequestTime"]
jmx["jboss.as:subsystem=argus,request-resource=RequestResource","pageRequestCount"]
jmx["jboss.as:subsystem=argus,request-
resource=RequestResource","totalPageRequestTime"]
jmx["jboss.as:subsystem=argus,worker-resource=default","activeCount"]
jmx["jboss.as:subsystem=argus,worker-resource=default","completedTaskCount"]
jmx["jboss.as:subsystem=argus,worker-resource=default","taskCount"]
jvm.request.resource[{$JMX_USERNAME},{$JMX_PASSWORD}]
jvm.worker.resource[{$JMX_USERNAME},{$JMX_PASSWORD}]
kernel.maxfiles
kernel.maxproc
Network interface discovery: Incoming network traffic on docker0
Network interface discovery: Incoming network traffic on ens224
Network interface discovery: Outgoing network traffic on docker0
Network interface discovery: Outgoing network traffic on ens224
net.tcp.service[http,{HOST.IP},8080]
proc.cpu.util[java,argus]
proc.num[,,run]
proc.num[]
system.boottime
system.cpu.intr
system.cpu.load[percpu,avg1]
system.cpu.load[percpu,avg5]
system.cpu.load[percpu,avg15]
system.cpu.switches
system.cpu.util[,idle]
system.cpu.util[,interrupt]
system.cpu.util[,iowait]
system.cpu.util[,nice]
system.cpu.util[,softirq]
system.cpu.util[,steal]
system.cpu.util[,system]
```

```
system.cpu.util[,user]
system.hostname
system.localtime
system.swap.in[,pages]
system.swap.out[,pages]
system.swap.size[,free]
system.swap.size[,pfree]
system.swap.size[,total]
system.uname
system.uptime
system.users.num
vfs.file.cksum[/etc/passwd]
Mounted filesystem discovery: Free inodes on / (percentage)
Mounted filesystem discovery: Free inodes on /argus (percentage)
Mounted filesystem discovery: Free disk space on /
Mounted filesystem discovery: Free disk space on / (percentage)
Mounted filesystem discovery: Total disk space on /
Mounted filesystem discovery: Used disk space on /
Mounted filesystem discovery: Free disk space on /argus
Mounted filesystem discovery: Free disk space on /argus (percentage)
Mounted filesystem discovery: Total disk space on /argus
Mounted filesystem discovery: Used disk space on /argus
vm.memory.size[available]
vm.memory.size[total]
```

Параметры мониторинга сервера БД

Item Key

```
pgsql.archive_command.archived_files[{$PG_CONNINFO}]
pgsql.archive_command.count_files_to_archive[{$PG_CONNINFO}]
pgsql.archive_command.failed_trying_to_archive[{$PG_CONNINFO}]
pgsql.archive_command.size_files_to_archive[{$PG_CONNINFO}]
pgsql.autovacumm.count[{$PG_CONNINFO}]
pgsql.bgwriter.buffers_alloc[{$PG_CONNINFO}]
pgsql.bgwriter.buffers_backend[{$PG_CONNINFO}]
pgsql.bgwriter.buffers_backend_fsync[{$PG_CONNINFO}]
pgsql.bgwriter.buffers_checkpoint[{$PG_CONNINFO}]
pgsql.bgwriter.buffers_clean[{$PG_CONNINFO}]
pgsql.bgwriter.maxwritten_clean[{$PG_CONNINFO}]
pgsql.blocks.hit[{$PG_CONNINFO}]
pgsql.blocks.read[{$PG_CONNINFO}]
pgsql.buffers.dirty[{$PG_CONNINFO}]
pgsql.buffers.size[{$PG_CONNINFO}]
pgsql.buffers.twice_used[{$PG_CONNINFO}]
pgsql.cache.hit[{$PG_CONNINFO}]
pgsql.checkpoint.checkpoint_sync_time[{$PG_CONNINFO}]
pgsql.checkpoint.count_timed[{$PG_CONNINFO}]
pgsql.checkpoint.count_wal[{$PG_CONNINFO}]
pgsql.checkpoint.write_time[{$PG_CONNINFO}]
pgsql.connections.active[{$PG_CONNINFO}]
pgsql.connections.disabled[{$PG_CONNINFO}]
pgsql.connections.fastpath_function_call[{$PG_CONNINFO}]
pgsql.connections.idle[{$PG_CONNINFO}]
pgsql.connections.idle_in_transaction[{$PG_CONNINFO}]
pgsql.connections.idle_in_transaction_aborted[{$PG_CONNINFO}]
pgsql.connections.max_connections[{$PG_CONNINFO}]
pgsql.connections.total[{$PG_CONNINFO}]
pgsql.connections.waiting[{$PG_CONNINFO}]
```

```
Database discovery: Count of bloating tables in database: integration
Database discovery: Count of bloating tables in database: opta
Database discovery: Max age (datfrozenxid) in: integration
Database discovery: Max age (datfrozenxid) in: opta
Database discovery: Database integration: size
Database discovery: Database opta: size
pgsql.events.checksum_failures[{$PG_CONNINFO}]
pgsql.events.conflicts[{$PG_CONNINFO}]
pgsql.events.deadlocks[{$PG_CONNINFO}]
pgsql.events.xact_rollback[{$PG_CONNINFO}]
pgsql.oldest.transaction_time[{$PG_CONNINFO}]
pgsql.oldest.xid_age[{$PG_CONNINFO}]
pgsql.pg_locks.accessexclusive[{$PG_CONNINFO}]
pgsql.pg_locks.accessshare[{$PG_CONNINFO}]
pgsql.pg_locks.exclusive[{$PG_CONNINFO}]
pgsql.pg_locks.rowexclusive[{$PG_CONNINFO}]
pgsql.pg_locks.rowshare[{$PG_CONNINFO}]
pgsql.pg_locks.sharerowexclusive[{$PG_CONNINFO}]
pgsql.pg_locks.shareupdateexclusive[{$PG_CONNINFO}]
pgsql.pg_locks.share[{$PG_CONNINFO}]
pgsql.ping[{$PG_CONNINFO}]
pgsql.replication_lag.sec[{$PG_CONNINFO}]
pgsql.stat.dirty_bytes[{$PG_CONNINFO}]
pgsql.stat.other_time[{$PG_CONNINFO}]
pgsql.stat.read_bytes[{$PG_CONNINFO}]
pgsql.stat.read_time[{$PG_CONNINFO}]
pgsql.stat.write_bytes[{$PG_CONNINFO}]
pgsql.stat.write_time[{$PG_CONNINFO}]
pgsql.temp.bytes[{$PG_CONNINFO}]
pgsql.temp.files[{$PG_CONNINFO}]
pgsql.transactions.total[{$PG_CONNINFO}]
pgsql.tuples.deleted[{$PG_CONNINFO}]
pgsql.tuples.fetched[{$PG_CONNINFO}]
pgsql.tuples.inserted[{$PG_CONNINFO}]
pgsql.tuples.returned[{$PG_CONNINFO}]
pgsql.tuples.updated[{$PG_CONNINFO}]
pgsql.uptime[{$PG_CONNINFO}]
pgsql.wal.count[{$PG_CONNINFO}]
pgsql.wal.write[{$PG_CONNINFO}]
system.cpu.idle
system.cpu.iowait
system.cpu.irq
system.cpu.nice
system.cpu.softirq
system.cpu.system
system.cpu.user
system.disk.all_read
system.disk.all_read_b
system.disk.all_write
system.disk.all_write_b
system.la.1
system.memory.active
system.memory.apps
system.memory.buffers
system.memory.cached
system.memory.committed
system.memory.inactive
system.memory.mapped
system.memory.page_tables
system.memory.slab
system.memory.swap
system.memory.swap_cache
system.memory.unused
```

```
system.memory.vmalloc_used
Net iface discovery: Network device eth0: RX bytes/s
Net iface discovery: Network device eth0: RX drops/s
Net iface discovery: Network device eth0: RX errors/s
Net iface discovery: Network device eth0: TX bytes/s
Net iface discovery: Network device eth0: TX drops/s
Net iface discovery: Network device eth0: TX errors/s
system.open_files
system.processes.blocked
system.processes.forkrate
system.processes.running
system.up_time
VFS discovery: Mount point /boot: free
VFS discovery: Mount point /proc/sys/fs/binfmt_misc: free
VFS discovery: Mount point /: free
VFS discovery: Mount point /boot: free in percents
VFS discovery: Mount point /proc/sys/fs/binfmt_misc: free in percents
VFS discovery: Mount point /: free in percents
VFS discovery: Mount point /boot: used
VFS discovery: Mount point /proc/sys/fs/binfmt_misc: used
VFS discovery: Mount point /: used
```