|  |  |
| --- | --- |
|  |  |
|  |  |
|  |  |

Платформа бизнес-аналитики  
«Alpha BI»

Руководство администратора

АБСМ.506100.001.И4

Содержание

[Определения, обозначения и сокращения 11](#_Toc174374225)

[1 Введение 16](#_Toc174374226)

[1.1 Область применения 16](#_Toc174374227)

[1.2 Краткое описание возможностей 16](#_Toc174374228)

[1.3 Уровень подготовки пользователей 16](#_Toc174374229)

[1.4 Перечень эксплуатационной документации 16](#_Toc174374230)

[2 Назначение и условие применения 17](#_Toc174374231)

[2.1 Виды деятельности и автоматизируемые функции в Системе 17](#_Toc174374232)

[2.2 Условия применения 17](#_Toc174374233)

[2.2.1 Требования к техническому обеспечению 17](#_Toc174374234)

[2.2.1.1 Аппаратные требования по размещению Системы 17](#_Toc174374235)

[2.2.1.2 Требования к обеспечению каналами связи 18](#_Toc174374236)

[2.2.1.3 Требования к техническому обеспечению клиентских машин 19](#_Toc174374237)

[2.2.2 Требования к программному обеспечению 19](#_Toc174374238)

[2.2.3 Языки программирования 22](#_Toc174374239)

[3 Функциональная архитектура Системы 23](#_Toc174374240)

[3.1 Описание компонентов Системы 23](#_Toc174374241)

[3.1.1 Бизнес-объекты 23](#_Toc174374242)

[3.1.2 OLAP 24](#_Toc174374243)

[3.1.3 ETL 26](#_Toc174374244)

[3.1.3.1 Монитор ETL 27](#_Toc174374245)

[3.1.4 Планировщик 28](#_Toc174374246)

[3.1.5 Формы 28](#_Toc174374247)

[3.1.6 Регламентные отчеты 29](#_Toc174374248)

[3.1.7 Аналитические панели 31](#_Toc174374249)

[3.1.8 Сообщения 34](#_Toc174374250)

[3.1.9 Файловое хранилище 34](#_Toc174374251)

[3.1.10 Валидация компонентов Системы 35](#_Toc174374252)

[4 Подготовка к работе 41](#_Toc174374253)

[4.1 Топология 41](#_Toc174374254)

[4.1.1 Конфигурационные параметры 42](#_Toc174374255)

[4.1.2 Состав дистрибутива 44](#_Toc174374256)

[4.2 Развертывание серверов 44](#_Toc174374257)

[4.2.1 Сервер БД 44](#_Toc174374258)

[4.2.1.1 Зависимости 44](#_Toc174374259)

[4.2.1.1.1 Ntpd 44](#_Toc174374260)

[4.2.1.2 Установка 45](#_Toc174374261)

[4.2.1.3 Конфигурирование и запуск 46](#_Toc174374262)

[4.2.2 Сервер OLAP 47](#_Toc174374263)

[4.2.2.1 Вариант 1. Развертывание сервера OLAP с установкой Tomcat из репозитория 47](#_Toc174374264)

[4.2.2.1.1 Ntpd 47](#_Toc174374265)

[4.2.2.1.2 Развертывание Tomcat 48](#_Toc174374266)

[4.2.2.1.2.1 Зависимости Tomcat 48](#_Toc174374267)

[4.2.2.1.2.2 Первоначальная установка Tomcat 48](#_Toc174374268)

[4.2.2.1.2.3 Конфигурирование Tomcat 48](#_Toc174374269)

[4.2.2.1.3 Развертывание Mondrian 48](#_Toc174374270)

[4.2.2.1.3.1 Зависимости Mondrian 48](#_Toc174374271)

[4.2.2.1.3.2 Конфигурирование Mondrian 49](#_Toc174374272)

[4.2.2.1.3.3 Установка Mondrian 49](#_Toc174374273)

[4.2.2.2 Вариант 2. Развертывание сервера OLAP с установкой Tomcat дистрибутивом из официального сайта Apache Tomcat 50](#_Toc174374274)

[4.2.2.2.1 Ntpd 50](#_Toc174374275)

[4.2.2.2.2 Развертывание Tomcat 51](#_Toc174374276)

[4.2.2.2.2.1 Зависимости Tomcat 51](#_Toc174374277)

[4.2.2.2.2.2 Первоначальная установка Tomcat 51](#_Toc174374278)

[4.2.2.2.2.3 Конфигурирование Tomcat 51](#_Toc174374279)

[4.2.2.2.2.4 Запуск Tomcat 52](#_Toc174374280)

[4.2.2.2.3 Развертывание Mondrian 53](#_Toc174374281)

[4.2.2.2.3.1 Зависимости Mondrian 53](#_Toc174374282)

[4.2.2.2.3.2 Конфигурирование Mondrian 53](#_Toc174374283)

[4.2.2.2.3.3 Установка Mondrian 54](#_Toc174374284)

[4.2.3 Сервер файлового хранилища 54](#_Toc174374285)

[4.2.3.1 Первоначальная установка 54](#_Toc174374286)

[4.2.3.2 Конфигурирование 55](#_Toc174374287)

[4.2.3.3 Запуск 55](#_Toc174374288)

[4.2.4 Серверы состояния 55](#_Toc174374289)

[4.2.4.1 Установка зависимостей 55](#_Toc174374290)

[4.2.4.1.1 Ntpd 55](#_Toc174374291)

[4.2.4.2 Первоначальная установка 56](#_Toc174374292)

[4.2.4.3 Сборка и запуск кластера Etcd 56](#_Toc174374293)

[4.2.4.3.1 Конфигурирование первого узла кластера Etcd 56](#_Toc174374294)

[4.2.4.3.2 Конфигурирование последующих узлов кластера Etcd 57](#_Toc174374295)

[4.2.4.4 Конфигурирование кластера Etcd 58](#_Toc174374296)

[4.2.5 Cервер балансировки 59](#_Toc174374297)

[4.2.5.1 Зависимости 59](#_Toc174374298)

[4.2.5.1.1 Ntpd 59](#_Toc174374299)

[4.2.5.2 Установка 59](#_Toc174374300)

[4.2.5.3 Конфигурирование и запуск 60](#_Toc174374301)

[4.2.6 Сервер приложения 64](#_Toc174374302)

[4.2.6.1 Установка зависимостей 64](#_Toc174374303)

[4.2.6.1.1 Ntpd 64](#_Toc174374304)

[4.2.6.1.2 NFS 65](#_Toc174374305)

[4.2.6.1.3 Зависимости для .NET 65](#_Toc174374306)

[4.2.6.1.4 Зависимости серверного генератора отчетов 65](#_Toc174374307)

[4.2.6.1.5 Прочие зависимости 65](#_Toc174374308)

[4.2.6.2 Первоначальная установка 66](#_Toc174374309)

[4.2.6.3 Конфигурирование 66](#_Toc174374310)

[4.2.6.3.1 Настройка хранения файлов при кластерном режиме 66](#_Toc174374311)

[4.2.6.3.2 Настройка хранения файлов при некластерном режиме 67](#_Toc174374312)

[4.2.6.3.3 Настройка скрипта запуска Alpha BI 67](#_Toc174374313)

[4.2.6.3.4 Создание конфигурационного файла 67](#_Toc174374314)

[4.2.6.4 Проведение миграций 69](#_Toc174374315)

[4.2.6.5 Запуск 69](#_Toc174374316)

[4.3 Обновление Системы 69](#_Toc174374317)

[4.3.1 Обновление версии приложения 70](#_Toc174374318)

[4.3.2 Обновление Mondrian 71](#_Toc174374319)

[4.3.2.1 Вариант 1 (если использовался вариант установки Tomcat из репозитория) 71](#_Toc174374320)

[4.3.2.2 Вариант 2 (если использовался вариант установки Tomcat вручную дистрибутивом с официального сайта) 71](#_Toc174374321)

[4.3.3 Проведение миграций 71](#_Toc174374322)

[4.4 Конфигурирование Alpha BI 72](#_Toc174374323)

[4.4.1 Параметры конфигурации Alpha BI 72](#_Toc174374324)

[4.4.2 Конфигурация Alpha BI в кластерном режиме 75](#_Toc174374325)

[4.4.2.1 Кластерные параметры конфигурации 75](#_Toc174374326)

[4.4.2.2 Чтение и запись конфигурации через консольное приложение 77](#_Toc174374327)

[4.4.3 Конфигурация Alpha BI через переменные окружения 78](#_Toc174374328)

[4.4.3.1 Переменные окружения – параметры конфигурации Alpha BI 78](#_Toc174374329)

[4.5 Отдельные операции 79](#_Toc174374330)

[4.5.1 Переход из некластерного режима в кластерный режим 79](#_Toc174374331)

[4.5.2 Развертывание имеющегося дампа 80](#_Toc174374332)

[4.5.3 Внешний сервер балансировки 81](#_Toc174374333)

[4.5.3.1 Внешний сервер балансировки Haproxy 81](#_Toc174374334)

[4.5.3.1.1 Установка зависимостей 81](#_Toc174374335)

[4.5.3.1.2 Первоначальная установка 81](#_Toc174374336)

[4.5.3.1.3 Конфигурирование 81](#_Toc174374337)

[4.5.3.1.4 Запуск 84](#_Toc174374338)

[4.5.3.2 Внешний сервер балансировки HAProxy отказоустойчивый 84](#_Toc174374339)

[4.5.3.2.1 Первоначальная установка Keepalived 84](#_Toc174374340)

[4.5.3.2.2 Конфигурирование Keepalived 84](#_Toc174374341)

[4.5.3.2.3 Запуск Keepalived 87](#_Toc174374342)

[4.5.3.3 Внешний сервер балансировки Nginx 87](#_Toc174374343)

[4.5.3.3.1 Установка зависимостей 87](#_Toc174374344)

[4.5.3.3.2 Первоначальная установка 88](#_Toc174374345)

[4.5.3.3.3 Конфигурирование 88](#_Toc174374346)

[4.5.3.3.4 Запуск 88](#_Toc174374347)

[4.5.3.4 Внешний сервер балансировки Nginx отказоустойчивый 89](#_Toc174374348)

[4.5.3.4.1 Первоначальная установка Keepalived 89](#_Toc174374349)

[4.5.3.4.2 Конфигурирование Keepalived 89](#_Toc174374350)

[4.5.3.4.3 Запуск Keepalived 92](#_Toc174374351)

[4.5.4 Внутренний сервер балансировки 92](#_Toc174374352)

[4.5.4.1 Внутренний сервер балансировки Haproxy 92](#_Toc174374353)

[4.5.4.1.1 Зависимости 92](#_Toc174374354)

[4.5.4.1.2 Первоначальная установка 92](#_Toc174374355)

[4.5.4.1.3 Конфигурирование 92](#_Toc174374356)

[4.5.4.1.4 Запуск 96](#_Toc174374357)

[4.5.4.2 Внутренний сервер балансировки HAProxy отказоустойчивый 96](#_Toc174374358)

[4.5.4.2.1 Первоначальная установка Keepalived 96](#_Toc174374359)

[4.5.4.2.2 Конфигурирование Keepalived 96](#_Toc174374360)

[4.5.4.2.3 Запуск Keepalived 99](#_Toc174374361)

[4.5.4.3 Внутренний сервер балансировки Nginx 99](#_Toc174374362)

[4.5.4.3.1 Зависимости 99](#_Toc174374363)

[4.5.4.3.2 Первоначальная установка 100](#_Toc174374364)

[4.5.4.3.3 Конфигурирование 100](#_Toc174374365)

[4.5.4.3.4 Запуск 101](#_Toc174374366)

[4.5.4.4 Внутренний сервер балансировки Nginx отказоустойчивый 101](#_Toc174374367)

[4.5.4.4.1 Первоначальная установка Keepalived 102](#_Toc174374368)

[4.5.4.4.2 Конфигурирование Keepalived 102](#_Toc174374369)

[4.5.4.4.3 Запуск Keepalived 105](#_Toc174374370)

[4.5.5 Настройка расположения файлов приложения 105](#_Toc174374371)

[4.5.5.1 Настройки расположения файлов приложения в Linux 105](#_Toc174374372)

[4.5.5.1.1 Вариант 1 105](#_Toc174374373)

[4.5.5.1.2 Вариант 2 106](#_Toc174374374)

[4.5.5.2 Настройка расположения файлов приложения в Windows 107](#_Toc174374375)

[4.5.5.2.1 Вариант 1 107](#_Toc174374376)

[4.5.5.2.2 Вариант 2 107](#_Toc174374377)

[4.5.6 Сервер файлового хранилища отказоустойчивый 110](#_Toc174374378)

[4.5.6.1 Первоначальная установка NFS 110](#_Toc174374379)

[4.5.6.2 Конфигурирование NFS 110](#_Toc174374380)

[4.5.6.3 Запуск NFS 111](#_Toc174374381)

[4.5.6.4 Первоначальная установка Lsyncd 111](#_Toc174374382)

[4.5.6.5 Конфигурирование Lsyncd 111](#_Toc174374383)

[4.5.6.6 Запуск Lsyncd 113](#_Toc174374384)

[4.5.6.7 Первоначальная установка Keepalived 113](#_Toc174374385)

[4.5.6.8 Конфигурирование Keepalived 113](#_Toc174374386)

[4.5.6.9 Запуск Keepalived 115](#_Toc174374387)

[4.6 Запуск Системы 116](#_Toc174374388)

[4.7 Настройка Системы для авторизации и аутентификации 117](#_Toc174374389)

[4.8 Порядок проверки работоспособности 117](#_Toc174374390)

[4.8.1 Просмотр и настройка логов на серверах приложений 118](#_Toc174374391)

[4.8.1.1 Просмотр и настройка логов Alpha BI 118](#_Toc174374392)

[4.8.1.1.1 Размещение лог-файлов и настройка уровней логирования 118](#_Toc174374393)

[4.8.1.1.2 Структура лог-файлов 119](#_Toc174374394)

[4.8.1.2 Логирование Mondrian 120](#_Toc174374395)

[4.8.1.2.1 Общая информация 120](#_Toc174374396)

[4.8.1.2.2 Настройка расширенного логирования 121](#_Toc174374397)

[4.8.1.2.2.1 Изменение уровня логирования Mondrian 121](#_Toc174374398)

[4.8.1.2.2.2 Включение логирования MDX-запросов 121](#_Toc174374399)

[4.8.1.2.2.3 Включение логирования SQL-запросов 122](#_Toc174374400)

[4.8.1.2.2.4 Изменение размера и количества файлов логов Mondrian 122](#_Toc174374401)

[4.8.1.3 Логирование PostgreSQL 123](#_Toc174374402)

[4.8.1.3.1 Общая информация 123](#_Toc174374403)

[4.8.1.3.2 Настройка расширенного логирования 123](#_Toc174374404)

[4.8.1.3.2.1 Включение логирования подключений, выполнения контрольных точек, блокировок 123](#_Toc174374405)

[4.8.1.3.2.2 Включение логирования длительности запросов и создания временных файлов 124](#_Toc174374406)

[4.8.1.3.2.3 Использование pgbadger для анализа логов PostgreSQL 124](#_Toc174374407)

[4.8.2 Просмотр и настройка логов в Системе 125](#_Toc174374408)

[4.8.2.1 Просмотр лога действий 125](#_Toc174374409)

[4.8.2.2 Просмотр лога авторизации 128](#_Toc174374410)

[4.8.2.3 Монитор сессий пользователей 129](#_Toc174374411)

[5 Описание операций 132](#_Toc174374412)

[5.1 Работа в контрольной панели 132](#_Toc174374413)

[5.1.1 Лицензия 133](#_Toc174374414)

[5.1.1.1 Объекты лицензии 134](#_Toc174374415)

[5.1.1.2 Виды лицензий в Системе 134](#_Toc174374416)

[5.1.1.3 Ключи 135](#_Toc174374417)

[5.1.1.4 Описание порядка активации основной лицензии 135](#_Toc174374418)

[5.1.1.5 Описание порядка активации дополнительной лицензии 136](#_Toc174374419)

[5.1.1.6 Виды ключей 137](#_Toc174374420)

[5.1.1.7 Состав программного продукта 138](#_Toc174374421)

[5.1.1.8 Коды ключей 138](#_Toc174374422)

[5.1.1.9 Форма и состав стандартной поставки программного продукта 138](#_Toc174374423)

[5.1.1.10 Обновление ПО 139](#_Toc174374424)

[5.1.1.11 Прочие условия 139](#_Toc174374425)

[5.1.2 Требуемое обслуживание 139](#_Toc174374426)

[5.1.3 Состояние 141](#_Toc174374427)

[5.1.4 Протоколы синхронизации 141](#_Toc174374428)

[5.1.5 Информация 142](#_Toc174374429)

[5.1.6 Выход 143](#_Toc174374430)

[6 Аварийные ситуации 144](#_Toc174374431)

[6.1 Нештатные ситуации 144](#_Toc174374432)

[7 Рекомендации по освоению 150](#_Toc174374433)

[7.1 Последовательность действий при работе с Системой 150](#_Toc174374434)

# Определения, обозначения и сокращения

В настоящем документе применяют следующие термины и сокращения с соответствующими определениями и обозначениями:

| Термин, сокращение | Определение, обозначение |
| --- | --- |
| Active Directory | «Активный каталог» – LDAP-совместимая реализация службы каталогов корпорации Microsoft для операционных систем семейства Windows NT. Active Directory позволяет использовать групповые политики для обеспечения единообразия настройки пользовательской рабочей среды, развертывать программное обеспечение на множестве компьютеров через групповые политики или посредством System Center Configuration Manager, устанавливать обновления операционной системы, прикладного и серверного программного обеспечения на всех компьютерах в сети, используя Службу обновления Windows Server. Active Directory хранит данные и настройки среды в централизованной базе данных |
| API | Application Programming Interface – интерфейс программирования приложений, интерфейс прикладного программирования |
| Backup-сервер | Резервный сервер, вступающий в работу, когда master-сервер не функционирует |
| BI | Business Intelligence – инструменты для поиска, анализа, моделирования и доставки информации, необходимой для принятия решения |
| CPU | Central Processing Unit – центральное процессорное устройство |
| drag-and-drop | Способ оперирования элементами интерфейса в интерфейсах пользователя (как графическим, так и текстовым) при помощи манипулятора «мышь» или сенсорного экрана |
| DWH | Data Warehouse – предметно-ориентированная информационная база данных, специально разработанная и предназначенная для подготовки отчетов и бизнес-анализа с целью поддержки принятия решений в организации |
| EPEL | Extra Packages for Enterprise Linux – репозиторий, доступный для установки из стандартных репозиториев CentOS |
| ETL | Extract, Transform, Load – один из основных процессов в управлении хранилищами данных, который включает в себя извлечение данных из внешних источников, их трансформацию и очистку, чтобы они соответствовали потребностям бизнес-модели и загрузка данных в хранилище данных |
| HAProxy | Серверное программное обеспечение для обеспечения высокой доступности и балансировки нагрузки для TCP и HTTP-приложений, посредством распределения входящих запросов на несколько обслуживающих серверов |
| HDD | Hard (magnetic) disk drive – накопитель на жестких магнитных дисках, жесткий диск − запоминающее устройство (устройство хранения информации), основанное на принципе магнитной записи |
| HTML | HyperText Markup Language – стандартизированный язык разметки документов во Всемирной паутине |
| HTTP | HyperText Transfer Protocol – протокол прикладного уровня передачи данных в первую очередь в виде текстовых сообщений. Основой HTTP является технология «клиент-сервер», то есть предполагается существование потребителей (клиентов), которые инициируют соединение и посылают запрос, и поставщиков (серверов), которые ожидают соединения для получения запроса, производят необходимые действия и возвращают обратно сообщение с результатом |
| HTTPS | HyperText Transfer Protocol Secure – расширение протокола HTTP, поддерживающее шифрование. Данные, передаваемые по протоколу HTTP, «упаковываются» в криптографический протокол SSL, тем самым обеспечивается защита этих данных |
| ICU | International Components for Unicode – библиотека с открытым исходным кодом для C/C++ и Java, обеспечивающая поддержку Unicode и интернационализацию программного обеспечения |
| IIS | Internet Information Service – проприетарный набор серверов для нескольких служб Интернета от компании Майкрософт. Основным компонентом IIS является веб-сервер, который позволяет размещать в Интернете сайты |
| IP | Internet Protocol – маршрутизируемый протокол сетевого уровня стека TCP/IP |
| IP-адрес | Internet Protocol Address – уникальный сетевой адрес узла в компьютерной сети, построенной по протоколу IP |
| JRE | Java Runtime Environment |
| JSON | JavaScript Object Notation – простой формат обмена данными, удобный для чтения и написания как человеком, так и компьютером |
| JVM | Java Virtual Machine – виртуальная машина Java – основная часть исполняющей системы Java, так называемой Java Runtime Environment (JRE) |
| LDAP | Lightweight Directory Access Protocol – протокол прикладного уровня для доступа к службе каталогов X.500, использующий TCP/IP и позволяющий производить операции аутентификации, поиска и сравнения, а также операции добавления, изменения или удаления объектов в службе каталогов |
| Master-сервер | Основной сервер, выполняющий свои функции при штатном режиме работы системы |
| MDX | Media Data eXtended – язык запросов для доступа к многомерным структурам данных. |
| NFS | Network File System – протокол сетевого доступа к файловым системам |
| Node | (Node – узел) – компьютер, входящий в состав кластера |
| Ntpd | Network Time Protocol daemon – программа, которая устанавливает и поддерживает системное время, используется для синхронизации серверами точного времени |
| OLAP | Online Analytical Processing – технология обработки данных, заключающаяся в подготовке суммарной (агрегированной) информации на основе больших массивов данных, структурированных по многомерному принципу |
| OLAP-представление | Интерфейс многомерной иерархической таблицы, построенной на данных куба по технологии OLAP |
| OLAP-сервер | Сервер, с помощью которого проводится обработка многомерных структур данных и обеспечивается связь между БД и пользователями системы |
| OpenID | Система единого входа (авторизации) на сайты, порталы, блоги и форумы |
| OpenJDK | Проект по созданию полностью совместимого Java Development Kit, состоящего исключительно из свободного и открытого исходного кода |
| OpenSSL | Криптографический пакет с открытым исходным кодом для работы с SSL/TLS. Позволяет создавать ключи RSA, DH, DSA и сертификаты X.509, подписывать их, формировать CSR и CRT. Также имеется возможность шифрования данных и тестирования SSL/TLS соединений |
| PostgreSQL | Свободная объектно-реляционная система управления базами данных |
| RAM | Random Access Memory – оперативная память – энергозависимая часть системы компьютерной памяти, в которой во время работы компьютера хранится выполняемый машинный код (программы), а также входные, выходные и промежуточные данные, обрабатываемые процессором |
| RHEL | Red Hat Enterprise Linux – дистрибутив Linux компании Red Hat |
| SAML | Security assertion markup language – язык разметки, основанный на языке XML. Открытый стандарт обмена данными аутентификации и авторизации между участниками, в частности, между поставщиком учетных записей (IdP) и поставщиком сервиса (SP). Язык Security Assertion Markup Language (SAML) является открытым стандартом безопасного обмена данными аутентификации и авторизации между корпоративным провайдером идентификации и провайдером сервиса. Этот подход называется также SAML Web Single Sign On (Технологией единого веб-входа SAML) |
| SAS | Serial Attached SCSI (Small Computer System Interface) – последовательный компьютерный интерфейс, разработанный для подключения различных устройств хранения данных, например, жестких дисков и ленточных накопителей |
| SELinux | Security-Enhanced Linux – реализация системы принудительного контроля доступа, которая может работать параллельно с классической избирательной системой контроля доступа |
| SMTP | Simple Mail Transfer Protocol – простой протокол передачи почты – широко используемый сетевой протокол, предназначенный для передачи электронной почты в сетях TCP/IP |
| SQL | Structured Query Language – формальный непроцедурный язык программирования, применяемый для создания, модификации и управления данными в произвольной реляционной базе данных, управляемой соответствующей системой управления базами данных (СУБД) |
| SSH | Secure Shell – «безопасная оболочка» — сетевой протокол прикладного уровня, позволяющий производить удаленное управление операционной системой и туннелирование TCP-соединений (например, для передачи файлов) |
| SSL | Криптографический протокол, обеспечивающий безопасную передачу данных по сети Интернет. При его использовании создается защищенное соединение между клиентом и сервером |
| TCP/IP | Набор сетевых протоколов передачи данных, используемых в сетях, включая сеть Интернет |
| TLS | Transport Layer Security – безопасность транспортного уровня – криптографические протоколы, обеспечивающие защищённую передачу данных между узлами в сети Интернет. Используют асимметричную криптографию для аутентификации, симметричное шифрование для конфиденциальности и коды аутентичности сообщений для сохранения целостности сообщений |
| URL | Uniform Resource Locator – стандартизированный способ записи адреса ресурса в сети Интернет |
| UTF-8 | Unicode Transformation Format, 8-bit – распространённый стандарт кодирования символов, позволяющий более компактно хранить и передавать символы Юникода, используя переменное количество байт (от 1 до 4), и обеспечивающий полную обратную совместимость с 7-битной кодировкой ASCII |
| Web-браузер | Программное обеспечение для поиска, просмотра web-страниц (преимущественно из сети Интернет), их обработки, вывода и перехода от одной страницы к другой. Например, Google Chrome, Opera, Mozilla Firefox и т.п. |
| XML | eXtensible Markup Language — расширяемый язык разметки |
| XMLA | XML for Analysis – отраслевой стандарт доступа к данным в аналитических системах, таких как OLAP и интеллектуальный анализ данных |
| zip | Свободный файловый архиватор с высокой степенью сжатия данных |
| Администратор Системы | Сотрудник, наделенный полномочиями управления Системой |
| АО «БАРС Груп» | Акционерное общество «БАРС Груп» |
| БД | База данных |
| БО | Бизнес-объект |
| ГБ | Гигабайт |
| ГБ/с | Гигабайт в секунду |
| ГГц | Гигагерц |
| ГК РФ | Гражданский кодекс Российской Федерации |
| Иерархия (иерархия размерности) | Логическая структура размерности, в которой в качестве средства организации и объединения данных используются упорядоченные уровни (например, иерархия размерности «Период» объединяет уровни: «Год», «Квартал», «Месяц») |
| Инстанс | Экземпляр приложения, инсталляция с единой БД и лицензионным ключом |
| Кластер | Вычислительная система, состоящая из множества независимых компьютеров, связанных между собой каналами передачи данных |
| Куб | Многомерная структура, содержащая размерности (измерения) и меры |
| МБ | Мегабайт |
| Мера | Числовое значение (объем продаж, площадь и т.п.), представляющее интерес для конечного пользователя. Показатель (признак, атрибут), количественно описывающий исследуемый процесс или объект |
| ОЗУ | Оперативное запоминающее устройство |
| ОС | Операционная система |
| Патч | Информация, предназначенная для автоматизированного внесения определенных изменений в компьютерные файлы |
| ПО | Программное обеспечение |
| Раздел | Элемент логической структуры Системы, который может включать подразделы |
| Размерность | То, чем характеризуется цифра в массиве данных (товар, продавец, учреждение и т.п.) и определяет структуру данных |
| Редактор цвета | Панель задания координат цвета в системах HEX, RGB и HSV с поддержкой альфа-канала RGBA, HSVA |
| РСУБД | Реляционная система управления базами данных |
| РФ | Российская Федерация |
| Система, Alpha BI | Платформа бизнес-аналитики «Alpha BI» |
| СУБД | Система управления базами данных |
| Транзакция | Группа логически объединенных последовательных операций по работе с данными, обрабатываемая или отменяемая целиком |
| ФИО | Фамилия, имя, отчество |
| ЦОД | Центр обработки данных |

# Введение

## Область применения

Платформа бизнес-аналитики «Alpha BI» (далее – Система, Alpha BI) предназначена для консолидации значений основных статистических и качественных показателей, касающихся деятельности организации (Заказчика), их обработки и представления, позволяющие руководству организации (Заказчика) оперативно принимать управленческие решения.

## Краткое описание возможностей

Система предназначена для решения следующих задач:

* сбор оперативной информации, касающейся деятельности организации (Заказчика);
* построение единого хранилища статистических и качественных показателей (индикаторов);
* детализация показателей в разрезе учреждений/структурных подразделений, временных промежутков;
* информативная визуализация собранных и поступающих данных;
* получение оперативных OLAP-отчетов и регламентных отчетов (в том числе формирование пользователем шаблонов отчетов).

## Уровень подготовки пользователей

Администратор Системы должен иметь опыт администрирования ОС Linux, ПО PostgreSQL, Apache Tomcat, Nginx, в случае развертывания кластера – дополнительно опыт администрирования ПО HAProxy, Etcd, NFS-server.

Если Система устанавливается на ОС Windows – опыт администрирования ОС Windows, ПО IIS, PostgreSQL, Apache Tomcat.

## Перечень эксплуатационной документации

Для работы с Системой ознакомьтесь с данным руководством.

# Назначение и условие применения

## Виды деятельности и автоматизируемые функции в Системе

Основными целями создания Системы являются:

* формирование единого информационного пространства для организации (Заказчика);
* формирование оперативной аналитической информации (отчетности) и ее визуализация, на основе собственных данных, а также данных из смежных информационных систем;
* обеспечение необходимого уровня качества данных, используемых в отчетности, за счет их детализации до первоисточников.

## Условия применения

### Требования к техническому обеспечению

#### Аппаратные требования по размещению Системы

Заданные показатели назначения и надежности Системы должны осуществляться при условии выполнения следующих аппаратных требований по размещению Системы:

* сервер БД:
  + поддерживаемые процессорные архитектуры: x86-64, Arm64;
  + ресурсы CPU: не менее 8 ядер по 2 ГГц;
  + ресурсы RAM: не менее 16 ГБ;
  + объем HDD: не менее 64 ГБ, рекомендуемый тип используемых дисков SAS (10000 об/мин) или 20 iops на 1 ГБ.
* сервер приложения:
  + поддерживаемые процессорные архитектуры: x86-64, Arm64;
  + ресурсы CPU: не менее 8 ядер по 2 ГГц;
  + ресурсы RAM: не менее 16 ГБ;
  + объем HDD: не менее 32 ГБ, рекомендуемый тип используемых дисков SAS (10000 об/мин) или 2 iops на 1 ГБ.
* OLAP-сервер:
  + поддерживаемые процессорные архитектуры: x86-64, Arm64;
  + ресурсы CPU: не менее 8 ядер по 2 ГГц;
  + ресурсы RAM: не менее 16 ГБ;
  + объем HDD: не менее 32 ГБ, рекомендуемый тип используемых дисков SAS (10000 об/мин) или 10 iops на 1 ГБ.
* серверы для организации кластерного режима работы:
  + сервер балансировки:
* поддерживаемые процессорные архитектуры: x86-64, Arm64;
* ресурсы CPU: от 1 ядра по 2 ГГц;
* ресурсы RAM: не менее 2 ГБ;
* объем HDD: не менее 10 ГБ, рекомендуемый тип используемых дисков SAS (7200 об/мин) или 1 iops на 1 ГБ.
  + сервер состояния:
* поддерживаемые процессорные архитектуры: x86-64, Arm64;
* ресурсы CPU: от 2 ядер по 2 ГГц;
* ресурсы RAM: не менее 8 ГБ;
* объем HDD: не менее 32 ГБ, рекомендуемый тип используемых дисков SAS (7200 об/мин) или 2 iops на 1 ГБ.
  + сервер файлового хранилища:
* поддерживаемые процессорные архитектуры: x86-64, Arm64;
* ресурсы CPU: от 1 ядра по 2 ГГц;
* ресурсы RAM: не менее 2 ГБ;
* объем HDD: не менее 64 ГБ, рекомендуемый тип используемых дисков SAS (10000 об/мин) или 2 iops на 1 ГБ.

#### Требования к обеспечению каналами связи

Для связи между всеми серверами приложения: web-приложение, OLAP, БД, сервера состояния, файловое хранилище, балансировщики и т. д., используется канал с пропускной способностью не менее 1 Гбит/сек.

Характеристики каналов связи для пользователей представлены в таблице ниже (Таблица 1).

Таблица  – Каналы связи

| Требования к количеству пользователей, подключенных к каналу связи | Требования к каналам связи между клиентскими машинами и севером приложений |
| --- | --- |
| От 1 до 50 пользователей | Канал связи: 10 Мбит/с |
| От 50 до 100 пользователей | Канал связи: 30 Мбит/с |
| От 100 до 200 пользователей | Канал связи: 70 Мбит/с |
| От 200 пользователей | Канал связи: 100 Мбит/с |
| Примечание – Подразумевается стабильный канал связи | |

#### Требования к техническому обеспечению клиентских машин

Для клиентских машин устанавливаются следующие минимальные технические требования:

* процессор не менее 2 ядер по 2 ГГц;
* ОЗУ – 4 ГБ;
* объем свободного места на диске – не менее 5 ГБ;
* сетевая карта 1 ГБ/с;
* монитор, поддерживающий разрешение не менее 1920x1080;
* клавиатура, манипулятор типа мышь.

### Требования к программному обеспечению

Для обеспечения функционирования Системы необходимо следующее программное обеспечение:

1. сервер РСУБД:

* СУБД PostgreSQL версии 12 или выше

Примечание – Рекомендуется использовать последнюю доступную версию PostgreSQL.

* одна из следующих ОС:
  + Microsoft Windows Server 2012 R2 или выше;
  + дистрибутивы Linux:
* Ubuntu 20.04 или выше;
* CentOS 7 или выше;
* Alpine Linux 3.14 и выше;
* Debian 10 и выше;
* RHEL 7 и выше;
* РЕД ОС 7.3;
* Astra Linux 1.7.

Примечание – Рекомендуется использовать Ubuntu 22.04.

1. сервер web-приложения:

* одна из следующих ОС:
  + операционная система Microsoft Windows:
* Microsoft Windows Server 2012 R2 или выше.
  + дистрибутивы Linux:
* Ubuntu 20.04 или выше;
* CentOS 7 или выше;
* Alpine Linux 3.14 и выше;
* Debian 10 и выше;
* RHEL 7 и выше;
* РЕД ОС 7.3;
* Astra Linux 1.7.

Примечание – Рекомендуется использовать Ubuntu 22.04.

* для операционной системы Microsoft Windows:
  + .NET Framework 4.7.2 или выше;
  + IIS 7 или выше;
  + ASP.NET Core Runtime 6.0.
* для операционной системы Linux:
  + .NET Runtime 6.0.
* для операционных систем Microsoft Windows и Linux:
  + Java 11 или выше.

Примечание – Рекомендуется использовать версию OpenJDK 17 или AxiomJDK 17.

* поддерживаемые прокси-серверы:
  + Nginx;
  + Haproxy.

1. OLAP-сервер:

* одна из следующих ОС:
  + Microsoft Windows:
* Microsoft Windows Server 2012 R2 и выше.
  + дистрибутивы Linux:
* Ubuntu 20.04 или выше;
* CentOS 7 или выше;
* Alpine Linux 3.14 и выше;
* Debian 10 и выше;
* RHEL 7 и выше;
* РЕД ОС 7.3;
* Astra Linux 1.7.

Примечание – Рекомендуется использовать Ubuntu 22.04.

* web-сервер Apache Tomcat версии 9 или 10.1;

Примечание – Рекомендуется использовать последнюю версию Apache Tomcat 10.1.

* рабочая среда для Tomcat – Java 8 (JRE) или выше.

Примечание – Рекомендуется использовать версию OpenJDK 17 или AxiomJDK 17.

1. сервер балансировки:

* одна из следующих ОС:
  + Linux:
* Ubuntu 20.04 или выше;
* CentOS 7 или выше;
* Alpine Linux 3.14 и выше;
* Debian 10 и выше;
* RHEL 7 и выше;
* РЕД ОС 7.3;
* Astra Linux 1.7.

Примечание – Рекомендуется использовать Ubuntu 22.04.

1. сервер состояния:

* одна из следующих ОС:
  + Linux:
* Ubuntu 20.04 или выше;
* CentOS 7 или выше;
* Alpine Linux 3.14 и выше;
* Debian 10 и выше;
* RHEL 7 и выше;
* РЕД ОС 7.3;
* Astra Linux 1.7.

Примечание – Рекомендуется использовать Ubuntu 22.04.

1. сервер файлового хранилища:

* одна из следующих ОС:
  + Linux:
* Ubuntu 20.04 или выше;
* CentOS 7 или выше;
* Alpine Linux 3.14 и выше;
* Debian 10 и выше;
* RHEL 7 и выше;
* РЕД ОС 7.3;
* Astra Linux 1.7.

Примечание – Рекомендуется использовать Ubuntu 22.04.

Требования к обеспечению клиентских рабочих мест:

* web-браузеры:
  + Microsoft Edge версии 125 или выше;
  + Google Chrome версии 125 или выше;
  + Mozilla Firefox версии 127 или выше;
  + Safari версии 17.5 или выше;
  + Opera версии 110 или выше;
  + Яндекс.Браузер версии 24.4.3 и выше.
* одна из следующих ОС: Microsoft Windows, Linux (Unix), MacOS, IOS и др., поддерживающая работу указанных web-браузеров.

Примечание – После выхода новой версии Системы необходимо обновлять используемый web-браузер до последней актуальной версии.

### Языки программирования

Используемые в Системе языки программирования:

* C# – используется в разработке серверной части приложения;
* Java Script – используется в разработке клиентской части приложения;
* HTML – используется в разработке клиентской части приложения;
* SQL – используется для управления данными в реляционной базе данных;
* MDX – используется для запросов к многомерным хранилищам данных;
* Java – язык программирования OLAP-сервера Mondrian.

# Функциональная архитектура Системы

## Описание компонентов Системы

Общая функциональная архитектура Системы представлена набором компонентов, отображенных на схеме ниже (Рисунок 1).

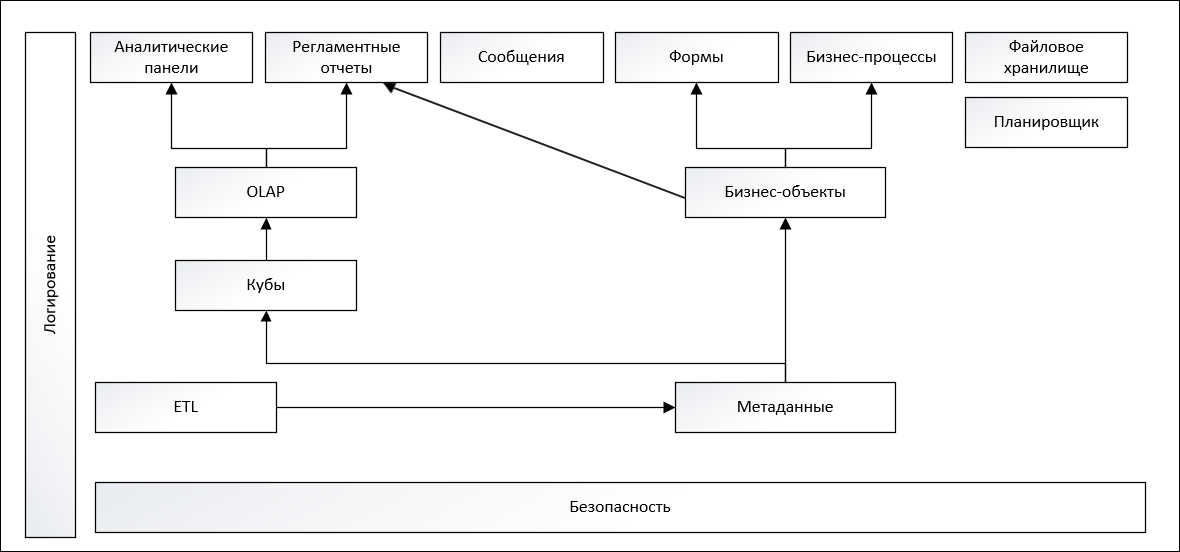


Рисунок  – Общая функциональная архитектура Системы

### Бизнес-объекты

Компонент «Бизнес-объекты» предназначен для конструирования структур, хранения данных и определения правил их использования. Содержит метаданные всех системных объектов и является одним из основных компонентов в архитектуре Системы.

Функции бизнес-объектов (далее – БО):

* создание структур в реляционном хранилище, в том числе добавление и удаление полей различных типов данных;
* дополнительная настройка полей БО (подробное описание приведено в разделе 4 документа «Платформа бизнес-аналитики «Alpha BI». Руководство пользователя. АБСМ.506100.001.И3»);
* просмотр записей БО;
* версионирование записей;
* просмотр версий записи в версионных БО и возможность сделать актуальной одну из предыдущих версий записи;
* отображение количества записей БО;
* редактирование и удаление записей;
* сортировка записей БО по возрастанию/убыванию;
* поиск записей в столбце БО;
* отображение/скрытие столбца БО;
* копирование БО;
* копирование записей БО;
* экспорт данных в файл .xlsx.

Схема компонента представлена ниже (Рисунок 2).



Рисунок  – Схема компонента «Бизнес-объекты»

### OLAP

Компонент «OLAP» предназначен для создания многомерных кубов и проведения анализа данных.

Общая схема компонента «OLAP» представлена ниже (Рисунок 3).

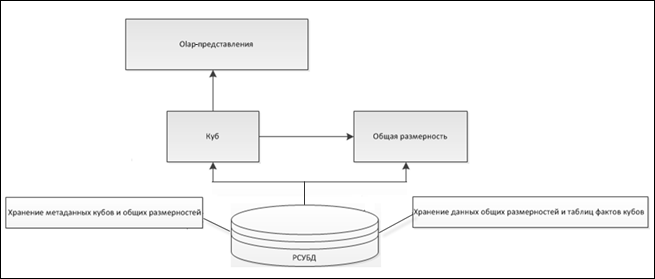


Рисунок  – Общая схема компонента «OLAP»

Функции компонента «OLAP»:

* управление многомерным хранилищем, которое включает:
  + создание, удаление кубов, мер и общих размерностей;
  + редактирование метаданных кубов и общих размерностей;
  + добавление расчетных элементов кубов;
  + создание, удаление и редактирование виртуальных кубов;
  + просмотр таблицы фактов куба и общей размерности.
* проведение анализа данных:
  + построение аналитических таблиц с использованием технологии OLAP (т.е. построение OLAP-представлений);
  + поддержка контекстно-зависимых действий: детализация данных ячейки из стороннего источника, переход к другой таблице, возможность создания пользовательских действий. Функция детализации поддерживает механизм drill-through (выборка из фактов куба, на основании которых сформирована конкретная ячейка OLAP). В окне детализации данных, на основании которых формировалось значение ячейки таблицы;
  + выбор используемых мер и размерностей до выполнения запроса данных;
  + включение/отключение автоматической актуализации запроса данных при изменении настроек куба;
  + добавление вычисляемых элементов в OLAP-представления на основе уже имеющихся в кубе элементов и других расчетных элементов, созданных ранее на пользовательском уровне, без внесения изменений в структуру хранилища данных;
  + сброс настроек, связанных с оформлением текущего запроса OLAP-представления;
  + копирование сохраненных OLAP-представлений;
  + поддержка использования переменных внутри запроса;
  + выбор расположения размерностей (измерений) в строках и столбцах таблицы с использованием drag-and-drop, а также настройка их последовательности, ручное транспонирование строк в столбцы и обратно;
  + расчет вычисляемых итогов по строкам и столбцам таблицы;
  + возможность скрытия элементов, использованных в кубе, без осуществления повторного запроса данных;
  + выбор отображения атрибутов размерностей, использованных в кубе;
  + фильтрация полученной таблицы по произвольному набору использованных в ней мер и размерностей, задание сложносоставных фильтров, возможность поиска при создании фильтров;
  + просмотр результатов запроса в форме диаграммы, графика;
  + сортировка столбцов таблицы по возрастанию и убыванию, с разрывом иерархичности и без (в случае использования иерархий размерностей);
  + выгрузка полученной таблицы в файл формата .xlsx;
  + формирование регламентных отчетов на основании OLAP-представления.

### ETL

Компонент предназначен для извлечения данных (extract), преобразования данных (transform), то есть приведения их к необходимому формату, обработки в соответствии с определенными правилами, комбинировании с другими данными и т.п., а также для загрузки данных (load), записи данных в хранилище или в другую базу.

ETL-процессы создаются пользователями Системы и состоят из набора ETL -блоков (реже из одного блока) с установленными между ними связями. Конфигурация всех ETL-процессов Системы хранится в реляционном хранилище в виде файлов формата .json.

Схема компонента «ETL» представлена ниже (Рисунок 4).

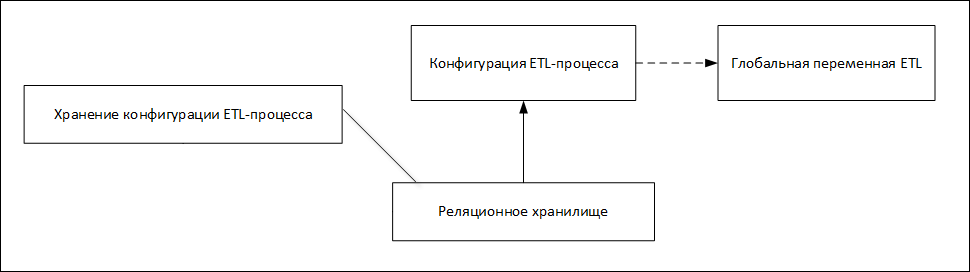


Рисунок  – Схема компонента «ETL»

Функции компонента:

* настройка алгоритмов обработки данных в форме дерева блоков-обработчиков;
* возможность графического отображения структуры алгоритма;
* подсветка совместимых блоков при создании ETL-процессов;
* настройка периодичности обработки данных с указанием даты начала выполнения алгоритма, его завершения, периодичности запуска и имперсонацией;
* информирование администратора Системы о результатах выполнения алгоритмов;
* контроль целостности данных, транзакционность операций с данными;
* предпросмотр результатов работы для отдельных блоков-обработчиков без запуска алгоритма, включая блоки, не являющиеся конечными в дереве алгоритма;
* генерация внешней ссылки для каждого блока-обработчика;
* автозаполнение набора полей структуры данных при их наследовании по дереву алгоритма.

#### Монитор ETL

Компонент предназначен для просмотра результатов запусков ETL-процессов, отображает следующую информацию:

* название и код ETL-процесса;
* дата и время запуска и завершения ETL-процесса;
* текущее состояние ETL-процесса;
* длительность выполнения ETL-процесса;
* имя пользователя, запустившего ETL-процесс;
* текущее состояние ETL-процесса;
* каким образом был инициирован запуск ETL-процесса: с помощью планировщика или непосредственно через ETL-процесс;
* если в результате выполнения ETL-процесса возникла ошибка, отображается текст ошибки.

### Планировщик

Планировщик позволяет производить запуск фоновых процессов (например, ETL-процессов) по расписанию.

Реализован с использованием внешнего компонента – Quartz.NET. Схема компонента «Планировщик» представлена ниже (Рисунок 5).



Рисунок  – Схема компонента «Планировщик»

### Формы

Компонент «Формы вывода» предназначен для создания форм ввода и хранения первичных и сводных отчетных данных в единой базе данных (Рисунок 6).

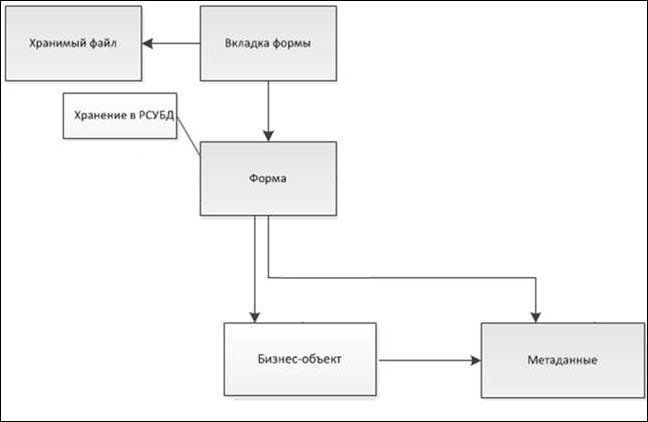


Рисунок  – Схема компонента «Формы»

Система позволяет создавать динамические формы вывода с поддержкой функций фильтрации и автоматического расчета значений в ячейках. Формы вывода в Системе формируются на основе метаданных Системы с реляционным типом хранилища данных.

Доступна настройка параметров для каждой создаваемой формы вывода. Параметры формы предназначены для формирования пользователем среза представления данных при работе с формой.

Максимальное количество параметров формы зависит от набора полей этих метаданных.

Реализован экспорт данных формы в файл в формате .xlsx.

### Регламентные отчеты

Компонент «Регламентные отчеты» предназначен для создания регламентных отчетов, источниками данных для которых могут выступать кубы и БО Системы.

Схема компонента «Регламентные отчеты» приведена ниже (Рисунок 7).

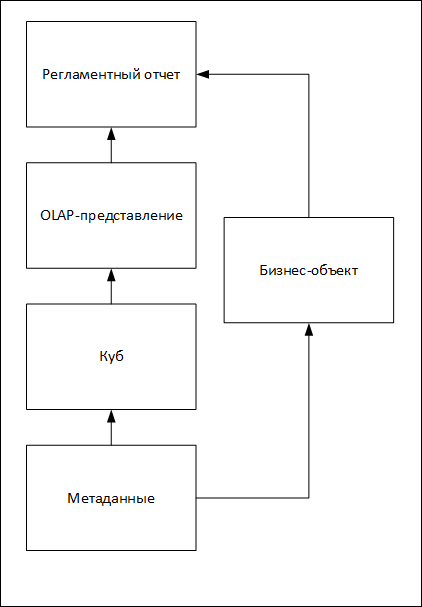


Рисунок  – Схема компонента «Регламентные отчеты»

Шаблоны регламентных отчетов хранятся в реляционном хранилище в виде текста формата .json.

Для создания регламентных отчетов используются OLAP-представления и бизнес-объекты. В OLAP-представлениях можно определить состав отображаемых данных, выбрав подходящие характеристики (размерности) и меры куба. Для создания регламентного отчета на одной его странице могут быть размещены несколько OLAP-представлений. В регламентном отчете можно создать несколько страниц. Также в регламентном отчете можно разместить статическую информацию (произвольно набранный текст).

Функции компонента «Регламентные отчеты»:

* макет регламентного отчета для экспорта может настраиваться пользователями Системы при наличии прав доступа;
* макет регламентного отчета для экспорта может настраиваться пользователями сразу в web-интерфейсе Системы;
* экспортируемый регламентный отчет может содержать текст, таблицы;
* сформированный регламентный отчет экспортируется в виде файла формата. xlsx и данные могут выбираться в момент вызова регламентного отчета;
* на этапе настройки реализовано форматирование отдельных ячеек регламентных отчетов;
* допустимо использование расчетных элементов.

### Аналитические панели

Компонент «Аналитические панели» предназначен для настройки и просмотра информации в виде графиков, диаграмм, карт, таблиц. Схема компонента «Аналитические панели» приведена ниже (Рисунок 8).

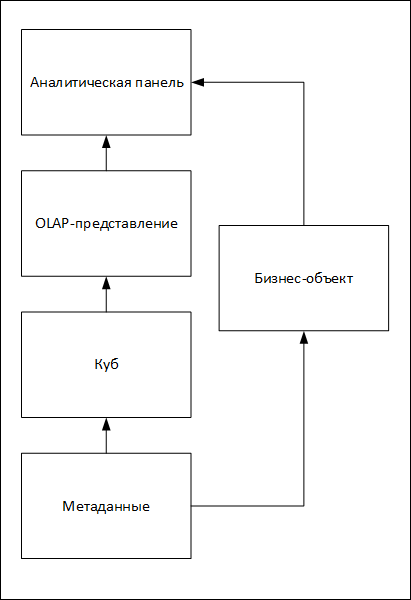


Рисунок  – Схема компонента «Аналитические панели»

Функции аналитических панелей:

* создание, удаление аналитических панелей;
* добавление набора областей, связанных между собой пространственными указателями, имеющих длину и ширину и служащих контейнерами для виджетов;
* пространственная конфигурация контейнеров;
* добавление на аналитическую панель произвольного количества различных типов виджетов;
* элементы, отображающие значения показателей, могут скрываться при отображении;
* добавление динамического разделителя между виджетами (при необходимости);
* редактирование уже созданных панелей и их отдельных элементов. В редакторе аналитических панелей реализована возможность перемещения виджетов между контейнерами и изменение размеров виджета/контейнера с помощью drag-and-drop;
* настройки цветовых схем виджетов с помощью редактора цвета;
* настройка источников данных для виджетов. В качестве источников выступают реестры БО и сформированные на основе кубов хранилища данных OLAP-представления и иерархия OLAP-представления. Применимость источников зависит от типа виджета. Совместимость источников данных с виджетами описана в таблице ниже (Таблица 2). Символ «+» на пересечении названия виджета с названием вида источника данных означает, что к виджету можно применить данный вид источника. Символ «-» означает, что к виджету данный вид источника применить невозможно;
* в рамках аналитической панели может поддерживаться использование переменных.

Таблица  – Совместимость источника данных и виджета

| Виджет | OLAP | Иерархия OLAP | Реестр | Иерархический реестр |
| --- | --- | --- | --- | --- |
| Обычный список | + | + | + | + |
| Выпадающий список | + | + | + | + |
| Иерархический список | + | + | - | - |
| Переключатель | + | + | + | + |
| OLAP-представление | + | - | - | - |
| Столбчатая диаграмма | + | - | + | + |
| Линейная диаграмма | + | - | + | + |
| Радар | + | - | + | + |
| Круговая диаграмма | + | - | + | + |
| Пирамида | + | - | + | + |
| Карта | + | - | - | - |
| Древовидная карта | + | - | + | + |

Компонент содержит настройки визуального отображения различных виджетов: цвета, шрифты, границы, стиль, интервалы, отступы, фон, выравнивание.

### Сообщения

Компонент «Сообщения» предоставляет интерфейс для создания, удаления, отправки/приема и сортировки сообщений, с обеспечением фиксации времени отправки, получения и авторства сообщений.

Компонент обеспечивает не только отправку внутрисистемных сообщений, но и сообщений на внешние e-mail-адреса посредством SMTP-сервера.

### Файловое хранилище

Файловое хранилище – это выделенная область, предназначенная для хранения файлов произвольного формата.

Общий объем файлового хранилища ограничен свободным дисковым пространством сервера, на котором оно размещено. Можно использовать как локальный каталог, так и каталог на удаленном сервере.

Для ОС Microsoft Windows максимальный объем загружаемого файла ограничен настройками в файле web.config. Для увеличения допустимого объема файла необходимо изменить значение maxAllowedContentLength, как правило, установлено 1073741823 бит.

Пример – Пример секции представлен ниже:

<requestFiltering>

<requestLimits maxUrl ="10999" maxQueryString ="2097151" maxAllowedContentLength ="1073741823" />

</requestFiltering>

Для ОС Microsoft Windows для увеличения таймаута времени загрузки файла в файловое хранилище, возможно, потребуется изменение значения параметра <httpRuntime executionTimeout="12000"> в файле web.config. Значение указано в миллисекундах.

### Валидация компонентов Системы

Валидация предназначена для выявления ошибок в компонентах Системы, которые в последующем могут затруднить или сделать невозможной работу с ними.

Ниже представлено описание механизма внутренней работы валидации состояния объектов Системы.

Основные задачи механизма валидации компонентов:

* обеспечение согласованного состояния компонентов, образующих рабочее приложение для конечного пользователя;
* обеспечение быстрой обратной связи пользователю в процессе редактирования компонентов.

Результатом валидации Системы является перечень ошибок. Каждая ошибка характеризуется:

* компонентом, к которому данная ошибка относится;
* списком компонентов, которые влияют на ошибку (являются причиной ошибки);
* идентификатором не найденного компонента (если ошибка выявлена в ссылке на компонент);
* сообщением об ошибке.

Механизм валидации запускается при выполнении следующих операций:

1. при запуске Системы – запускается валидация для проверки всех компонентов Системы перед их активацией. При наличии ошибок пользователю выдается сообщение с детальным описанием полученных ошибок (Рисунок 9);

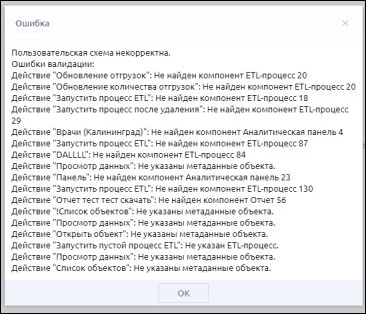


Рисунок  – Ошибки валидации при старте приложения

1. при публикации объектов – при нажатии кнопки «Синхронизировать» запускается валидация для проверки всех компонентов перед синхронизацией (Рисунок 10);

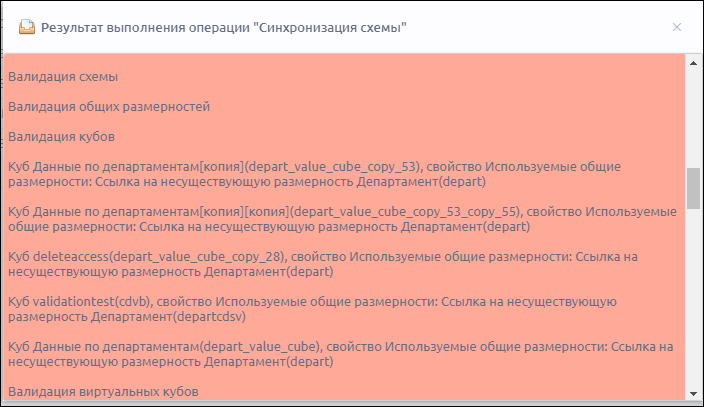


Рисунок  – Ошибки валидации при публикации объектов

1. при импорте компонентов – последовательно запускается валидация всех схем с применением к ним пакета импорта и проверяется возможность импорта (Рисунок 11);

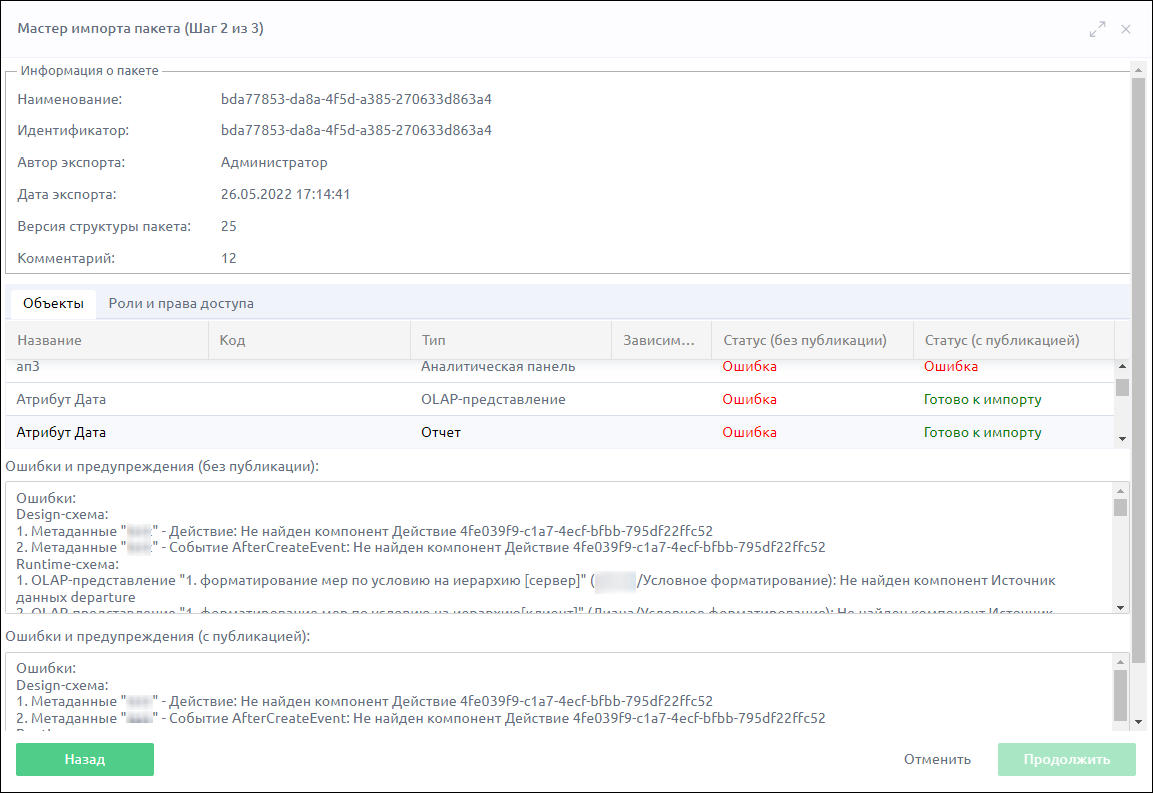


Рисунок  – Ошибки валидации при импорте компонентов

Примечание – Валидация играет большую роль при осуществлении экспорта и импорта компонентов. Механизм осуществления экспорта и импорта описан в документе «Платформа бизнес-аналитики «Alpha BI». Руководство пользователя. АБСМ.506100.001.И3».

1. при сохранении изменений объектов, требующих публикации – запускается валидация, соответствующая тому состоянию объектов, которые в настоящий момент редактируются и изменения в которых применяются только после публикации. Если в результате внесенных изменений хотя бы один компонент становится невалидным, то Система считается невалидной и внесенные изменения отклоняются (Рисунок 12);

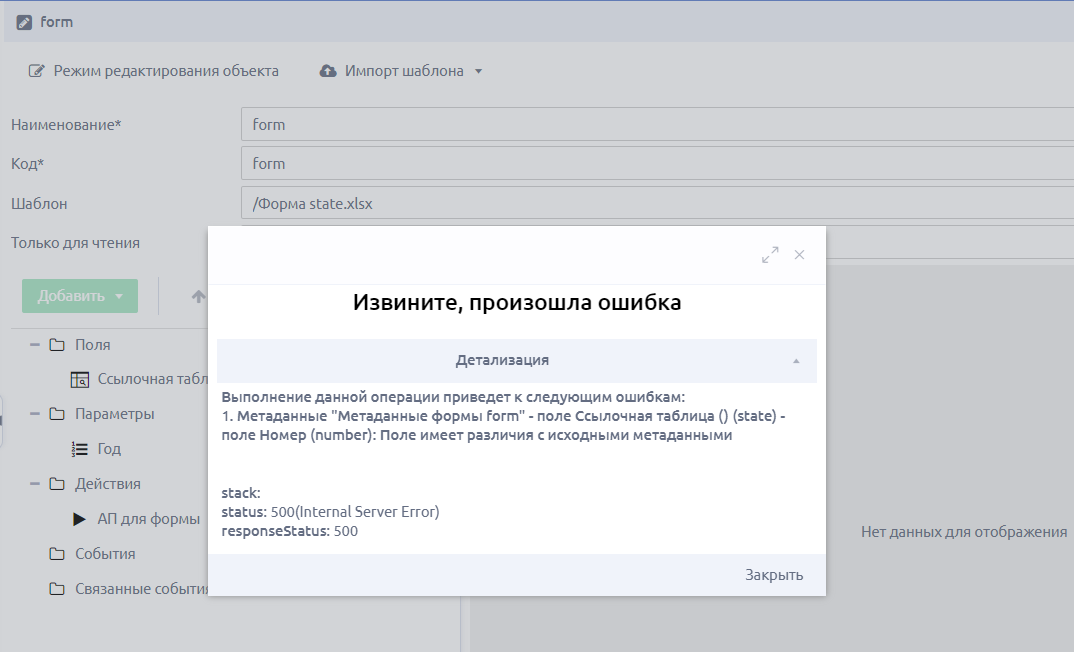


Рисунок  – Ошибка валидации при сохранении изменений

1. при редактировании опубликованного объекта – запускается валидация, которая строится в процессе редактирования опубликованных компонентов и компонентов, которые редактируются, минуя этап публикации. Если в результате внесенных изменений хотя бы один компонент становится невалидным, то вся Система считается невалидной, и изменения отклоняются (Рисунок 13).

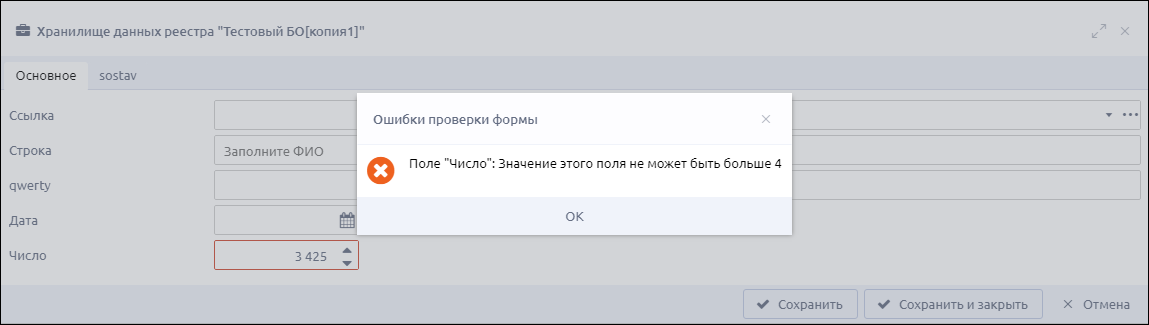


Рисунок  – Ошибка валидации при редактировании опубликованного объекта

При формировании сообщений об ошибках валидации учитываются права доступа пользователя.

В случае выявления ошибки администратору Системы выдается сообщение с детальным описанием полученных ошибок. Пользователю Системы сообщение об ошибке может быть выдано в следующих форматах:

* если у пользователя есть доступ ко всем компонентам, в которых есть ошибки, то сообщение об ошибке будет аналогично тому, что выдается администратору Системы (Рисунок 14);

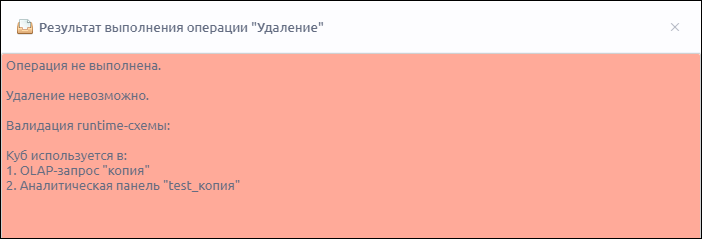


Рисунок  – Сообщение об ошибке валидации

* если у пользователя нет доступа ни к одному компоненту с ошибками, то в тексте сообщения для исправления ошибок валидации пользователю будет предложено обратиться к администратору Системы (Рисунок 15);

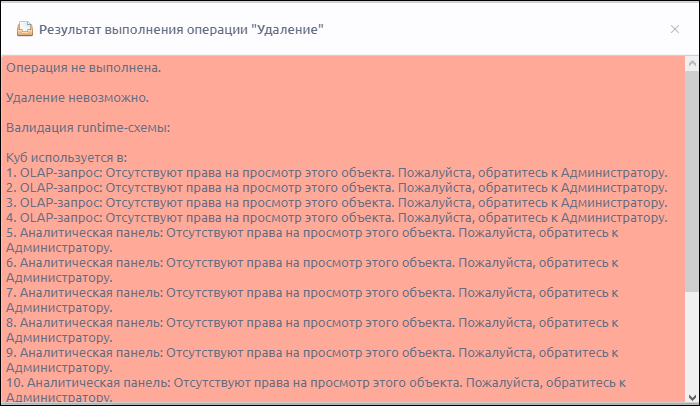


Рисунок  – Сообщение об ошибке валидации для пользователя, у которого отсутствуют права на компоненты Системы

* если у пользователя есть доступ не ко всем компонентам с ошибками, пользователю будет выдано сообщение с описанием ошибок в доступных компонентах и предложением обратиться к администратору для исправления ошибок в недоступных ему компонентах (Рисунок 16).

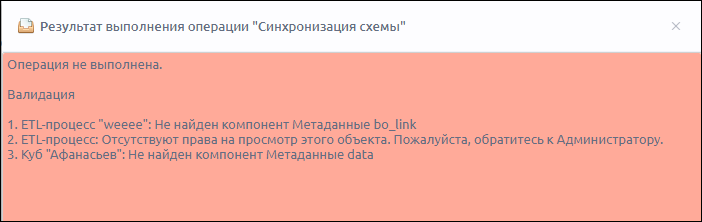


Рисунок  – Сообщение об ошибке валидации для пользователя, у которого присутствуют права на некоторые компоненты Системы

# Подготовка к работе

## Топология

На диаграмме ниже представлена наиболее общая топология развертывания Системы в кластерном режиме (Рисунок 17).

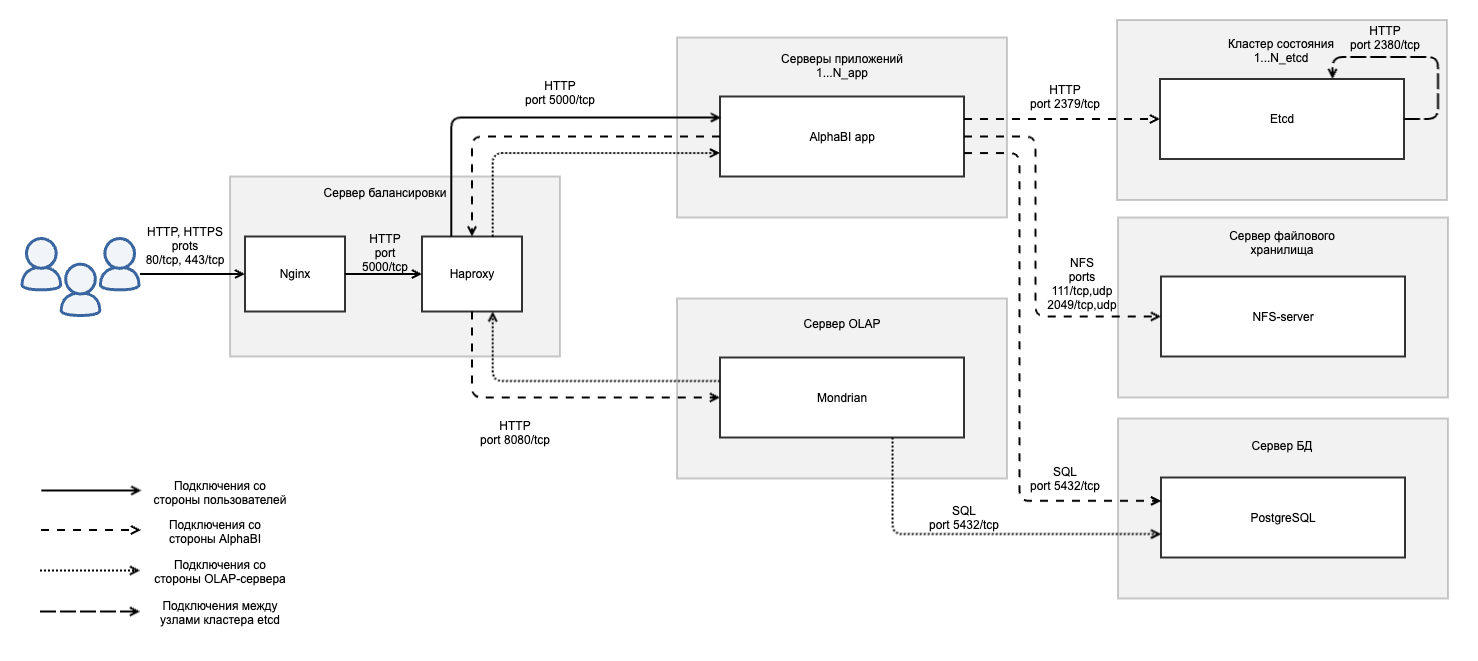


Рисунок  – Наиболее общая топология развертывания Системы в кластерном режиме

Система состоит из следующих серверов:

* сервер балансировки – единственный сервер, непосредственно доступный по сети для конечных пользователей. Отвечает за маршрутизацию запросов к web-приложению и терминирование TLS-соединений (при использовании HTTPS). Сервер, отвечающий за маршрутизацию запросов между компонентами Системы. Реализует отказоустойчивость и масштабируемость за счет балансировки входящих HTTP-запросов между серверами приложений;
* сервер (серверы) web-приложения Alpha BI – основная часть Alpha BI – сервер, отвечающий за обработку пользовательских запросов. Возможно наличие более одного сервера для отказоустойчивости и масштабируемости;
* сервер OLAP – сервер, отвечающий за обработку аналитических запросов к данным;
* серверы состояния – серверы, отвечающие за синхронизацию нескольких серверов приложения Alpha BI при запуске в режиме кластера;
* сервер файлового хранилища – сервер, отвечающий за хранение и доступ к файлам, которые были загружены пользователями и сформированы Системой;
* сервер БД – сервер, отвечающий за хранение метаданных Системы и пользовательских данных.

В этой топологии возможны следующие варианты:

1. некластерный режим запуска Alpha BI. В этом случае:
   * есть всего один сервер приложения (N\_app = 1);
   * отсутствуют серверы Etcd (N\_etcd = 0);
   * допускается исключить сервер файлового хранилища и размещать файловое хранилище на диске сервера приложения.
2. допускается объединять компоненты из нескольких серверов в один сервер, если планируемая нагрузка на Систему позволяет это сделать.

### Конфигурационные параметры

В п. 4.1 – 4.5 делаются отсылки к именованным значениям, представленным в таблице ниже (Таблица 3). Для ссылки на приведенные значения будет использоваться нотация ${ParameterName}. Данные ссылки, приведенные в п. 4.1 – 4.5, следует заменять на значение соответствующего параметра из таблицы ниже (Таблица 3).

Таблица – Конфигурационные параметры

| Наименование | Описание | Правило определения | Пример |
| --- | --- | --- | --- |
| ExternalURL | Внешний URL. Адрес, по которому Alpha BI доступна для конечного пользователя |  | **https://some-company.ru/alphabi** |
| ExternalURLPathPrefix | Абсолютный путь внешнего URL приложения | Путь внешнего URL приложения | /alphabi  / |
| BalancerHost | Имя сервера или IP-адрес внутреннего балансировщика |  | balancer.localdomain  192.168.56.1 |
| WebAppHost\_1, …, WebAppHost\_n | Имена или IP-адреса серверов приложений |  | web\_1.localdomain  192.168.56.11 |
| FsHost | Имя или IP-адрес сервера файлового хранилища |  | samba\_fs.localdomain  192.168.56.2 |
| WebAppPort | Номер TCP-порта, который используется для web-приложения | Рекомендуемое значение: 5000 | 5000 |
| HaproxyPort | Номер TCP-порта, который используется для Haproxy | Рекомендуемое значение: 5000 | 5000 |
| EtcdHost\_1, …, EtcdHost\_n | Имена или IP-адреса серверов состояния |  | etcd\_1.localdomain  192.168.56.21 |
| EtcdPort | Номер TCP-порта, который используется для подключения клиентов Etcd | Рекомендуемое значение: 2379 | 2379 |
| EtcdPeerPort | Номер TCP-порта, который используется для организации кластера Etcd | Рекомендуемое значение: 2380 | 2380 |
| EtcdUser | Имя пользователя для доступа к Etcd | Рекомендуется использовать отдельного пользователя | alphabi |
| EtcdPassword | Пароль пользователя для доступа к Etcd | Рекомендуется сгенерировать случайный пароль |  |
| EtcdPrefix | Префикс, который используется для ключей в Etcd | Рекомендуемое значение: /alphabi | /alphabi |
| PostgresHost | Имя или IP-адрес сервера БД |  | postgres.localdomain  192.168.56.3 |
| PostgresPort | Номер TCP-порта, который используется для PostgreSQL | Рекомендуемое значение: 5432 | 5432 |
| PostgresUser | Имя пользователя для доступа к БД Alpha BI | Рекомендуется использовать отдельного пользователя | alphabi |
| PostgresPassword | Пароль от пользователя ${PostgresUser} | Рекомендуется сгенерировать случайный пароль |  |
| PostgresDatabase | Имя БД в Postgres, в которой хранятся служебные данные Alpha BI | Рекомендуется использовать отдельную БД | alphabi |
| MondrianHost | Имя или IP-адрес сервера OLAP |  | olap.localdomain  192.168.56.4 |
| MondrianPort | Номер TCP-порта, который используется для Mondrian | Рекомендуемое значение: 8080 | 8080 |
| MondrianAppName | Имя web-приложения Mondrian | Совпадает с именем war-архива без расширения | mondrian |
| MondrianApiKey | Ключ доступа, используемый для аутентификации между Mondrian и Alpha BI | Рекомендуется сгенерировать случайный ключ доступа |  |
| AlphaControlCenterPassword | Пароль от web-интерфейса Центра управления Alpha BI | Рекомендуется сгенерировать случайный пароль |  |
| AlphaAdminPassword | Пароль для пользователя admin в Alpha BI | Рекомендуется сгенерировать случайный пароль |  |
| JavaExecutable | Путь до исполняемого файла java на сервере приложения |  | java  /usr/lib/jvm/java-17-openjdk-amd64/bin/java |

### Состав дистрибутива

Дистрибутив – это zip-архив с именем Bars.Alpha-{version}.zip (где {version} – это версия Alpha BI). В этом архиве содержатся:

* основное web-приложение Alpha BI, собранное под несколько вариантов целевых архитектур:
  + Bars.Alpha.Web-linux-x86\_64-{version}.zip. Сборка для ОС Linux для процессоров с архитектурой x86-64;
  + Bars.Alpha.Web-linux-arm64-{version}.zip. Сборка для ОС Linux для процессоров с архитектурой ARM64;
  + Bars.Alpha.Web-windows-x64-{version}.zip. Сборка для ОС Windows для процессоров с архитектурой x86-64;
  + Bars.Alpha.Web-netcore-{version}.zip. Сборка, не привязанная к ОС и архитектуре процессора и требующая установки .NET 6 Runtime.
* Bars.Alpha.Mondrian-{mondrian\_version}.war. OLAP-сервер Alpha BI.

## Развертывание серверов

### Сервер БД

Примечание – Все команды выполняются под суперпользователем (root). Стать суперпользователем можно, например, с помощью команды sudo su.

#### Зависимости

##### Ntpd

Примечание – Необходимо, чтобы на сервере БД системное время было корректным.

Для настройки Ntpd выполните следующие действия:

1. установите Ntpd:

apt update

apt -y install ntp

1. отредактируйте файл **/etc/ntp.conf**: замените директивы pool на блок директив:

server 0.ru.pool.ntp.org

server 1.ru.pool.ntp.org

server 2.ru.pool.ntp.org

server 3.ru.pool.ntp.org

Сделать это можно командами (не забывая проверить корректность вставленного текста):

sed -i '/pool\ 0/c server 0.ru.pool.ntp.org' /etc/ntp.conf

sed -i '/pool\ 1/c server 1.ru.pool.ntp.org' /etc/ntp.conf

sed -i '/pool\ 2/c server 2.ru.pool.ntp.org' /etc/ntp.conf

sed -i '/pool\ 3/c server 3.ru.pool.ntp.org' /etc/ntp.conf

1. запустите Ntpd:

systemctl restart ntp

#### Установка

Для установки сервера БД выполните следующие действия:

1. подключите репозиторий с официальными пакетами PostgreSQL и импортируйте ключ:

sudo sh -c 'echo "deb http://apt.postgresql.org/pub/repos/apt $(lsb\_release -cs)-pgdg main" > /etc/apt/sources.list.d/pgdg.list'

wget --quiet -O - https://www.postgresql.org/media/keys/ACCC4CF8.asc | sudo apt-key add -

1. установите сервер PostgreSQL:

apt update

apt -y install postgresql-15

Примечание – При необходимости использования иной версии PostgreSQL следует учесть, что версия должна быть не ниже 10.

#### Конфигурирование и запуск

Для конфигурирования и запуска выполните следующие действия:

1. включите опцию max\_prepared\_transactions с рекомендуемым значением 100 в конфигурационном файле **/etc/postgresql/15/main/postgresql.conf**:

Примечание – Согласно официальной документации PostgreSQL рекомендуется указывать значение, равное максимальному количеству подключений max\_connections.

sed -e 's/#max\_prepared\_transactions.\*/max\_prepared\_transactions = 100/' -i /etc/postgresql/15/main/postgresql.conf

1. разрешите подключение к СУБД от других серверов. Для этого выполните следующие действия:
2. в файле **/etc/postgresql/15/main/postgresql.conf** измените настройку listen\_addresses:

sed -e "s/#listen\_addresses.\*/listen\_addresses = '\*'/" -i /etc/postgresql/15/main/postgresql.conf

Примечание – При необходимости можно изменить символ \* на ограниченный набор сетевых интерфейсов.

1. в файле **/etc/postgresql/15/main/pg\_hba.conf** добавьте строку:

host all all 0.0.0.0/0 scram-sha-256

Примечание – При необходимости подсеть 0.0.0.0/0 замените на ограниченную подсеть или подсети, включающие в себя сервера приложений и сервер OLAP. Вместо md5 можно использовать scram-sha-256.

1. рекомендуется внести изменения в файл настроек **/etc/postgresql/15/main/postgresql.conf** в зависимости от характеристик сервера, например, можно воспользоваться помощником по конфигурации сервера (см. ссылки [**https://www.pgconfig.org/**](https://www.pgconfig.org/) и [**https://pgtune.sainth.de/**](https://pgtune.sainth.de/));
2. перезапустите PostgreSQL:

systemctl restart postgresql

1. создайте пользователя и БД для Alpha BI:

su postgres -c "psql -c \"create user ${PostgresUser} password '${PostgresPassword}' createdb\""

su postgres -c "psql -c \"create database ${PostgresDatabase} owner ${PostgresUser}\""

Примечание – Подробнее об используемых конфигурационных параметрах описано в п. 4.1.1.

### Сервер OLAP

Есть два способа развертывания сервера OLAP, которые отличаются методом получения дистрибутива Tomcat. Первый вариант упрощенный, второй следует использовать, если по каким-либо причинам нет возможности воспользоваться первым.

#### Вариант 1. Развертывание сервера OLAP с установкой Tomcat из репозитория

Примечание – Все команды выполняются под суперпользователем (root). Стать суперпользователем можно, например, с помощью команды sudo su.

##### Ntpd

Примечание – Необходимо, чтобы на сервере OLAP системное время было корректным.

Для настройки Ntpd выполните следующие действия:

1. установите Ntpd:

apt update

apt -y install ntp

1. отредактируйте файл **/etc/ntp.conf**: замените директивы pool на блок директив:

server 0.ru.pool.ntp.org

server 1.ru.pool.ntp.org

server 2.ru.pool.ntp.org

server 3.ru.pool.ntp.org

Сделать это можно командами (не забывая проверить корректность вставленного текста):

sed -i '/pool\ 0/c server 0.ru.pool.ntp.org' /etc/ntp.conf

sed -i '/pool\ 1/c server 1.ru.pool.ntp.org' /etc/ntp.conf

sed -i '/pool\ 2/c server 2.ru.pool.ntp.org' /etc/ntp.conf

sed -i '/pool\ 3/c server 3.ru.pool.ntp.org' /etc/ntp.conf

1. запустите Ntpd:

systemctl restart ntp

##### Развертывание Tomcat

###### Зависимости Tomcat

Установите зависимости из репозитория:

apt update

apt -y install openjdk-17-jre

###### Первоначальная установка Tomcat

Установите Tomcat 9 из репозитория:

apt -y install tomcat9

###### Конфигурирование Tomcat

Для конфигурирования Tomcat выполните следующие действия:

1. внесите изменения в переменные запуска Tomcat 9:

echo 'JAVA\_OPTS="-Djava.awt.headless=true -Djava.security.egd=file:/dev/./urandom -Xms512M -Xmx1024M -server -XX:+UseParallelGC"' > /etc/default/tomcat9

echo 'JAVA\_HOME="/usr/lib/jvm/java-17-openjdk-amd64"' >> /etc/default/tomcat9

Примечание – Вместо значения -Xmx1024M подберите значение, соответствующее размеру доступной памяти сервера. В качестве начального приближения можно использовать следующее значение: max(0.7 \* RAM, RAM – 4GB), где RAM – это объем памяти сервера. Например, если на сервере 4 ГБ памяти, то установите значение -Xmx2867M, а если 64 ГБ – то значение -Xmx61440M.

1. перезапустите Tomcat и убедитесь, что сервис запустился штатно:

systemctl restart tomcat9

systemctl status tomcat9

##### Развертывание Mondrian

###### Зависимости Mondrian

Установите файловый менеджер и утилиты архивации zip, unzip:

apt -y install mc unzip zip

###### Конфигурирование Mondrian

Предполагается, что дистрибутив Alpha BI Mondrian находится в файле **/tmp/Bars.Alpha.Mondrian-${mondrian\_version}.war** (где ${mondrian\_version} – версия mondrian). С помощью файлового менеджера mc есть возможность работать с архивами war как с обычными каталогами. Пути будут указываться от корня архива.

Для конфигурирования Mondrian выполните следующие действия:

1. отредактируйте файл **WEB-INF/web.xml**:
2. измените значение параметра DataSourcesConfig:

<init-param>

<param-name>DataSourcesConfig</param-name>

<param-value>http://${BalancerHost}:${HaproxyPort}/mondrian/datasources</param-value>

</init-param>

Если в приложении настроен префикс, в таком случае значение параметра будет выглядеть следующим образом:

<init-param>

<param-name>DataSourcesConfig</param-name>

<param-value>http://${BalancerHost}:${HaproxyPort}/${ExternalURLPathPrefix}/mondrian/datasources</param-value>

</init-param>

1. измените значение параметра AlphaApiKey:

<init-param>

<param-name>AlphaApiKey</param-name>

<param-value>${MondrianApiKey}</param-value>

</init-param>

Примечание – Подробнее об используемых конфигурационных параметрах описано в п. 4.1.1.

1. отредактируйте файл **WEB-INF/classes/mondrian.properties**: в этом файле замените значение параметра mondrian.util.memoryMonitor.percentage.threshold на 99:

mondrian.util.memoryMonitor.percentage.threshold=99

###### Установка Mondrian

Для установки Mondrian выполните следующие действия:

1. переместите файл **/tmp/Bars.Alpha.Mondrian-${mondrian\_version}.war** в каталог приложений Tomcat, изменив его название на mondrian.war:

mv /tmp/Bars.Alpha.Mondrian-${mondrian\_version}.war /var/lib/tomcat9/webapps/mondrian.war

1. перезапустите Tomcat и убедитесь, что сервис запустился штатно:

systemctl restart tomcat9

systemctl status tomcat9

#### Вариант 2. Развертывание сервера OLAP с установкой Tomcat дистрибутивом из официального сайта Apache Tomcat

Примечание – Все команды выполняются под суперпользователем (root). Стать суперпользователем можно, например, с помощью команды sudo su.

##### Ntpd

Примечание – Необходимо, чтобы на сервере OLAP системное время было корректным.

Для настройки Ntpd выполните следующие действия:

1. установите Ntpd:

apt update

apt -y install ntp

1. отредактируйте файл **/etc/ntp.conf**: замените директивы pool на блок директив:

server 0.ru.pool.ntp.org

server 1.ru.pool.ntp.org

server 2.ru.pool.ntp.org

server 3.ru.pool.ntp.org

Сделать это можно командами (не забывая проверить корректность вставленного текста):

sed -i '/pool\ 0/c server 0.ru.pool.ntp.org' /etc/ntp.conf

sed -i '/pool\ 1/c server 1.ru.pool.ntp.org' /etc/ntp.conf

sed -i '/pool\ 2/c server 2.ru.pool.ntp.org' /etc/ntp.conf

sed -i '/pool\ 3/c server 3.ru.pool.ntp.org' /etc/ntp.conf

1. запустите ntpd:

systemctl restart ntp

##### Развертывание Tomcat

###### Зависимости Tomcat

Установите зависимости из репозитория:

apt update

apt -y install openjdk-17-jre

###### Первоначальная установка Tomcat

Для первоначальной установки Tomcat выполните следующие действия:

1. создайте отдельного непривилегированного пользователя для запуска Tomcat:

useradd -m --system tomcat

1. скачайте дистрибутив Tomcat 10.1:

wget -P /tmp https://dlcdn.apache.org/tomcat/tomcat-10/v10.1.12/bin/apache-tomcat-10.1.12.tar.gz

Примечание – Рекомендуется использовать актуальную версию Tomcat 10, найти которую можно на официальном сайте: **https://tomcat.apache.org/download-10.cgi**. При этом следует поменять версию («10.1.12») в последующих шагах.

1. распакуйте дистрибутив, удалите архив и назначьте права доступа:

mkdir /opt/tomcat

tar xf /tmp/apache-tomcat-10.1.12.tar.gz -C /opt/tomcat --strip-components 1

rm /tmp/apache-tomcat-10.1.12.tar.gz

chown -R tomcat /opt/tomcat

###### Конфигурирование Tomcat

Для конфигурирования Tomcat выполните следующие действия:

1. удалите приложения по умолчанию:

rm -rf /opt/tomcat/webapps/\*

1. отключите неиспользуемый порт. Для этого отредактируйте файл **/opt/tomcat/conf/server.xml**:

sed -i 's/Server port="8005"/Server port="-1"/' /opt/tomcat/conf/server.xml

1. добавьте файл с переменными:

echo "CATALINA\_BASE='/opt/tomcat'" >> /opt/tomcat/conf/environment.conf

echo "CATALINA\_OPTS='-Xms512M -Xmx1024M -server -XX:+UseParallelGC" >> /opt/tomcat/conf/environment.conf

echo "JAVA\_OPTS='-Djava.awt.headless=true -Djava.security.egd=file:/dev/./urandom'" >> /opt/tomcat/conf/environment.conf

Примечание – Вместо значения -Xmx1024M подберите значение, соответствующее размеру доступной памяти сервера. В качестве начального приближения можно использовать следующее значение: max(0.7 \* RAM, RAM – 4GB), где RAM – это объем памяти сервера. Например, если на сервере 4 ГБ памяти, то установите значение -Xmx2867M, а если 64 ГБ – то значение -Xmx61440M.

1. настройте ведение логов в системном каталоге логов:

mkdir /var/log/tomcat

chown tomcat:adm /var/log/tomcat

rm -r /opt/tomcat/logs

ln -s /var/log/tomcat /opt/tomcat/logs

###### Запуск Tomcat

Для запуска Tomcat выполните следующие действия:

1. создайте файл запуска приложения **/etc/systemd/system/tomcat.service** со следующим содержимым:

# Systemd unit file for tomcat

[Unit]

Description=Apache Tomcat Web Application Container

After=syslog.target network.target

[Service]

Type=forking

Environment=CATALINA\_PID=/opt/tomcat/temp/item.pid

Environment=CATALINA\_HOME=/opt/tomcat

Environment=CATALINA\_BASE=/opt/tomcat

EnvironmentFile=/opt/tomcat/conf/environment.conf

ExecStart=/opt/tomcat/bin/startup.sh

ExecStop=/bin/kill -15 $MAINPID

User=tomcat

UMask=0007

RestartSec=10

Restart=always

[Install]

WantedBy=multi-user.target

1. перезагрузите конфигурацию менеджера systemd:

systemctl daemon-reload

1. настройте автозапуск сервиса:

systemctl enable tomcat

##### Развертывание Mondrian

###### Зависимости Mondrian

Установите файловый менеджер и утилиты архивации zip, unzip:

apt -y install mc unzip zip

###### Конфигурирование Mondrian

Предполагается, что дистрибутив Alpha BI Mondrian находится в файле **/tmp/Bars.Alpha.Mondrian-${mondrian\_version}.war** (где ${mondrian\_version} – версия mondrian). С помощью файлового менеджера mc есть возможность работать с архивами war как с обычными каталогами. Пути будут указываться от корня архива.

Для конфигурирования Mondrian выполните следующие действия:

1. отредактируйте файл **WEB-INF/web.xml**:
2. измените значение параметра DataSourcesConfig:

<init-param>

<param-name>DataSourcesConfig</param-name>

<param-value>http://${BalancerHost}:${HaproxyPort}/mondrian/datasources</param-value>

</init-param>

Если в приложении настроен префикс, в таком случае значение параметра будет выглядеть следующим образом:

<init-param>

<param-name>DataSourcesConfig</param-name>

<param-value>http://${BalancerHost}:${HaproxyPort}/${ExternalURLPathPrefix}/mondrian/datasources</param-value>

</init-param>

1. измените значение параметра AlphaApiKey:

<init-param>

<param-name>AlphaApiKey</param-name>

<param-value>${MondrianApiKey}</param-value>

</init-param>

Примечание – Подробнее об используемых конфигурационных параметрах описано в п. 4.1.1.

1. отредактируйте файл **WEB-INF/classes/mondrian.properties**: в этом файле замените значение параметра mondrian.util.memoryMonitor.percentage.threshold на 99:

mondrian.util.memoryMonitor.percentage.threshold=99

###### Установка Mondrian

Для установки Mondrian выполните следующие действия:

1. создайте каталог для приложений:

mkdir -p /opt/tomcat/webapps-javaee

Примечание – Название каталога должно быть webapps-javaee. Это связано с переходом с Java EE на Jakarta EE.

1. переместите файл **/tmp/Bars.Alpha.Mondrian-${mondrian\_version}.war** в каталог приложений Tomcat, изменив его название на mondrian.war:

mv /tmp/Bars.Alpha.Mondrian-${mondrian\_version}.war /opt/tomcat/webapps-javaee/mondrian.war

1. запустите Tomcat и убедитесь, что сервис запустился штатно:

systemctl start tomcat

systemctl status tomcat

### Сервер файлового хранилища

Примечание – Все команды выполняются под суперпользователем (root). Стать суперпользователем можно, например, с помощью команды sudo su.

#### Первоначальная установка

Установите сервер NFS:

apt update

apt -y install nfs-kernel-server

#### Конфигурирование

Для конфигурирования выполните следующие действия:

1. создайте каталог для хранения файлов и назначьте его владельцем пользователя nobody:

mkdir /var/lib/alphabi\_file\_storage

chown nobody:nogroup /var/lib/alphabi\_file\_storage/

1. зарегистрируйте каталог с файлами в NFS-сервере:

echo "/var/lib/alphabi\_file\_storage \*(rw,async,all\_squash,anonuid=65534,anongid=65534,no\_subtree\_check)" >> /etc/exports

exportfs -ra

Примечание – Символ \* можно заменить на указание подсети, содержащей серверы приложений (в виде x.y.z.w/n).

Примечание – Ниже описано, как настроить анонимный NFS-сервер. Рекомендуется настраивать только в защищенной сети.

#### Запуск

Перезапустите NFS-сервер:

systemctl restart nfs-kernel-server

### Серверы состояния

Примечание – Все команды выполняются под суперпользователем (root). Стать суперпользователем можно, например, с помощью команды sudo su.

#### Установка зависимостей

##### Ntpd

Примечание – Необходимо, чтобы на серверах состояния системное время было корректным.

Для настройки Ntpd выполните следующие действия:

1. установите Ntpd на все узлы кластера etcd:

apt update

apt –y install ntp

1. отредактируйте файл **/etc/ntp.conf**: замените директивы pool на блок директив:

server 0.ru.pool.ntp.org

server 1.ru.pool.ntp.org

server 2.ru.pool.ntp.org

server 3.ru.pool.ntp.org

Сделать это можно командами:

sed –i '/pool\ 0/c server 0.ru.pool.ntp.org' /etc/ntp.conf

sed –i ‘/pool\ 1/c server 1.ru.pool.ntp.org’ /etc/ntp.conf

sed –i ‘/pool\ 2/c server 2.ru.pool.ntp.org’ /etc/ntp.conf

sed –i ‘/pool\ 3/c server 3.ru.pool.ntp.org’ /etc/ntp.conf

1. запустите Ntpd:

systemctl restart ntp

#### Первоначальная установка

Для первоначальной установки выполните следующие действия:

1. установите etcd на все узлы кластера:

apt -y install etcd

1. остановите etcd и удалите данные, генерируемые при установке на каждом узле кластера:

systemctl stop etcd

rm -rf /var/lib/etcd/default/member

#### Сборка и запуск кластера Etcd

Кластер из узлов Etcd следует формировать последовательно. Для этого по порядку настройте и запустите каждый узел, выполнив шаги, описанные ниже.

##### Конфигурирование первого узла кластера Etcd

Для конфигурирования первого узла кластера Etcd выполните следующие действия:

1. отредактируйте файл **/etc/default/etcd**, указав следующие значения:

ETCD\_NAME="etcd\_alphabi\_1"

ETCD\_LISTEN\_CLIENT\_URLS="http://${EtcdHost\_1}:${EtcdPort},http://localhost:${EtcdPort}"

ETCD\_ADVERTISE\_CLIENT\_URLS="http://${EtcdHost\_1}:${EtcdPort}"

ETCD\_LISTEN\_PEER\_URLS="http://${EtcdHost\_1}:${EtcdPeerPort}"

ETCD\_INITIAL\_ADVERTISE\_PEER\_URLS="http://${EtcdHost\_1}:${EtcdPeerPort}"

ETCD\_INITIAL\_CLUSTER\_TOKEN="alphabi"

ETCD\_INITIAL\_CLUSTER="etcd\_alphabi\_1=http://${EtcdHost\_1}:${EtcdPeerPort}"

ETCD\_INITIAL\_CLUSTER\_STATE="new"

ETCD\_AUTO\_COMPACTION\_RETENTION="24"

* где ETCD\_AUTO\_COMPACTION\_RETENTION – параметр, отвечающий за автоматическое сжатие пространства ключей.

Примечание – Подробнее об используемых конфигурационных параметрах описано в п. 4.1.1.

1. запустите Etcd на первом узле:

systemctl start etcd

##### Конфигурирование последующих узлов кластера Etcd

Предполагается, что в данный момент конфигурируется узел ${Current} – номер текущего узла кластера Etcd, у которого имя сервера – ${EtcdHost\_Current}.

Для конфигурирования второго и следующих узлов кластера Etcd выполните следующие действия:

1. зарегистрируйте новый узел кластера:

ETCDCTL\_API=3 etcdctl --endpoints ${EtcdHost\_1}:${EtcdPort} member add etcd\_alphabi\_${Current} --peer-urls http://${EtcdHost\_Current}:${EtcdPeerPort}

Примечание – Подробнее об используемых конфигурационных параметрах описано в п. 4.1.1.

При этом в консоль будут записаны строки со значениями:

ETCD\_NAME

ETCD\_INITIAL\_CLUSTER

ETCD\_INITIAL\_CLUSTER\_STATE

1. отредактируйте файл **/etc/default/etcd**, указав следующие значения:

ETCD\_NAME="etcd\_alphabi\_${Current}"

ETCD\_LISTEN\_CLIENT\_URLS="http://${EtcdHost\_Current}:${EtcdPort},http://localhost:${EtcdPort}"

ETCD\_ADVERTISE\_CLIENT\_URLS="http://${EtcdHost\_Current}:${EtcdPort}"

ETCD\_LISTEN\_PEER\_URLS="http://${EtcdHost\_Current}:${EtcdPeerPort}"

ETCD\_INITIAL\_ADVERTISE\_PEER\_URLS="http://${EtcdHost\_Current}:${EtcdPeerPort}"

ETCD\_INITIAL\_CLUSTER\_TOKEN="alphabi"

ETCD\_INITIAL\_CLUSTER="${ETCD\_INITIAL\_CLUSTER}"

ETCD\_INITIAL\_CLUSTER\_STATE="existing"

ETCD\_AUTO\_COMPACTION\_RETENTION="24"

где:

* ${ETCD\_INITIAL\_CLUSTER} – это значение ETCD\_INITIAL\_CLUSTER, выведенное командой etcdctl member add с прошлого шага;
* ETCD\_AUTO\_COMPACTION\_RETENTION – параметр, отвечающий за автоматическое сжатие пространства ключей.

Примечание – Подробнее об используемых конфигурационных параметрах описано в п. 4.1.1.

1. запустите Etcd на текущем узле:

systemctl start etcd

#### Конфигурирование кластера Etcd

Данные команды следует выполнить на одном из узлов кластера Etcd:

1. создайте пользователя root:

ETCDCTL\_API=3 etcdctl user add root:123

Примечание – Вместо значения 123 введите значение нового пароля.

1. создайте пользователя в Etcd и назначьте этому пользователю права на диапазон ключей, выделенный для Alpha BI:

ETCDCTL\_API=3 etcdctl role add ${EtcdUser}

ETCDCTL\_API=3 etcdctl role grant-permission ${EtcdUser} --prefix=true readwrite ${EtcdPrefix}

ETCDCTL\_API=3 etcdctl user add ${EtcdUser}:${EtcdPassword}

ETCDCTL\_API=3 etcdctl user grant-role ${EtcdUser} ${EtcdUser}

Примечание – Подробнее об используемых конфигурационных параметрах описано в п. 4.1.1.

1. включите аутентификацию:

ETCDCTL\_API=3 etcdctl auth enable

### Cервер балансировки

Примечание – Все команды выполняются под суперпользователем (root). Стать суперпользователем можно, например, с помощью команды sudo su.

#### Зависимости

##### Ntpd

Примечание – Необходимо, чтобы на сервере балансировки системное время было корректным.

Для настройки Ntpd выполните следующие действия:

1. установите Ntpd:

apt update

apt -y install ntp

1. отредактируйте файл **/etc/ntp.conf**: замените директивы pool на блок директив:

server 0.ru.pool.ntp.org

server 1.ru.pool.ntp.org

server 2.ru.pool.ntp.org

server 3.ru.pool.ntp.org

Сделать это можно командами (не забывая проверить корректность вставленного текста):

sed -i '/pool\ 0/c server 0.ru.pool.ntp.org' /etc/ntp.conf

sed -i '/pool\ 1/c server 1.ru.pool.ntp.org' /etc/ntp.conf

sed -i '/pool\ 2/c server 2.ru.pool.ntp.org' /etc/ntp.conf

sed -i '/pool\ 3/c server 3.ru.pool.ntp.org' /etc/ntp.conf

1. запустите ntpd:

systemctl restart ntp

#### Установка

Для установки выполните следующие действия:

1. установите nginx и haproxy:

apt -y install nginx haproxy

1. удалите файлы default в каталогах **/etc/nginx/sites-available/** и **/etc/nginx/sites-enabled/**:

rm /etc/nginx/sites-available/default /etc/nginx/sites-enabled/default

#### Конфигурирование и запуск

Для конфигурирования и запуска выполните следующие действия:

1. настройте nginx:
2. создайте файл **/etc/nginx/sites-available/alphabi.conf** со следующим содержимым:

server {

listen 80 default\_server;

location / {

return 301 https://$host$request\_uri;

}

}

server {

listen 443 ssl http2;

ssl\_certificate ${PathToCertFile};

ssl\_certificate\_key ${PathToCertKeyFile};

ssl\_session\_timeout 1d;

ssl\_session\_cache shared:MozSSL:10m;

ssl\_session\_tickets off;

# modern configuration

ssl\_protocols TLSv1.2 TLSv1.3;

ssl\_prefer\_server\_ciphers off;

add\_header Strict-Transport-Security "max-age=63072000" always;

# OCSP stapling

ssl\_stapling on;

ssl\_stapling\_verify on;

# replace with the IP address of your resolver

resolver 127.0.0.1;

location / {

proxy\_pass http://127.0.0.1:${HaproxyPort};

include proxy\_params;

}

}

где:

* ${PathToCertFile} и ${PathToCertKeyFile} – пути до файлов сертификата (сам сертификат и ключ к нему);
* ${HaproxyPort} – порт, настроенный в haproxy в разделе frontend.

1. создайте или отредактируйте файл **/etc/nginx/proxy\_params**, приведя его содержимое к следующему виду:

proxy\_redirect off;

proxy\_set\_header Host $http\_host;

proxy\_set\_header X-Forwarded-Proto $scheme;

proxy\_set\_header X-Real-IP $remote\_addr;

proxy\_set\_header X-Forwarded-For $proxy\_add\_x\_forwarded\_for;

proxy\_set\_header Connection "";

proxy\_http\_version 1.1;

proxy\_hide\_header x-aspnet-version;

proxy\_hide\_header x-powered-by;

proxy\_hide\_header x-aspnetmvc-version;

client\_max\_body\_size 512m;

client\_body\_buffer\_size 8m;

proxy\_connect\_timeout 120;

proxy\_send\_timeout 1800;

proxy\_read\_timeout 1800;

proxy\_buffer\_size 1024k;

proxy\_buffers 32 1024k;

proxy\_temp\_file\_write\_size 1024k;

proxy\_intercept\_errors on;

1. создайте символьную ссылку:

ln -s /etc/nginx/sites-available/alphabi.conf /etc/nginx/sites-enabled/alphabi.conf

1. проверьте конфигурацию nginx на наличие ошибок и перезапустите сервис:

nginx -t

systemctl restart nginx

1. настройте haproxy:
2. конфигурационный файл **/etc/haproxy/haproxy.cfg** приведите к следующему виду:

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

maxconn 20000

daemon

#Default SSL material locations

ca-base /etc/ssl/certs

crt-base /etc/ssl/private

#Default ciphers to use on SSL-enabled listening sockets.

#For more information, see ciphers(1SSL). This list is from:

# See: https://ssl-config.mozilla.org/#server=haproxy&server-version=2.0.3&config=intermediate

ssl-default-bind-ciphers ECDHE-ECDSA-AES128-GCM-SHA256:ECDHE-RSA-AES128-GCM-SHA256:ECDHE-ECDSA-AES256-GCM-SHA384:ECDHE-RSA-AES256-GCM-SHA384:ECDHE-ECDSA-CHACHA20-POLY1305:ECDHE-RSA-CHACHA20-POLY1305:DHE-RSA-AES128-GCM-SHA256:DHE-RSA-AES256-GCM-SHA384

ssl-default-bind-ciphersuites TLS\_AES\_128\_GCM\_SHA256:TLS\_AES\_256\_GCM\_SHA384:TLS\_CHACHA20\_POLY1305\_SHA256

ssl-default-bind-options ssl-min-ver TLSv1.2 no-tls-tickets

defaults

log global

mode http

timeout connect 60s

timeout client 30m

timeout server 30m

timeout http-request 5m

timeout queue 5m

timeout http-keep-alive 10s

timeout check 30s

errorfile 400 /etc/haproxy/errors/400.http

errorfile 403 /etc/haproxy/errors/403.http

errorfile 408 /etc/haproxy/errors/408.http

errorfile 500 /etc/haproxy/errors/500.http

errorfile 502 /etc/haproxy/errors/502.http

errorfile 503 /etc/haproxy/errors/503.http

errorfile 504 /etc/haproxy/errors/504.http

compression algo gzip

compression type text/html text/plain text/xml text/css application/javascript application/json

option forwardfor

option http-pretend-keepalive

frontend FE-AlphaBI-http

bind ${BalancerHost}:${HaproxyPort}

mode http

option httplog

option dontlognull

option log-separate-errors

option forwardfor header X-Real-IP

backlog 4096

maxconn 20000

default\_backend BE-BI-ALPHA

frontend FE-Mondrian-http

bind ${BalancerHost}:${MondrianPort}

mode http

option httplog

option dontlognull

option log-separate-errors

option forwardfor header X-Real-IP

backlog 4096

maxconn 20000

default\_backend BE-MONDRIAN

backend BE-BI-ALPHA

mode http

balance leastconn

cookie NodeID insert indirect httponly nocache

log global

option log-health-checks

fullconn 20000

http-check disable-on-404

option httpchk HEAD /cc/

server alpha-bi-01 ${WebAppHost\_1}:${WebAppPort} check inter 5s weight 32 maxconn 1000 cookie alphabi-01

server alpha-bi-N ${WebAppHost\_N}:${WebAppPort} check inter 5s weight 32 maxconn 1000 cookie alphabi-N

backend BE-MONDRIAN

mode http

balance leastconn

cookie NodeID insert indirect nocache

log global

option log-health-checks

fullconn 20000

http-check disable-on-404

option httpchk HEAD /${MondrianAppName}

server olap-01 ${ModrianHost}:${MondrianPort} check inter 5s weight 32 maxconn 1000 cookie olap-01

где:

* ${WebAppPort} – порт, на котором работает Alpha BI;
* ${WebAppHost\_1} – адрес сервера, на котором развернута Alpha BI;
* вместо строки «server alpha-bi-N ${WebAppHost\_N}:${WebAppPort} check inter 5s weight 32 maxconn 1000 cookie alphabi-N» добавьте нужное количество строк по количеству серверов с развернутым приложением Alpha BI, вместо N укажите порядковый номер.

1. проверьте конфигурацию haproxy и перезапустите сервис:

haproxy -f /etc/haproxy/haproxy.cfg -c

systemctl restart haproxy

### Сервер приложения

Примечание – Все команды выполняются под суперпользователем (root). Стать суперпользователем можно, например, с помощью команды sudo su.

#### Установка зависимостей

##### Ntpd

Примечание – Необходимо, чтобы на сервере приложения системное время было корректным.

Для настройки Ntpd выполните следующие действия:

1. установите Ntpd:

apt update

apt -y install ntp

1. отредактируйте файл **/etc/ntp.conf**: замените директивы pool на блок директив:

server 0.ru.pool.ntp.org

server 1.ru.pool.ntp.org

server 2.ru.pool.ntp.org

server 3.ru.pool.ntp.org

Сделать это можно командами:

sed -i '/pool\ 0/c server 0.ru.pool.ntp.org' /etc/ntp.conf

sed -i '/pool\ 1/c server 1.ru.pool.ntp.org' /etc/ntp.conf

sed -i '/pool\ 2/c server 2.ru.pool.ntp.org' /etc/ntp.conf

sed -i '/pool\ 3/c server 3.ru.pool.ntp.org' /etc/ntp.conf

1. запустите Ntpd:

systemctl restart ntp

##### NFS

Установите пакеты для работы с NFS:

apt -y install nfs-common

##### Зависимости для .NET

Установите системные библиотеки, необходимые для работы .NET 6:

* krb5;
* ICU;
* OpenSSL;
* zlib;
* gdip.

Для этого выполните команду:

apt -y install libgssapi-krb5-2 libicu70 libssl3 zlib1g libgdiplus

Примечание – В зависимости от версии дистрибутива замените libicu70 на одно из следующих значений:

* Debian 10: libicu63;
* Debian 11: libicu67;
* Ubuntu 20.x: libicu66;
* Ubuntu 22.x: libicu70.

Версия пакета libssl также может отличаться в разных дистрибутивах:

* Ubuntu 20.x: libssl1.1.

##### Зависимости серверного генератора отчетов

Установите Java Runtime Environment версии 17 или выше.

apt -y install openjdk-17-jre

##### Прочие зависимости

Установите распаковщик архивов:

apt -y install unzip

#### Первоначальная установка

Для первоначальной установки Alpha BI выполните следующие действия:

1. скопируйте дистрибутив Alpha BI на сервер. Для определенности предполагается, что дистрибутив сохранен в файл **/tmp/dist/Bars.Alpha.Web-linux-x86\_64-${version}.zip**;

Примечание – В зависимости от ОС и архитектуры процессора следует использовать подходящий дистрибутив (см. п. 4.1.2).

1. создайте непривилегированного пользователя для запуска Alpha BI:

useradd -m --system alphabi

1. создайте каталоги для web-приложения и логов, распакуйте дистрибутив:

mkdir -p /opt/alphabi /var/log/alphabi

unzip /tmp/Bars.Alpha.Web-linux-x86\_64-${version}.zip -d /opt/alphabi

chown -R alphabi /opt/alphabi /var/log/alphabi

Примечание – Значение Bars.Alpha.Web-linux-x86\_64-${version}.zip замените на настоящее имя файла дистрибутива.

В результате должна получиться примерно следующая структура каталогов в файловой системе сервера:

$ ls -lh /opt/alphabi

итого 40K

-rw-r--r--. 1 alphabi root 4.8K апр 29 11:45 alpha.release.config

drwxr-xr-x. 4 alphabi root 20K май 22 15:07 bin

-rw-r--r--. 1 alphabi root 130 май 16 14:12 info.txt

1. создайте каталог для размещения временных файлов в процессе работы приложения:

mkdir -p /var/tmp/alphabi

chown -R alphabi /var/tmp/alphabi

Примечание – Не рекомендуется размещать временный каталог в tmpfs, т.к. размеры временных файлов могут быть большими.

#### Конфигурирование

##### Настройка хранения файлов при кластерном режиме

Примонтируйте файловое хранилище:

mkdir -p /mnt/nfs/alphabi\_file\_storage

echo '${FsHost}:/var/lib/alphabi\_file\_storage /mnt/nfs/alphabi\_file\_storage/ nfs rw,sync,intr 0 0' >> /etc/fstab

mount –a

Примечание – Замените значение ${FsHost} на адрес NFS-сервера.

##### Настройка хранения файлов при некластерном режиме

Создайте каталог для хранения файлов и назначьте права учетной записи Alpha BI:

mkdir -p /var/lib/alphabi\_file\_storage/

chown alphabi:alphabi /var/lib/alphabi\_file\_storage

##### Настройка скрипта запуска Alpha BI

Создайте файл **/etc/systemd/system/alphabi.service** следующего содержания:

[Unit]

Description=AlphaBI: Web application

After=syslog.target network.target

[Service]

User=alphabi

Environment=ASPNETCORE\_ENVIRONMENT=Production

Environment=ASPNETCORE\_URLS=http://\*:${WebAppPort}/

Environment=ASPNETCORE\_URL\_BASE=${ExternalURLPathPrefix}

Environment=ASPNETCORE\_USE\_XFORWARDEDFOR=true

WorkingDirectory=/opt/alphabi

ExecStart=/opt/alphabi/bin/Bars.Alpha.Web

RestartSec=10

Restart=always

[Install]

WantedBy=multi-user.target

Примечание – Подробнее об используемых конфигурационных параметрах описано в п. 4.1.1.

##### Создание конфигурационного файла

Для создания конфигурационного файла выполните следующие действия:

1. создайте файл **/opt/alphabi/alpha.config** следующего содержания: (alpha.config строится на основе alpha.release.config; alpha.release.config содержит комментарии):

{

"ControlCenterConfig": {

"Password": "${AlphaControlCenterPassword}"

},

"DatabaseConfig": {

"ServerAddress": "${PostgresHost}",

"ServerPort": ${PostgresPort},

"User": "${PostgresUser}",

"Password": "${PostgresPassword}",

"AdditionalParams": "",

"DatabaseName": "${PostgresDatabase}",

"UserCommandTimeoutSeconds": 30

},

"Cluster": {

"IsClustered": true,

"Etcd": {

"Host": "${EtcdHost\_1}",

"Port": ${EtcdPort},

"UserName": "${EtcdUser}",

"Password": "${EtcdPassword}",

"KeyPrefix": "${EtcdPrefix}"

}

},

"OlapConfig": {

"XmlaEndpoint": "http://${BalancerHost}:${MondrianPort}/${MondrianAppName}/xmla",

"ApplicationBaseUrl": "http://${BalancerHost}:${WebAppPort}${ExternalURLPathPrefix}",

"ApiKey": "${MondrianApiKey}"

},

"FileStorageConfig": {

"RootPath": "/mnt/nfs/alphabi\_file\_storage"

},

"LogConfig": {

"FileLogging": "info",

"DbLogging": "info",

// Путь до директории с логами. По умолчанию директория .logs в рабочей директории приложения

"LogDirectory": "/var/log/alphabi"

},

"TempDir": {

"Path": "/var/tmp/alphabi",

"Clear": true

},

"ApplicationInfo": {

"PublicUrl": "${ExternalURL}"

},

"ReportGenerator": {

"JavaExecutable": "${JavaExecutable}"

},

}

Примечание – Для некластерного режима используйте другие значения для ключа Cluster:

"Cluster": {

"IsClustered": false,

"Etcd": {

"Host": "",

"Port": 2379,

"UserName": "",

"Password": "",

"KeyPrefix": ""

}

},

При некластерном режиме в значении ключа FileStorageConfig.RootPath укажите значение /var/lib/alphabi\_file\_storage.

Подробнее об используемых конфигурационных параметрах описано в п. 4.1.1.

1. назначьте необходимые права на конфигурационный файл:

chown alphabi /opt/alphabi/alpha.config

#### Проведение миграций

Проведите первичные миграции из каталога с приложением под пользователем Alpha BI в случае нулевого развертывания с пустой базой:

su - alphabi -s /bin/bash

cd /opt/alphabi

./bin/Bars.Alpha.Web install ${Password}

Примечание – Запускайте из директории приложения, вместо ${Password} укажите пароль от базового пользователя admin. Запускается только один раз на одном из серверов приложения, если таковых несколько.

#### Запуск

Для запуска выполните следующие действия:

1. запустите web-приложение Alpha BI и включите автоматический запуск при загрузке:

systemctl start alphabi

systemctl enable alphabi

1. установите лицензию (см. п. 5.1.1).

## Обновление Системы

Для обновления Alpha BI на новую версию выполните следующие действия:

1. обновите версию приложения (см. п. 4.3.1);
2. обновите Mondrian (см. п. 4.3.2);
3. проведите миграции (см. п. 4.3.3).

### Обновление версии приложения

Примечание – Все команды выполняются под суперпользователем (root). Стать суперпользователем можно, например, с помощью команды sudo su.

На каждом сервере приложения выполните следующие действия:

1. скопируйте дистрибутив Alpha BI на сервер. Для определенности предполагается, что дистрибутив сохранен в файл **/tmp/Bars.Alpha.Web-linux-x86\_64-${version}.zip;**
2. остановите приложение Alpha BI:

systemctl stop alphabi

1. создайте резервную копию всех файлов Alpha BI:

cp -Rf /opt/alphabi/ /opt/alphabi-backup-$(date +'%Y%m%d%H%M%S')

Примечание – Резервную копию можно будет удалить после завершения обновления.

1. удалите файлы, подлежащие обновлению:

find /opt/alphabi/ -mindepth 1 -not \( -path '\*.logs\*' \) -not \( -path '\*.license\*' \) -not \( -name alpha.config -o -name cluster-node-info.json \) –delete

Примечание – В каталоге Alpha BI обновлению не подлежат следующие файлы и каталоги:

* alpha.config – конфигурационный файл приложения;
* cluster-node-info.json – файл с информацией о членстве узла в кластере для кластерного режима запуска;
* .license – каталог с лицензиями;
* .logs – каталог с логами приложения.

1. распакуйте дистрибутив:

unzip /tmp/Bars.Alpha.Web-linux-x86\_64-${version}.zip -d /opt/alphabi/

chown -R alphabi /opt/alphabi

Примечание – Значение Bars.Alpha.Web-linux-x86\_64-${version}.zip замените на настоящее имя файла дистрибутива.

### Обновление Mondrian

Примечание – Все команды выполняются под суперпользователем (root). Стать суперпользователем можно, например, с помощью команды sudo su.

В зависимости от выбранного способа установки существует 2 варианта обновления Mondrian:

1. если использовался вариант установки Tomcat из репозитория;
2. если использовался вариант установки Tomcat вручную дистрибутивом с официального сайта.

#### Вариант 1 (если использовался вариант установки Tomcat из репозитория)

Для обновления Mondrian на сервере OLAP выполните следующие действия:

1. остановите Tomcat:

systemctl stop tomcat9

1. удалите имеющийся экземпляр Mondrian:

rm /var/lib/tomcat9/webapps/mondrian.war

1. повторно выполните действия, описанные в п. 4.2.2.1.3.2, 4.2.2.1.3.3.

#### Вариант 2 (если использовался вариант установки Tomcat вручную дистрибутивом с официального сайта)

Для обновления Mondrian на сервере OLAP выполните следующие действия:

1. остановите Tomcat:

systemctl stop tomcat

1. удалите имеющийся экземпляр Mondrian:

rm /opt/tomcat/webapps-javaee/mondrian.war

1. повторно выполните действия, описанные в п. 4.2.2.2.3.2, 4.2.2.2.3.3.

### Проведение миграций

Примечание – Все команды выполняются под суперпользователем (root). Стать суперпользователем можно, например, с помощью команды sudo su.

Для проведения миграций выполните следующие действия:

1. на одном из серверов приложения выполните следующие команды:

su - alphabi -s /bin/bash

cd /opt/alphabi

bin/Bars.Alpha.Web update

1. запустите Alpha BI на всех серверах приложения:

systemctl start alphabi

## Конфигурирование Alpha BI

Alpha BI имеет три возможных источника конфигурации. Возможные источники параметров конфигурации перечислены в порядке применения:

* централизованное хранилище конфигурации в Etcd (применяется только в кластерном режиме);
* файл конфигурации alpha.config в каталоге приложения Alpha BI;
* переменные окружения ALPHABI\_CONFIG\_ при запуске приложения Alpha BI.

Значения из каждого следующего источника имеют больший приоритет по сравнению с предыдущим.

Примечание – Исключение составляют некоторые параметры конфигурации, для которых в кластерном режиме единственным источником является централизованное хранилище конфигурации в Etcd. Наличие такого параметра в других источниках конфигурации для кластерного режима будет игнорироваться. Список параметров-исключений приведен в п. 4.4.2.

### Параметры конфигурации Alpha BI

Alpha BI имеет иерархическую структуру конфигурации:

* конфигурация делится на секции;
* секция может содержать другие секции или конфигурационные параметры.

Список конфигурационных параметров и их значений по умолчанию:

* «Localization=Ru» – имя набора настроек форматов, используемого для форматирования чисел и дат;
* секция «ApplicationInfo»:
  + «PublicUrl» – URL приложения, под которым приложение доступно конечным пользователям.
* секция «AppSettings» – устаревшая секция, не рекомендуется использовать:
  + «authProvider=ldap» – включение аутентификации по LDAP.

Примечание – При использовании параметра «authProvider=ldap» произойдет полное переключение на аутентификацию по LDAP, без возможности использования локальных паролей в Alpha BI, в том числе для пользователя admin. Соединение c LDAP-сервером должно быть настроено перед включением данной настройки.

* секция «Debug» – отладочные параметры:
  + секция «Saml» – параметры отладки для «Saml»:
* «DisableAssertionSignatureValidation=false» – отключение проверки подписи «Saml Assertion».
* секция «Quartz» – настройки планировщика задач:
  + «EnableQuartzScheduler=true» – включение планировщика задач на текущем сервере.
* секция «Cluster» – настройки работы в кластере:
  + «IsClustered=false» – включение работы в кластере;
  + секция «Etcd» – настройки подключения к Etcd:
* «Host» – имя сервера, на котором расположен Etcd;
* «Port=2379» – TCP-порт, на котором расположен Etcd;
* «UserName» – имя пользователя в Etcd;
* «Password» – пароль пользователя в Etcd;
* «KeyPrefix» – префикс для ключей в Etcd.
* секция «ControlCenterConfig» – настройки центра управления:
  + «Password» – пароль для доступа к центру управления.
* секция «DatabaseConfig» – настройки для подключения к системной БД:
  + «ServerAddress» – имя или адрес сервера;
  + «ServerPort=5432» – порт, на котором расположена СУБД;
  + «User» – имя пользователя БД;
  + «Password» – пароль пользователя БД;
  + «DatabaseName» – имя БД;
  + «TcpKeepAlive=true» – включение применения параметра «TcpKeepAlive»;
  + «TcpKeepAliveTimeSeconds=15» – время перед первоначальной посылкой Keep-Alive и между последующими посылками Keep-Alive;
  + «ConnectionIdleLifetimeSeconds=60» – сохраненные соединения: через какое количество секунд неактивности закрывать соединение;
  + «ConnectionLifetimeSeconds=600» – сохраненные соединения: максимальное время переиспользования одного соединения;
  + «SystemCommandTimeoutSeconds=30» – таймаут для запросов к системным данным;
  + «UserCommandTimeoutSeconds=30» – таймаут для запросов к пользовательским данным, находящимся в системной БД;
  + «AdditionalParams».
* секция «FileStorageConfig» – настройки файлового хранилища:
  + «RootPath» – путь в файловой системе к файловому хранилищу.
* секция «LogConfig» – настройки логирования:
  + «FileLogging=info» – уровень логирования для записи логов в файл. Значения: off, fatal, error, warn, info, debug, trace;
  + «DbLogging=info» – уровень логирования для записи логов в БД. Значения: off, fatal, error, warn, info, debug, trace;
  + «LogDirectory» – путь к каталогу лог-файлов.
* секция «NetworkSettings» – настройки прокси-сервера для HTTP:
  + «ProxyAddress» – адрес HTTP-прокси-сервера;
  + «ProxyPort=80» – порт HTTP-прокси-сервера;
  + «ProxyUser» – имя пользователя;
  + «ProxyPassword» – пароль;
  + «RequestTimeoutSeconds=1800» – таймаут запросов;
  + «ExclusionAddresses» – адреса, для которых не применять прокси-сервер. Список, разделенный символом «;» или «,».
* секция «OlapConfig» – настройки подключения к Mondrian:
  + «XmlaEndpoint» – URL для подключения к Mondrian;
  + «ApplicationBaseUrl» – URL приложения Alpha BI, по которому оно доступно для Mondrian;
  + «ApiKey» – ключ аутентификации между Alpha BI и Mondrian;
  + «XmlaHttpTimeoutSeconds=1800» – таймаут для XMLA-запросов к Mondrian;
  + «ControlHttpTimeoutSeconds=180» – таймаут управляющих и диагностических запросов к Mondrian.
* секция «ResponseCompression» – настройки сжатия HTTP ответов:
  + «Enabled=false» – сжатие включено;
  + «EnabledForHttps=false» – сжатие включено для HTTPS;
  + «MimeTypes=[text/xml]» – типы контента, к которым применяется сжатие. Содержит массив значений;
  + секция «GzipCompression»:
* «Enabled=true»;
* «CompressionLevel=Optimal».
* секция «SecurityConfig» – настройки безопасности:
  + «SessionTimeout=180» – срок действия аутентификации пользователя при отсутствии активности в минутах;
  + «AuthenticationCookiePath» – путь (в URL) для аутентификационных cookie в сценариях применения кросс-аутентификации в специфических сценариях развертывания. Не рекомендуется изменять значение этого параметра. Если значение не задано, то значение формируется на основе параметра «ApplicationInfoPublicUrl».
* секция «ReportGenerator» – настройки серверного формирования отчетов:
  + «JavaExecutable=java» – путь к исполняемому файлу JVM;
  + «JvmAdditionalArguments» – содержит массив значений. Дополнительные аргументы запуска Java;
  + «JavaMaxMemorySizeInMB=1024» – максимальный размер памяти для Java в мегабайтах;
  + «DefaultReportResultLifetime=1» – срок хранения сформированных отчетов в днях по умолчанию («-1» – бессрочно).
* секция «TempDir»:
  + «Path» – путь к временному каталогу. Если значение пустое, то используется системный временный каталог ($TMPDIR, /tmp);
  + «Clear» – очистка временного каталога при запуске.

Примечание – Очистка временного каталога производится, только если установлено значение «Path».

### Конфигурация Alpha BI в кластерном режиме

#### Кластерные параметры конфигурации

При работе в кластерном режиме приложение Alpha BI хранит конфигурацию в централизованном хранилище Etcd (с возможностью хранения части конфигурации локально).

Каждому параметру конфигурации соответствует один ключ в Etcd. Имя ключа формируется следующим образом: ${EtcdPrefix}/AlphaConfig/{Key}, где ${EtcdPrefix} – это общий префикс пространства ключей в Etcd (задается параметром конфигурации Cluster.Etcd.KeyPrefix), где Key – название конфигурационного параметра (в качестве разделителя составного имени используется символ «/»).

Пример – Пример настройки имени ключа: /alphabi/AlphaConfig/ApplicationInfo/PublicUrl.

Параметры конфигурации, для которых в кластерном режиме единственным источником является централизованное хранилище конфигурации в Etcd:

* «Localization=Ru» – имя набора настроек форматов, используемого для форматирования чисел и дат;
* секция «ApplicationInfo»:
  + «ShortName» – краткое наименование приложения;
  + «PublicUrl» – URL приложения, под которым приложение доступно конечным пользователям.
* секция «ControlCenterConfig» – настройки центра управления:
  + «Password» – пароль для доступа к центру управления.
* секция «ResponseCompression» – настройки сжатия HTTP-ответов:
  + «Enabled=false» – сжатие включено;
  + «EnabledForHttps=false» – сжатие включено для HTTPS;
  + «MimeTypes=[text/xml]» – типы контента, к которым применяется сжатие. Содержит массив значений;
  + секция «GzipCompression»:
* «Enabled=true»;
* «CompressionLevel=Optimal».
* секция «OlapConfig» – настройки подключения к Mondrian:
  + «ApplicationBaseUrl» – URL приложения Alpha BI, по которому оно доступно для Mondrian;
  + «ApiKey» – ключ аутентификации между Alpha BI и Mondrian.
* секция «SecurityConfig» – настройки безопасности:
  + «GuestUserId» – идентификатор в БД гостевого пользователя;
  + «SessionTimeout=180» – срок действия аутентификации пользователя при отсутствии активности в минутах.
* секция «ReportGenerator» – настройки серверного формирования отчетов:
  + «DefaultReportResultLifetime=1» – срок хранения сформированных отчетов в днях по умолчанию («-1» – бессрочно).

Наличие параметра из приведенного выше списка в локальных источниках конфигурации для кластерного режима будет игнорироваться. Остальные параметры применяются в соответствии с приоритетом источника.

#### Чтение и запись конфигурации через консольное приложение

Для работы с конфигурацией предоставлены следующие консольные субкоманды у приложения Bars.Alpha.Web. Все эти команды работают только с конфигурацией, хранимой в централизованном хранилище Etcd:

* «list-clustered-config» – выводит значения всех хранимых параметров. Значения выводятся построчно в формате «Ключ=Значение»;

Пример – Пример значений выполнения команды «list-clustered-config»:

cd /opt/alphabi/

bin/Bars.Alpha.Web list-clustered-config

ControlCenterConfig.Password=123

OlapConfig.ApiKey=1234

OlapConfig.ApplicationBaseUrl=http://localhost:5000/

OlapConfig.XmlaEndpoint=http://localhost:9999/mondrian/xmla

* «get-clustered-config key» – выводит значение одного параметра. Выводится только значение параметра;

Пример – Пример значений выполнения команды «get-clustered-config key»:

cd /opt/alphabi/

bin/Bars.Alpha.Web get-clustered-config OlapConfig.ApiKey

1234

* «set-clustered-config key value» – записывает значение одного параметра в хранилище конфигурации. Данная команда не валидирует формат значения, поэтому при использовании необходимо убедиться, что сохраняется значение в корректном формате.

Пример – Пример значений выполнения команды «set-clustered-config key value»:

cd /opt/alphabi/

bin/Bars.Alpha.Web set-clustered-config OlapConfig.ApiKey 12345

bin/Bars.Alpha.Web get-clustered-config OlapConfig.ApiKey

12345

### Конфигурация Alpha BI через переменные окружения

#### Переменные окружения – параметры конфигурации Alpha BI

Существует возможность задавать конфигурацию приложения Alpha BI через переменные окружения (Environment Variables). Переменные окружения как источник конфигурации в общем случае имеют приоритет над файлом конфигурации и централизованным хранилищем конфигурации Etcd.

Имя переменных окружения, задающих конфигурацию Alpha BI, строится следующим образом: ALPHABI\_CONFIG\_{KEY}, где KEY – это название конфигурационного свойства (в качестве разделителя составного имени используется символ «\_\_» (два подчеркивания)). Имена свойств регистронезависимы.

Пример – Пример соответствия переменных окружения и JSON-файла alpha.config:

{

"ControlCenterConfig": {

"Password": "123"

},

"DatabaseConfig": {

"ServerAddress": "127.0.0.1",

"ServerPort": 5432,

"User": "bars",

"Password": "123",

"DatabaseName": "alpha\_developer"

},

"OlapConfig": {

"XmlaEndpoint": "http://localhost:9999/mondrian/xmla",

"ApplicationBaseUrl": "http://localhost:5000/",

"ApiKey": "1234",

},

"FileStorageConfig": {

"RootPath": "/home/user/var/alpha\_file\_storage/alpha"

}

}

Переменные окружения:

* ALPHABI\_CONFIG\_CONTROLCENTERCONFIG\_\_PASSWORD=123;
* ALPHABI\_CONFIG\_DATABASECONFIG\_\_SERVERADDRESS=127.0.0.1;
* ALPHABI\_CONFIG\_DATABASECONFIG\_\_USER=bars;
* ALPHABI\_CONFIG\_DATABASECONFIG\_\_DATABASENAME=alpha\_developer;
* ALPHABI\_CONFIG\_DATABASECONFIG\_\_PASSWORD=123;
* ALPHABI\_CONFIG\_FILESTORAGECONFIG\_\_ROOTPATH=/home/user/var/alpha\_file\_storage/alpha;
* ALPHABI\_CONFIG\_OLAPCONFIG\_\_XMLAENDPOINT=http://localhost:9999/mondrian/xmla;
* ALPHABI\_CONFIG\_OLAPCONFIG\_\_APPLICATIONBASEURL=http://localhost:5000/;
* ALPHABI\_CONFIG\_OLAPCONFIG\_\_APIKEY=1234.

Примечание – Если в переменных окружения заданы все обязательные параметры конфигурации, то наличие файла alpha.config является необязательным.

## Отдельные операции

### Переход из некластерного режима в кластерный режим

Предполагается, что существует развернутый экземпляр Alpha BI. Чтобы создать кластер Alpha BI, выполните следующие действия:

1. выполните развертывание кластера Etcd;
2. остановите Alpha BI:

systemctl stop alphabi

1. в файле **/opt/alphabi/alpha.config** измените значение ключа Cluster.IsClustered на true и заполните параметры подключения к Etcd (ключи внутри Cluster.Etcd):

"Cluster": {

"IsClustered": true,

"Etcd": {

"Host": "${EtcdHost\_1}",

"Port": ${EtcdPort},

"UserName": "${EtcdUser}",

"Password": "${EtcdPassword}",

"KeyPrefix": "${EtcdPrefix}"

}

},

1. повторно произведите процедуру инсталляции (это не приведет к потере данных):

cd /opt/alphabi/

bin/Bars.Alpha.Web install ${AlphaAdminPassword}

1. запустите Alpha BI:

systemctl start alphabi

1. заново произведите установку лицензии (см. п. 5.1.1).

### Развертывание имеющегося дампа

При получении готового дампа проверьте наличие следующих файлов:

* архив исполняемых файлов web-приложения;
* архив файлового хранилища (чаще всего в названии есть ключевое слово «storage»);
* архив базы данных (с расширением .backup).

Для установки приложения выполните следующие действия:

1. остановите web-приложение Alpha BI:

systemctl stop alphabi

1. сделайте резервную копию конфигурационного файла alpha.config:

cp /opt/alpha/alpha.config ~/alpha.config.backup-"$(date '+%Y%m%d-%H%M%S')"

1. обновите приложение из дампа:
2. перейдите в каталог приложения **/opt/alphabi**;
3. удалите все подкаталоги и файлы из каталога приложения;
4. в каталог приложения переместите все подкаталоги и файлы из каталога со сборкой из архива (1. Архив самого приложения).
5. восстановите из сделанной ранее резервной копии конфигурационный файл alpha.config в каталог приложения;
6. восстановите базу данных PostgreSQL:

psql -h localhost -p 5432 -U postgres

1. отсоедините активные сессии:

su postgres -c "psql -c \"select pg\_terminate\_backend(pid) from pg\_stat\_activity where datname='alphabi'\"";

Примечание – Вместо значения alphabi укажите имя созданной БД.

1. удалите существующую БД:

su postgres -c "psql -c \"drop database alphabi\"";

1. создайте новую БД:

su postgres -c "psql -c \"create database alphabi owner alphabi\"";

1. восстановите БД из дампа с помощью pg\_restore:

pg\_restore -h localhost -p 5432 -U alphabi --no-acl --no-owner --dbname "alphabi" --role "alphabi" --verbose "/home/user/alpha.backup"

Примечание – Также для этой цели можно использовать графическую оболочку pgAdmin.

1. восстановите файловое хранилище: удалите и пересоздайте каталог файлового хранилища (аналогично шагам при установке):

rm -rf /var/lib/alphabi-file-storage

mkdir -p /var/lib/alphabi-file-storage

chown -R alphabi /var/lib/alphabi-file-storage

1. запустите web-приложение:

systemctl start alphabi

### Внешний сервер балансировки

#### Внешний сервер балансировки Haproxy

##### Установка зависимостей

Подключите репозиторий SCL:

yum -y install centos-release-scl

##### Первоначальная установка

Установите haproxy:

yum -y install rh-haproxy18

##### Конфигурирование

Для конфигурирования выполните следующие действия:

1. переименуйте стандартный файл конфигурации **/etc/opt/rh/rh-haproxy18/haproxy/haproxy.cfg**:

mv /etc/opt/rh/rh-haproxy18/haproxy/haproxy.cfg /etc/opt/rh/rh-haproxy18/haproxy/haproxy.cfg.backup

1. создайте файл **/etc/opt/rh/rh-haproxy18/haproxy/haproxy.cfg** со следующим содержимым:

# ------------------------------------------------- --------------------

# Example configuration for a possible web application. See the

# full configuration options online.

# # http://haproxy.1wt.eu/download/1.8/doc/configuration.txt

# # ------------------------------------------------- --------------------

# ------------------------------------------------- --------------------

# Global settings

# ------------------------------------------------- --------------------

global

# to have these messages end up in /var/opt/rh/rh-haproxy18/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/opt/rh/rh-haproxy18/log/haproxy.log

# file. A line like the following can be added to

# /etc/sysconfig/syslog

#

# local2.\* /var/opt/rh/rh-haproxy18/log/haproxy.log

#

log 127.0.0.1 local2

chroot /var/opt/rh/rh-haproxy18/lib/haproxy

pidfile /var/run/rh-haproxy18-haproxy.pid

maxconn 4000

user haproxy

group haproxy

daemon

# turn on stats unix socket

stats socket /var/opt/rh/rh-haproxy18/lib/haproxy/stats

# utilize system-wide crypto-policies

ssl-default-bind-ciphers PROFILE=SYSTEM

ssl-default-server-ciphers PROFILE=SYSTEM

# ------------------------------------------------- --------------------

# 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

option http-server-close

option forwardfor except 127.0.0.0/8

option redispatch

retries 3

timeout http-request 15m

timeout queue 5m

timeout connect 10s

timeout client 15m

timeout server 15m

timeout http-keep-alive 65s

timeout check 30s

maxconn 3000

# ------------------------------------------------- --------------------

# main frontend which proxys to the backends

# ------------------------------------------------- --------------------

frontend main

bind \*:80

mode http

http-request deny deny\_status 400 if { req.body\_size gt 1073741824 }

http-request set-header connection keep-alive

option forwardfor header X-Client

acl url\_app path\_dir -i /${ExternalURLPathPrefix}

use\_backend be\_app if url\_app

# ------------------------------------------------- --------------------

# backends

# ------------------------------------------------- --------------------

backend be\_app

mode http

server app1 ${IntBalancerHost}:${WebAppPort} check

Префиксы fe\_ и be\_ расшифровываются как fe (frontend) и be (backend).

1. откройте порт в firewall:

firewall-cmd --permanent --add-port=80/tcp

firewall-cmd –reload

1. разрешите haproxy подключаться к другим серверам:

setsebool -P httpd\_can\_network\_connect 1

setsebool -P haproxy\_connect\_any 1

Это необходимо, чтобы подсистема безопасности SELinux (включена по умолчанию в CentOS, RHEL) позволяла haproxy устанавливать сетевые подключения (в том числе до web-сервера Alpha BI).

##### Запуск

Запустите haproxy:

systemctl enable rh-haproxy18-haproxy

systemctl start rh-haproxy18-haproxy

#### Внешний сервер балансировки HAProxy отказоустойчивый

Чтобы узнать значения ${LAN\_INTERFACE} и ${LAN\_IP}, выполните команду, которая выведет список сетевых интерфейсов и ip-адресов:

ip a

В данном примере рассматривается схема с использованием двух Haproxy-серверов, использующих на frontend 80-й порт. Служба Keepalived устанавливается на оба сервера.

##### Первоначальная установка Keepalived

Примечание – Необходимо, чтобы серверы, на которых будет развернута служба Keepalived, были в одном сегменте сети и чтобы в данном сегменте сети был хотя бы один незарезервированный ip-адрес, который будет выполнять роль плавающего ip-адреса.

Установите keepalived:

yum -y install keepalived

##### Конфигурирование Keepalived

Для конфигурирования Keepalived выполните следующие действия:

1. создайте пользователя keepalived:

useradd keepalived

1. удалите содержимое файла **/etc/keepalived/keepalived.conf** и откройте для редактирования на сервере, выполняющем роль master-сервера, и приведите к виду:

global\_defs {

vrrp\_garp\_master\_refresh 5

script\_user keepalived

enable\_script\_security

}

vrrp\_script chk\_haproxy\_service {

script "/usr/local/bin/keepalived\_haproxy\_check.sh ${LAN\_IP} 80"

timeout 1

interval 1

fall 3

rise 3

}

vrrp\_instance haproxy\_failover {

state MASTER

interface ${LAN\_INTERFACE}

virtual\_router\_id 51

priority 255

virtual\_ipaddress {

${VIRTUAL\_IP}/32 dev ${LAN\_INTERFACE}

}

track\_interface {

${LAN\_INTERFACE}

}

track\_script {

chk\_haproxy\_service

}

notify\_fault "/usr/bin/systemctl restart rh-haproxy18-haproxy" root

}

где:

* ${LAN\_IP} – локальный ip-адрес сервера, который будет использоваться для коммуникации с остальными серверами;
* ${LAN\_INTERFACE} – имя сетевого интерфейса, за которым закреплен ${LAN\_IP};
* ${VIRTUAL\_IP} – плавающий ip-адрес, незанятый ip-адрес в том же сегменте сети, который будет назначаться серверам, где будет использоваться служба keepalived. Плавающий ip-адрес един для экземпляров с одинаковым значением virtual\_router\_id.

1. удалите содержимое файла **/etc/keepalived/keepalived.conf** и откройте для редактирования на сервере, выполняющем роль backup-сервера, и приведите к виду:

global\_defs {

vrrp\_garp\_master\_refresh 5

script\_user keepalived

enable\_script\_security

}

vrrp\_script chk\_haproxy\_service {

script "/usr/local/bin/keepalived\_haproxy\_check.sh ${LAN\_IP} 80"

timeout 1

interval 1

fall 3

rise 3

}

vrrp\_instance haproxy\_failover {

state BACKUP

interface ${LAN\_INTERFACE}

virtual\_router\_id 51

priority 100

virtual\_ipaddress {

${VIRTUAL\_IP}/32 dev ${LAN\_INTERFACE}

}

track\_interface {

${LAN\_INTERFACE}

}

track\_script {

chk\_haproxy\_service

}

notify\_fault "/usr/bin/systemctl restart rh-haproxy18-haproxy" root

}

где:

* ${LAN\_IP} – локальный ip-адрес сервера, который будет использоваться для коммуникации с остальными серверами;
* ${LAN\_INTERFACE} – имя сетевого интерфейса, за которым закреплен ${LAN\_IP};
* ${VIRTUAL\_IP} – плавающий ip-адрес, незанятый ip-адрес в том же сегменте сети, который будет назначаться серверам, где будет использоваться служба keepalived. Плавающий ip-адрес един для экземпляров с одинаковым значением virtual\_router\_id.

1. создайте файл **/usr/local/bin/keepalived\_haproxy\_check.sh** со следующим содержимым:

#!/usr/bin/env bash

[[ -z "$1" ]] && echo "Empty arguments, please enter target ip address." && exit 1

[[ -n $(timeout -s SIGKILL 1 ss -n4 state listening "( sport = :${2:-80} )" src "${1}" or src 0.0.0.0 | tail -n +2) ]] && exit 0 || exit 1

1. предоставьте права на запуск скрипта:

chmod +x /usr/local/bin/keepalived\_haproxy\_check.sh

1. выполните команду:

echo "net.ipv4.ip\_forward = 1" >> /etc/sysctl.conf

sysctl -p

1. примените правила для firewall:

firewall-cmd --direct --permanent --add-rule ipv4 filter INPUT 0 --in-interface ${LAN\_INTERFACE} --destination 224.0.0.18 --protocol vrrp -j ACCEPT

firewall-cmd --direct --permanent --add-rule ipv4 filter OUTPUT 0 --out-interface ${LAN\_INTERFACE} --destination 224.0.0.18 --protocol vrrp -j ACCEPT

firewall-cmd --reload

* где ${LAN\_INTERFACE} – наименование сетевого интерфейса для входящих и исходящих соединений.

##### Запуск Keepalived

Запустите keepalived:

systemctl start keepalived

systemctl enable keepalived

#### Внешний сервер балансировки Nginx

##### Установка зависимостей

Подключите репозиторий EPEL:

yum -y install epel-release

##### Первоначальная установка

Установите nginx:

yum -y install nginx

##### Конфигурирование

Для конфигурирования выполните следующие действия:

1. создайте файл **/etc/nginx/default.d/alphabi.conf** следующего содержания:

location ${ExternalURLPathPrefix} {

proxy\_pass http://${IntBalancerHost}:${WebAppPort}${ExternalURLPathPrefix};

proxy\_http\_version 1.1;

proxy\_set\_header Connection keep-alive;

proxy\_set\_header X-Forwarded-For $proxy\_add\_x\_forwarded\_for;

proxy\_set\_header Host $http\_host;

gzip on;

gzip\_proxied any;

client\_max\_body\_size 1g;

}

1. откройте порт в firewall:

firewall-cmd --permanent --add-port=80/tcp

firewall-cmd –reload

1. разрешите nginx соединяться с остальными серверами:

setsebool -P httpd\_can\_network\_connect 1

Это необходимо, чтобы подсистема безопасности SELinux (включена по умолчанию в CentOS, RHEL) позволила nginx устанавливать сетевые соединения (в том числе до web-сервера Alpha BI).

##### Запуск

Запустите nginx:

systemctl enable nginx

systemctl start nginx

#### Внешний сервер балансировки Nginx отказоустойчивый

Чтобы узнать значения ${LAN\_INTERFACE} и ${LAN\_IP}, выполните команду, которая выведет список сетевых интерфейсов и ip-адресов:

ip a

В данном примере рассматривается схема с использованием двух Nginx-серверов, использующих на frontend 80-й порт. Служба Keepalived устанавливается на оба сервера.

##### Первоначальная установка Keepalived

Примечание – Необходимо, чтобы серверы, на которых будет развернута служба keepalived, были в одном сегменте сети и чтобы в данном сегменте сети был хотя бы один незарезервированный ip-адрес, который будет выполнять роль плавающего ip-адреса.

Установите keepalived:

yum -y install keepalived

##### Конфигурирование Keepalived

Для конфигурирования Keepalived выполните следующие действия:

1. создайте пользователя keepalived:

useradd keepalived

1. удалите содержимое файла **/etc/keepalived/keepalived.conf** и откройте для редактирования на сервере, выполняющем роль master-сервера, и приведите к виду:

global\_defs {

vrrp\_garp\_master\_refresh 5

script\_user keepalived

enable\_script\_security

}

vrrp\_script chk\_nginx\_service {

script "/usr/local/bin/keepalived\_nginx\_check.sh ${LAN\_IP} 80"

timeout 1

interval 1

fall 3

rise 3

}

vrrp\_instance nginx\_failover {

state MASTER

interface ${LAN\_INTERFACE}

virtual\_router\_id 51

priority 255

virtual\_ipaddress {

${VIRTUAL\_IP}/32 dev ${LAN\_INTERFACE}

}

track\_interface {

${LAN\_INTERFACE}

}

track\_script {

chk\_nginx\_service

}

notify\_fault "/usr/bin/systemctl restart nginx" root

}

где:

* ${LAN\_IP} – локальный ip-адрес сервера, который будет использоваться для коммуникации с остальными серверами;
* ${LAN\_INTERFACE} – имя сетевого интерфейса, за которым закреплен ${LAN\_IP};
* ${VIRTUAL\_IP} – плавающий ip-адрес, незанятый ip-адрес в том же сегменте сети, который будет назначаться серверам, где будет использоваться служба keepalived. Плавающий ip-адрес един для экземпляров с одинаковым значением virtual\_router\_id.

1. удалите содержимое файла **/etc/keepalived/keepalived.conf** и откройте для редактирования на сервере, выполняющем роль backup-сервера, и приведите к виду:

global\_defs {

vrrp\_garp\_master\_refresh 5

script\_user keepalived

enable\_script\_security

}

vrrp\_script chk\_nginx\_service {

script "/usr/local/bin/keepalived\_nginx\_check.sh ${LAN\_IP} 80"

timeout 1

interval 1

fall 3

rise 3

}

vrrp\_instance nginx\_failover {

state BACKUP

interface ${LAN\_INTERFACE}

virtual\_router\_id 51

priority 100

virtual\_ipaddress {

${VIRTUAL\_IP}/32 dev ${LAN\_INTERFACE}

}

track\_interface {

${LAN\_INTERFACE}

}

track\_script {

chk\_nginx\_service

}

notify\_fault "/usr/bin/systemctl restart nginx" root

}

где:

* ${LAN\_IP} – локальный ip-адрес сервера, который будет использоваться для коммуникации с остальными серверами;
* ${LAN\_INTERFACE} – имя сетевого интерфейса, за которым закреплен ${LAN\_IP};
* ${VIRTUAL\_IP} – плавающий ip-адрес, незанятый ip-адрес в том же сегменте сети, который будет назначаться серверам, где будет использоваться служба keepalived. Плавающий ip-адрес един для экземпляров с одинаковым значением virtual\_router\_id.

1. создайте файл **/usr/local/bin/keepalived\_nginx\_check.sh** со следующим содержимым:

#!/usr/bin/env bash

[[ -z "$1" ]] && echo "Empty arguments, please enter target ip address." && exit 1

[[ -n $(timeout -s SIGKILL 1 ss -n4 state listening "( sport = :${2:-80} )" src "${1}" or src 0.0.0.0 | tail -n +2) ]] && exit 0 || exit 1

1. предоставьте права на запуск скрипта:

chmod +x /usr/local/bin/keepalived\_nginx\_check.sh

1. выполните команду:

echo "net.ipv4.ip\_forward = 1" >> /etc/sysctl.conf

sysctl -p

1. примените правила для firewall:

firewall-cmd --direct --permanent --add-rule ipv4 filter INPUT 0 --in-interface ${LAN\_INTERFACE} --destination 224.0.0.18 --protocol vrrp -j ACCEPT

firewall-cmd --direct --permanent --add-rule ipv4 filter OUTPUT 0 --out-interface ${LAN\_INTERFACE} --destination 224.0.0.18 --protocol vrrp -j ACCEPT

firewall-cmd --reload

* где ${LAN\_INTERFACE} – наименование сетевого интерфейса для входящих и исходящих соединений.

##### Запуск Keepalived

Запустите keepalived:

systemctl start keepalived

systemctl enable keepalived

### Внутренний сервер балансировки

#### Внутренний сервер балансировки Haproxy

##### Зависимости

Выполните следующие действия:

1. подключите репозиторий SCL:

yum -y install centos-release-scl

1. установите утилиту для управления SELinux:

yum -y install policycoreutils-python

##### Первоначальная установка

Установите haproxy:

yum -y install rh-haproxy18

##### Конфигурирование

Для конфигурирования выполните следующие действия:

1. переименуйте стандартный файл конфигурации **/etc/opt/rh/rh-haproxy18/haproxy/haproxy.cfg**:

mv /etc/opt/rh/rh-haproxy18/haproxy/haproxy.cfg /etc/opt/rh/rh-haproxy18/haproxy/haproxy.cfg.backup

1. создайте файл **/etc/opt/rh/rh-haproxy18/haproxy/haproxy.cfg** со следующим содержимым:

#---------------------------------------------------------------------

# Example configuration for a possible web application. See the

# full configuration options online.

#

# http://haproxy.1wt.eu/download/1.8/doc/configuration.txt

#

#---------------------------------------------------------------------

#---------------------------------------------------------------------

# Global settings

#---------------------------------------------------------------------

global

# to have these messages end up in /var/opt/rh/rh-haproxy18/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/opt/rh/rh-haproxy18/log/haproxy.log

# file. A line like the following can be added to

# /etc/sysconfig/syslog

#

# local2.\* /var/opt/rh/rh-haproxy18/log/haproxy.log

#

log 127.0.0.1 local0 info

chroot /var/opt/rh/rh-haproxy18/lib/haproxy

pidfile /var/run/rh-haproxy18-haproxy.pid

maxconn 4000

user haproxy

group haproxy

daemon

# turn on stats unix socket

stats socket /var/opt/rh/rh-haproxy18/lib/haproxy/stats

# utilize system-wide crypto-policies

ssl-default-bind-ciphers PROFILE=SYSTEM

ssl-default-server-ciphers PROFILE=SYSTEM

#---------------------------------------------------------------------

# 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

option http-server-close

option forwardfor except 127.0.0.0/8

option redispatch

retries 3

timeout http-request 15m

timeout queue 5m

timeout connect 10s

timeout client 15m

timeout server 15m

timeout http-keep-alive 65s

timeout check 30s

maxconn 3000

#---------------------------------------------------------------------

# main frontend which proxys to the backends

#---------------------------------------------------------------------

frontend main

bind \*:${WebAppPort}

mode http

http-request deny deny\_status 400 if { req.body\_size gt 1073741824 }

http-request set-header connection keep-alive

option forwardfor header X-Client

acl url\_app\_datasources path\_dir -i /${ExternalURLPathPrefix}/mondrian/

use\_backend be\_app\_datasources if url\_app\_datasources

acl url\_app path\_dir -i /${ExternalURLPathPrefix}

use\_backend be\_app if url\_app

frontend olap

bind \*:${MondrianPort}

mode http

http-request deny deny\_status 400 if { req.body\_size gt 1073741824 }

default\_backend be\_olap

#---------------------------------------------------------------------

# leastconn balancing between the various backends

#---------------------------------------------------------------------

backend be\_app

mode http

balance leastconn

server app1 ${WebAppHost\_1}:${WebAppPort} check maxconn 300

# ...

server appN ${WebAppHost\_N}:${WebAppPort} check maxconn 300

backend be\_app\_datasources

mode http

balance leastconn

server app1 ${WebAppHost\_1}:${WebAppPort} check

# ...

server appN ${WebAppHost\_N}:${WebAppPort} check

backend be\_olap

mode http

server olap1 ${MondrianHost}:${MondrianPort} check

1. создайте файл **/etc/rsyslog.d/haproxy.conf** со следующим содержимым:

$ModLoad imudp

$UDPServerRun 514

$UDPServerAddress 127.0.0.1

# Send HAProxy messages to a dedicated logfile

if $programname startswith 'haproxy' then /var/log/haproxy/haproxy.log

&~

1. перезапустите службу rsyslog:

systemctl restart rsyslog

1. разрешите подключения от других серверов:

firewall-cmd --permanent --add-port=${WebAppPort}/tcp

firewall-cmd --permanent --add-port=${MondrianPort}/tcp

firewall-cmd --reload

1. разрешите haproxy соединяться с остальными серверами и принимать соединения:

setsebool -P httpd\_can\_network\_connect 1

setsebool -P haproxy\_connect\_any 1

semanage port -m -t http\_port\_t -p tcp ${WebAppPort}

semanage port -m -t http\_port\_t -p tcp ${MondrianPort}

Это необходимо, чтобы подсистема безопасности SELinux (по умолчанию включена в CentOS, RHEL) позволила haproxy устанавливать сетевые соединения (в том числе до web-сервера Alpha BI).

##### Запуск

Запустите haproxy:

systemctl enable rh-haproxy18-haproxy

systemctl start rh-haproxy18-haproxy

#### Внутренний сервер балансировки HAProxy отказоустойчивый

Чтобы узнать значения ${LAN\_INTERFACE} и ${LAN\_IP}, выполните команду, которая выведет список сетевых интерфейсов и ip-адресов:

ip a

В данном примере рассматривается схема с использованием двух Haproxy-серверов, использующих на frontend 80-й порт. Служба Keepalived устанавливается на оба сервера.

##### Первоначальная установка Keepalived

Примечание – Необходимо, чтобы серверы, на которых будет развернута служба keepalived, были в одном сегменте сети и чтобы в данном сегменте сети был хотя бы один незарезервированный ip-адрес, который будет выполнять роль плавающего ip-адреса.

Установите keepalived:

yum -y install keepalived

##### Конфигурирование Keepalived

Для конфигурирования Keepalived выполните следующие действия:

1. создайте пользователя keepalived:

useradd keepalived

1. удалите содержимое файла **/etc/keepalived/keepalived.conf** и откройте для редактирования на сервере, выполняющем роль master-сервера, и приведите к виду:

global\_defs {

vrrp\_garp\_master\_refresh 5

script\_user keepalived

enable\_script\_security

}

vrrp\_script chk\_haproxy\_service {

script "/usr/local/bin/keepalived\_haproxy\_check.sh ${LAN\_IP} 80"

timeout 1

interval 1

fall 3

rise 3

}

vrrp\_instance haproxy\_failover {

state MASTER

interface ${LAN\_INTERFACE}

virtual\_router\_id 51

priority 255

virtual\_ipaddress {

${VIRTUAL\_IP}/32 dev ${LAN\_INTERFACE}

}

track\_interface {

${LAN\_INTERFACE}

}

track\_script {

chk\_haproxy\_service

}

notify\_fault "/usr/bin/systemctl restart rh-haproxy18-haproxy" root

}

где:

* ${LAN\_IP} – локальный ip-адрес сервера, который будет использоваться для коммуникации с остальными серверами;
* ${LAN\_INTERFACE} – имя сетевого интерфейса, за которым закреплен ${LAN\_IP};
* ${VIRTUAL\_IP} – плавающий ip-адрес, незанятый ip-адрес в том же сегменте сети, который будет назначаться серверам, где будет использоваться служба keepalived. Плавающий ip-адрес един для экземпляров с одинаковым значением virtual\_router\_id.

1. удалите содержимое файла **/etc/keepalived/keepalived.conf** и откройте для редактирования на сервере, выполняющем роль backup-сервера, и приведите к виду:

global\_defs {

vrrp\_garp\_master\_refresh 5

script\_user keepalived

enable\_script\_security

}

vrrp\_script chk\_haproxy\_service {

script "/usr/local/bin/keepalived\_haproxy\_check.sh ${LAN\_IP} 80"

timeout 1

interval 1

fall 3

rise 3

}

vrrp\_instance haproxy\_failover {

state BACKUP

interface ${LAN\_INTERFACE}

virtual\_router\_id 51

priority 100

virtual\_ipaddress {

${VIRTUAL\_IP}/32 dev ${LAN\_INTERFACE}

}

track\_interface {

${LAN\_INTERFACE}

}

track\_script {

chk\_haproxy\_service

}

notify\_fault "/usr/bin/systemctl restart rh-haproxy18-haproxy" root

}

где:

* ${LAN\_IP} – локальный ip-адрес сервера, который будет использоваться для коммуникации с остальными серверами;
* ${LAN\_INTERFACE} – имя сетевого интерфейса, за которым закреплен ${LAN\_IP};
* ${VIRTUAL\_IP} – плавающий ip-адрес, незанятый ip-адрес в том же сегменте сети, который будет назначаться серверам, где будет использоваться служба keepalived. Плавающий ip-адрес един для экземпляров с одинаковым значением virtual\_router\_id.

1. создайте файл **/usr/local/bin/keepalived\_haproxy\_check.sh** со следующим содержимым:

#!/usr/bin/env bash

[[ -z "$1" ]] && echo "Empty arguments, please enter target ip address." && exit 1

[[ -n $(timeout -s SIGKILL 1 ss -n4 state listening "( sport = :${2:-80} )" src "${1}" or src 0.0.0.0 | tail -n +2) ]] && exit 0 || exit 1

1. предоставьте права на запуск скрипта:

chmod +x /usr/local/bin/keepalived\_haproxy\_check.sh

1. выполните команду:

echo "net.ipv4.ip\_forward = 1" >> /etc/sysctl.conf

sysctl -p

1. примените правила для firewall:

firewall-cmd --direct --permanent --add-rule ipv4 filter INPUT 0 --in-interface ${LAN\_INTERFACE} --destination 224.0.0.18 --protocol vrrp -j ACCEPT

firewall-cmd --direct --permanent --add-rule ipv4 filter OUTPUT 0 --out-interface ${LAN\_INTERFACE} --destination 224.0.0.18 --protocol vrrp -j ACCEPT

firewall-cmd --reload

* где ${LAN\_INTERFACE} – наименование сетевого интерфейса для входящих и исходящих соединений.

##### Запуск Keepalived

Запустите keepalived:

systemctl start keepalived

systemctl enable keepalived

#### Внутренний сервер балансировки Nginx

##### Зависимости

Для настройки зависимостей выполните следующие действия:

1. подключите репозиторий EPEL:

yum -y install epel-release

1. установите утилиту для управления SELinux:

yum -y install policycoreutils-python

##### Первоначальная установка

Установите nginx:

yum -y install nginx

##### Конфигурирование

Для конфигурирования выполните следующие действия:

1. создайте файл конфигурации **/etc/nginx/conf.d/alphabi.conf** следующего содержания:

upstream upstream\_alphabi\_webapp {

server ${WebAppHost\_1}:${WebAppPort} max\_conns=300;

# ...

server ${WebAppHost\_n}:${WebAppPort} max\_conns=300;

keepalive 500;

}

upstream upstream\_alphabi\_mondrian {

server ${MondrianHost}:${MondrianPort} max\_conns=100;

}

server {

listen ${WebAppPort};

listen [::]:${WebAppPort};

location ${ExternalURLPathPrefix} {

proxy\_read\_timeout 15m;

proxy\_pass http://upstream\_alphabi\_webapp${ExternalURLPathPrefix};

proxy\_http\_version 1.1;

proxy\_set\_header Connection keep-alive;

proxy\_set\_header X-Forwarded-For $proxy\_add\_x\_forwarded\_for;

proxy\_set\_header Host $http\_host;

gzip on;

gzip\_proxied any;

client\_max\_body\_size 1g;

#access\_log /var/log/nginx/access\_alphabi\_webapp.log; # Не рекомендуется включение на высоконагруженных приложениях

error\_log /var/log/nginx/error\_alphabi\_webapp.log;

}

}

server {

listen ${MondrianPort};

listen [::]:${MondrianPort};

location /${MondrianAppName} {

proxy\_read\_timeout 30m;

proxy\_pass http://upstream\_alphabi\_mondrian/${MondrianAppName};

proxy\_http\_version 1.1;

proxy\_set\_header Connection keep-alive;

proxy\_set\_header Host $http\_host;

client\_max\_body\_size 1g;

#access\_log /var/log/nginx/access\_alphabi\_mondrian.log; # Не рекомендуется включение на высоконагруженных приложениях

error\_log /var/log/nginx/error\_alphabi\_mondrian.log;

}

}

Примечание – Строчку "server …" внутри upstream\_alphabi\_webapp повторите для каждого сервера приложения.

1. разрешите подключения от других серверов:

firewall-cmd --permanent --add-port=${WebAppPort}/tcp

firewall-cmd --permanent --add-port=${MondrianPort}/tcp

firewall-cmd --reload

1. разрешите nginx соединяться с остальными серверами и принимать соединения:

setsebool -P httpd\_can\_network\_connect 1

semanage port -m -t http\_port\_t -p tcp 5000

semanage port -m -t http\_port\_t -p tcp 5001

Это необходимо, чтобы подсистема безопасности SELinux (по умолчанию включена в CentOS, RHEL) позволила nginx устанавливать сетевые соединения (в том числе до web-сервера Alpha BI).

##### Запуск

Запустите nginx:

systemctl enable nginx

systemctl start nginx

#### Внутренний сервер балансировки Nginx отказоустойчивый

Чтобы узнать значения ${LAN\_INTERFACE} и ${LAN\_IP}, выполните команду, которая выведет список сетевых интерфейсов и ip-адресов:

ip a

В данном примере рассматривается схема с использованием двух Nginx-серверов, использующих на frontend 80-й порт. Служба Keepalived устанавливается на оба сервера.

##### Первоначальная установка Keepalived

Примечание – Необходимо, чтобы серверы, на которых будет развернута служба keepalived, были в одном сегменте сети и чтобы в данном сегменте сети был хотя бы один незарезервированный ip-адрес, который будет выполнять роль плавающего ip-адреса.

Установите keepalived:

yum -y install keepalived

##### Конфигурирование Keepalived

Для конфигурирования Keepalived выполните следующие действия:

1. создайте пользователя keepalived:

useradd keepalived

1. удалите содержимое файла **/etc/keepalived/keepalived.conf** и откройте для редактирования на сервере, выполняющем роль master-сервера, и приведите к виду:

global\_defs {

vrrp\_garp\_master\_refresh 5

script\_user keepalived

enable\_script\_security

}

vrrp\_script chk\_nginx\_service {

script "/usr/local/bin/keepalived\_nginx\_check.sh ${LAN\_IP} 80"

timeout 1

interval 1

fall 3

rise 3

}

vrrp\_instance nginx\_failover {

state MASTER

interface ${LAN\_INTERFACE}

virtual\_router\_id 51

priority 255

virtual\_ipaddress {

${VIRTUAL\_IP}/32 dev ${LAN\_INTERFACE}

}

track\_interface {

${LAN\_INTERFACE}

}

track\_script {

chk\_nginx\_service

}

notify\_fault "/usr/bin/systemctl restart nginx" root

}

где:

* ${LAN\_IP} – локальный ip-адрес сервера, который будет использоваться для коммуникации с остальными серверами;
* ${LAN\_INTERFACE} – имя сетевого интерфейса, за которым закреплен ${LAN\_IP};
* ${VIRTUAL\_IP} – плавающий ip-адрес, незанятый ip-адрес в том же сегменте сети, который будет назначаться серверам, где будет использоваться служба keepalived. Плавающий ip-адрес един для экземпляров с одинаковым значением virtual\_router\_id.

1. удалите содержимое файла **/etc/keepalived/keepalived.conf** и откройте для редактирования на сервере, выполняющем роль backup-сервера, и приведите к виду:

global\_defs {

vrrp\_garp\_master\_refresh 5

script\_user keepalived

enable\_script\_security

}

vrrp\_script chk\_nginx\_service {

script "/usr/local/bin/keepalived\_nginx\_check.sh ${LAN\_IP} 80"

timeout 1

interval 1

fall 3

rise 3

}

vrrp\_instance nginx\_failover {

state BACKUP

interface ${LAN\_INTERFACE}

virtual\_router\_id 51

priority 100

virtual\_ipaddress {

${VIRTUAL\_IP}/32 dev ${LAN\_INTERFACE}

}

track\_interface {

${LAN\_INTERFACE}

}

track\_script {

chk\_nginx\_service

}

notify\_fault "/usr/bin/systemctl restart nginx" root

}

где:

* ${LAN\_IP} – локальный ip-адрес сервера, который будет использоваться для коммуникации с остальными серверами;
* ${LAN\_INTERFACE} – имя сетевого интерфейса, за которым закреплен ${LAN\_IP};
* ${VIRTUAL\_IP} – плавающий ip-адрес, незанятый ip-адрес в том же сегменте сети, который будет назначаться серверам, где будет использоваться служба keepalived. Плавающий ip-адрес един для экземпляров с одинаковым значение virtual\_router\_id.

1. создайте файл **/usr/local/bin/keepalived\_nginx\_check.sh** со следующим содержимым:

#!/usr/bin/env bash

[[ -z "$1" ]] && echo "Empty arguments, please enter target ip address." && exit 1

[[ -n $(timeout -s SIGKILL 1 ss -n4 state listening "( sport = :${2:-80} )" src "${1}" or src 0.0.0.0 | tail -n +2) ]] && exit 0 || exit 1

1. предоставьте права на запуск скрипта:

chmod +x /usr/local/bin/keepalived\_nginx\_check.sh

1. выполните команду:

echo "net.ipv4.ip\_forward = 1" >> /etc/sysctl.conf

sysctl -p

1. примените правила для firewall:

firewall-cmd --direct --permanent --add-rule ipv4 filter INPUT 0 --in-interface ${LAN\_INTERFACE} --destination 224.0.0.18 --protocol vrrp -j ACCEPT

firewall-cmd --direct --permanent --add-rule ipv4 filter OUTPUT 0 --out-interface ${LAN\_INTERFACE} --destination 224.0.0.18 --protocol vrrp -j ACCEPT

firewall-cmd --reload

* где ${LAN\_INTERFACE} – наименование сетевого интерфейса для входящих и исходящих соединений.

##### Запуск Keepalived

Запустите keepalived:

systemctl start keepalived

systemctl enable keepalived

### Настройка расположения файлов приложения

При развертывании приложения Alpha BI возможно размещать файлы приложения в нескольких каталогах файловой системы:

* каталог с исполняемыми и неизменяемыми файлам приложения. Содержит каталог bin;
* рабочий каталог приложения. Содержит следующие изменяемые файлы:
  + конфигурационный файл приложения (alpha.config);
  + лицензия;
  + информация о текущем узле кластера (cluster-node-info.json);
  + логи приложения.

Путь к каталогу с исполняемыми файлами приложения определяется автоматически на основе запущенного файла Bars.Alpha.Web.

Путь к рабочему каталогу задается одним из следующих способов (в порядке приоритета):

* аргумент командной строки:

--working-directory "Полный путь до рабочего каталога".

* переменная окружения ALPHABI\_WORKING\_DIRECTORY;
* текущий каталог.

Примечание – Путь к рабочему каталогу следует указывать также при вызове субкоманд bin/Bars.Alpha.Web install, bin/Bars.Alpha.Web update.

#### Настройки расположения файлов приложения в Linux

##### Вариант 1

Настройка разнесения конфигурационных файлов и исполняемых файлов в Linux достигается путем правки systemd unit-файла, с модификацией опции ExecStart, при использовании аргументов командной строки:

[Unit]

Description=AlphaBI: Web application

After=syslog.target network.target

[Service]

User=alphabi

Environment=ASPNETCORE\_ENVIRONMENT=Production

Environment=ASPNETCORE\_URLS=http://0.0.0.0:5000/

Environment=ASPNETCORE\_URL\_BASE=/alphabi

Environment=ASPNETCORE\_USE\_XFORWARDEDFOR=true

WorkingDirectory=/opt/alphabi/alphabi-bin

ExecStart=/opt/alphabi/alphabi-bin/bin/Bars.Alpha.Web

RestartSec=10

Restart=always

[Install]

WantedBy=multi-user.target

где:

* **/opt/alphabi/alphabi-conf** – рабочий каталог с конфигурационными файлами;
* **/opt/alphabi/alphabi-bin** – рабочий каталог с исполняемыми файлами.

##### Вариант 2

Настройка разнесения конфигурационных файлов и исполняемых файлов в Linux достигается путем правки systemd unit-файла, с модификацией опций Environment, при использовании переменной окружения:

[Unit]

Description=AlphaBI: Web application

After=syslog.target network.target

[Service]

User=alphabi

Environment=ASPNETCORE\_ENVIRONMENT=Production

Environment=ASPNETCORE\_URLS=http://0.0.0.0:5000/

Environment=ASPNETCORE\_URL\_BASE=/alphabi

Environment=ASPNETCORE\_USE\_XFORWARDEDFOR=true

Environment=ALPHABI\_WORKING\_DIRECTORY=/opt/alphabi/alphabi-conf

WorkingDirectory=/opt/alphabi/alphabi-bin

ExecStart=/opt/alphabi/alphabi-bin/bin/Bars.Alpha.Web

RestartSec=10

Restart=always

[Install]

WantedBy=multi-user.target

где:

* **/opt/alphabi/alphabi-conf** – рабочий каталог с конфигурационными файлами;
* **/opt/alphabi/alphabi-bin** – рабочий каталог с исполняемыми файлами.

#### Настройка расположения файлов приложения в Windows

Настройка разнесения конфигурационных файлов и исполняемых файлов в Windows достигается следующими способами:

* добавление пула приложения;
* добавление приложения в Defaul Web Site с указанием каталога с исполняемыми файлами и файлом конфигурации web.config;
* создание в корневом каталоге IIS дополнительных каталогов для хранения конфигурационных файлов Alpha BI и каталога для хранения логов.

##### Вариант 1

Для модификации конфигурационного файла web.config выполните следующую команду:

<?xml version="1.0" encoding="utf-8"?>

<configuration>

<location path="." inheritInChildApplications="false">

<system.webServer>

<handlers>

<add name="aspNetCore" path="\*" verb="\*" modules="AspNetCoreModule" resourceType="Unspecified" />

</handlers>

<directoryBrowse enabled="true" showFlags="Date,Time,Extension,Size" />

<aspNetCore processPath=".\bin\Bars.Alpha.Web.exe" arguments="--working-directory C:\WEB\WWW\alphabi-conf" stdoutLogEnabled="false" stdoutLogFile=".\logs\stdout" />

</system.webServer>

</location>

</configuration>

##### Вариант 2

Для использования переменных окружения выполните следующие действия:

1. нажмите по имени приложения в Default Web Site;
2. выберите файл Configuration Editor (Рисунок 18);

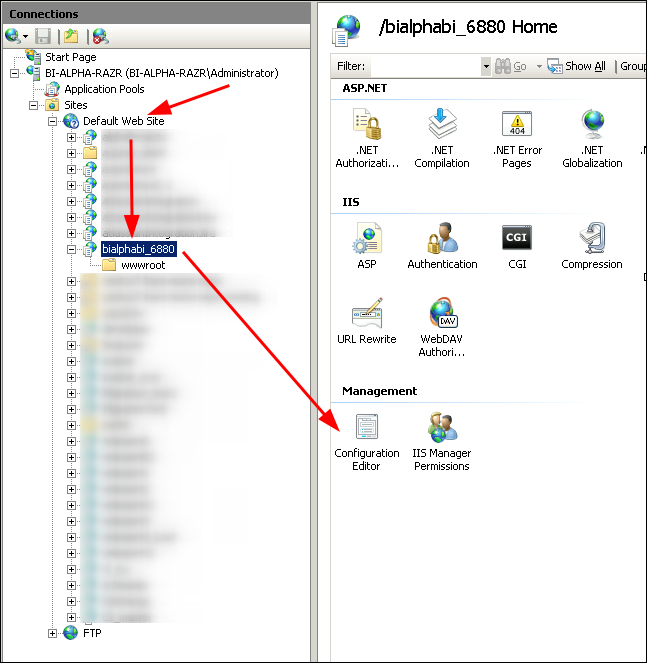


Рисунок 18 – Выбор файла Configuration Editor

1. в поле Section из выпадающего меню по дереву выберите значение system.webServer/aspNetCore;
2. найдите параметр environmetVariables и справа нажмите на кнопку  (Рисунок 19);

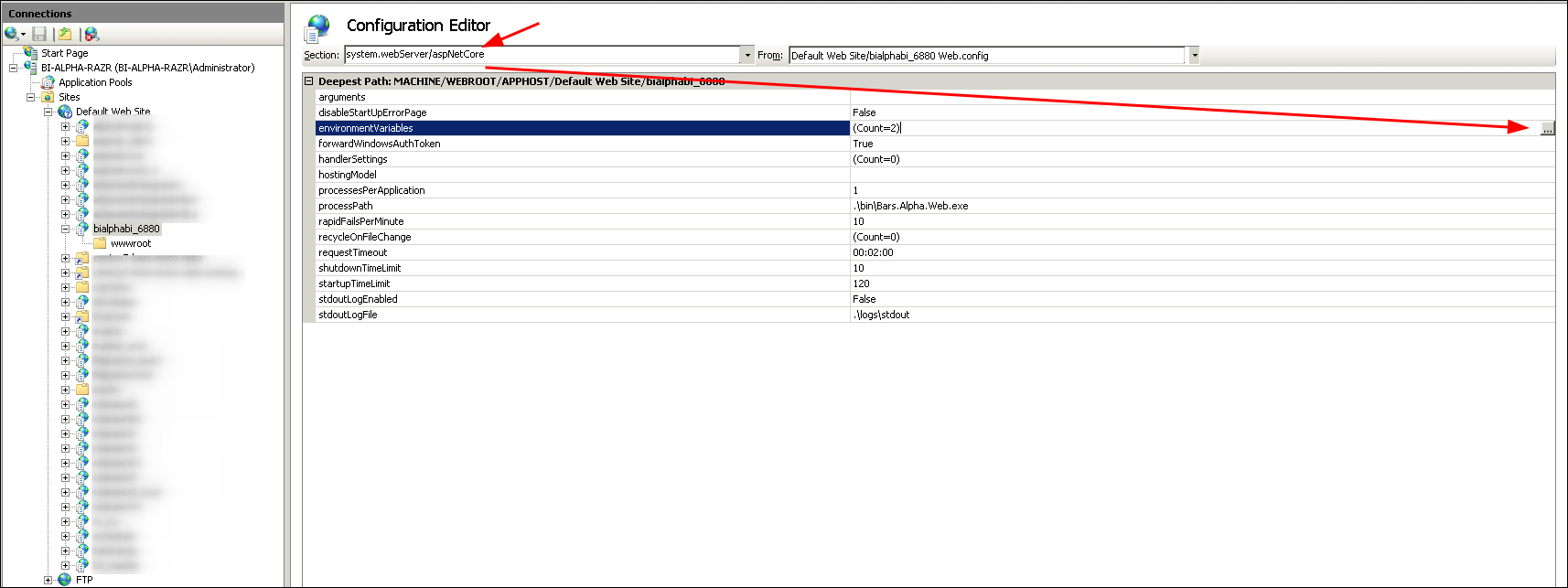


Рисунок  – Параметр environmetVariables

1. в открывшемся окне добавьте переменные окружения, как показано на рисунке (Рисунок 20);

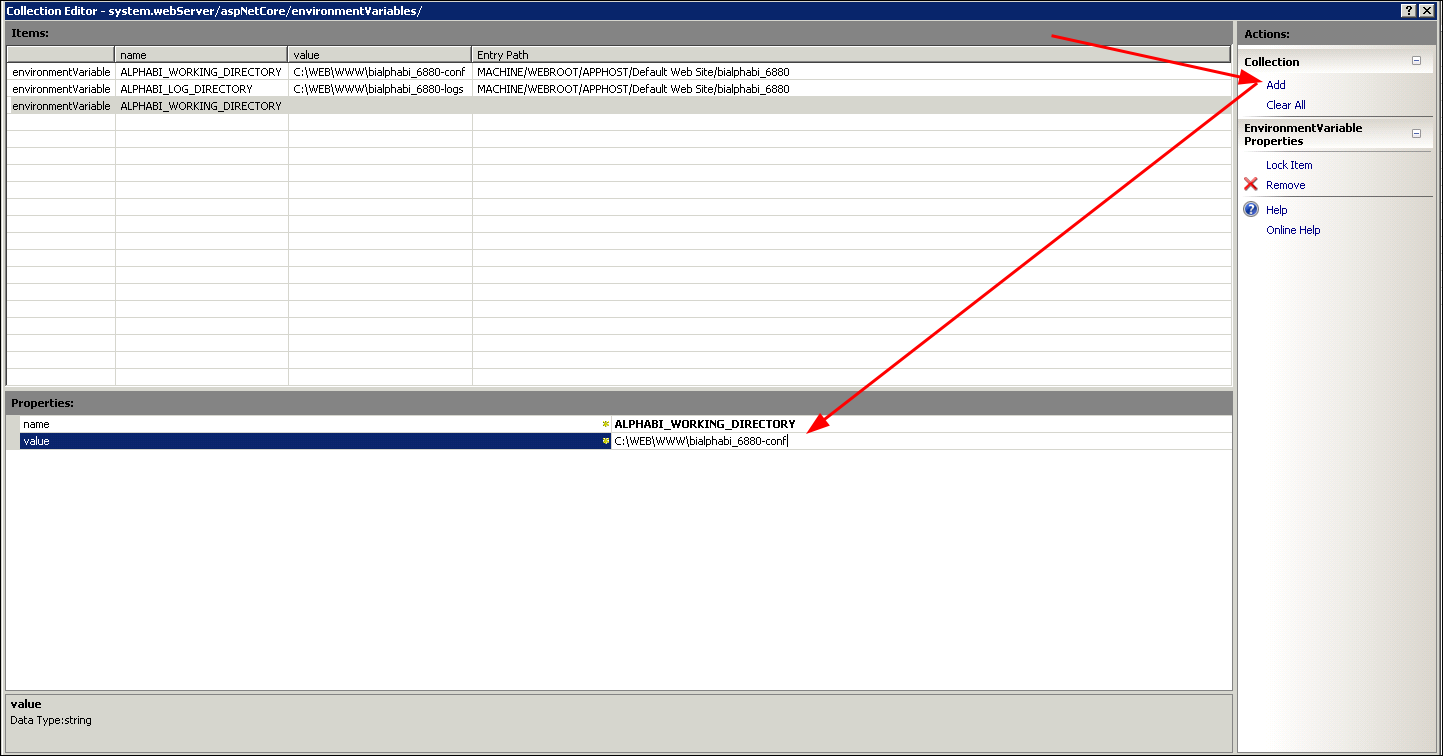


Рисунок – Добавление переменных окружения

* где **C:\WEB\WWW\alphabi-conf** – каталог хранения исполняемых файлов.

В конечном итоге будет автоматически сгенерирован файл web.config следующего содержания:

<?xml version="1.0" encoding="utf-8"?>

<configuration>

<location path="." inheritInChildApplications="false">

<system.webServer>

<handlers>

<add name="aspNetCore" path="\*" verb="\*" modules="AspNetCoreModule" resourceType="Unspecified" />

</handlers>

<directoryBrowse enabled="true" showFlags="Date,Time,Extension,Size" />

<aspNetCore processPath=".\bin\Bars.Alpha.Web.exe" stdoutLogEnabled="false" stdoutLogFile=".\logs\stdout">

<environmentVariables>

<environmentVariable name="ALPHABI\_WORKING\_DIRECTORY" value="C:\WEB\WWW\alphabi-conf" />

</environmentVariables>

</aspNetCore>

</system.webServer>

</location>

<appSettings>

<add key="ALPHABI\_WORKING\_DIRECTORY" value="C:\WEB\WWW\alphabi-conf" />

</appSettings>

</configuration>

### Сервер файлового хранилища отказоустойчивый

В данной схеме используются два сервера, на которых развернуты службы (NFS, Lsyncd и Keepalived).

Чтобы узнать значения ${LAN\_INTERFACE} и ${LAN\_IP}, выполните команду, которая выведет список сетевых интерфейсов и ip-адресов:

ip a

При настройке Lsyncd используется схема двусторонней репликации, поэтому в конфигурационных файлах прописывается адрес противоположного сервера (значение ${NFS\_SERVER}).

#### Первоначальная установка NFS

Установите сервер NFS:

yum -y install nfs-utils

#### Конфигурирование NFS

Для конфигурирования NFS выполните следующие действия:

1. создайте каталог для хранения файлов и назначьте его владельцем пользователя nfsnobody:

mkdir /var/lib/alphabi\_file\_storage

chown nfsnobody:nfsnobody /var/lib/alphabi\_file\_storage/

1. зарегистрируйте каталог с файлами в NFS-сервере:

echo "/var/lib/alphabi\_file\_storage \*(rw,sync,all\_squash,fsid=0)" >> /etc/exports

exportfs -a

Примечание – Символ \* можно заменить на указание подсети, содержащей серверы приложений (в виде x.y.z.w/n).

1. откройте для редактирования файл **/etc/sysconfig/nfs** и раскомментируйте следующие строки, содержащие:

LOCKD\_TCPPORT=32803

LOCKD\_UDPPORT=32769

MOUNTD\_PORT=892

STATD\_PORT=662

NFSD\_V4\_GRACE=90

NFSD\_V4\_LEASE=90

Для параметров NFSD\_V4\_GRACE и NFSD\_V4\_LEASE укажите значение 10, тем самым приведя к виду:

NFSD\_V4\_GRACE=10

NFSD\_V4\_LEASE=10

Сохраните внесенные изменения и закройте файл.

1. откройте необходимые порты в firewall и перезапустите службу NFS:

firewall-cmd --permanent --add-port=32803/tcp

firewall-cmd --permanent --add-port=32769/udp

firewall-cmd --permanent --add-port=892/tcp

firewall-cmd --permanent --add-port=892/udp

firewall-cmd --permanent --add-port=662/tcp

firewall-cmd --permanent --add-port=662/udp

firewall-cmd --permanent --add-port=2049/tcp

firewall-cmd --permanent --add-port=2049/udp

firewall-cmd --permanent --add-port=111/tcp

firewall-cmd --permanent --add-port=111/udp

firewall-cmd --reload

systemctl restart nfs

Примечание – Выше описано, как настроить анонимный NFS-сервер. Рекомендуется настраивать только в защищенной сети.

#### Запуск NFS

Запустите NFS-сервер:

systemctl enable nfs-server

systemctl start nfs-server

#### Первоначальная установка Lsyncd

Установите репозиторий epel, затем установите lsyncd:

yum -y install epel-release

yum -y install lsyncd

#### Конфигурирование Lsyncd

Для конфигурирования Lsyncd выполните следующие действия:

1. создайте пользователя nfssync:

useradd nfssync

1. назначьте пароль для пользователя nfssync:

passwd nfssync

1. сгенерируйте пользовательский SSH-ключ (на все вопросы ответьте положительно, нажав клавишу <Enter>) и сохраните публичный SSH-ключ на сервер-репликации. Необходимо будет ввести пароль пользователя nfssync на сервере-репликации. Перед выполнением данного шага необходимо, чтобы на сервере-репликации были выполнены шаги 1), 2). Шаги 3), 4) выполните поочередно на обоих серверах.

su - nfssync -c "ssh-keygen -t rsa"

su - nfssync

ssh-copy-id nfssync@{NFS\_SERVER}

* где ${NFS\_SERVER} – ip-адрес или доменное имя сервера-репликации.

1. предоставьте права на запись в каталог **/var/lib/alphabi\_file\_storage** пользователю nfssync:

setfacl -R -d -m u:nfssync:rwx /var/lib/alphabi\_file\_storage

setfacl -R -m u:nfssync:rwx /var/lib/alphabi\_file\_storage

1. отредактируйте файл **/etc/lsyncd.conf** следующим образом на обоих серверах, подставляя актуальные адреса серверов, данная конфигурация выполняет двунаправленную репликацию данных:

settings {

logfile = "/var/log/lsyncd/lsyncd.log",

statusFile = "/var/log/lsyncd/lsyncd.status"

}

sync {

default.rsyncssh,

source = "/var/lib/alphabi\_file\_storage/",

host = "nfssync@${NFS\_SERVER}",

targetdir = "/var/lib/alphabi\_file\_storage/",

exclude="lockfile",

rsync = {

rsh = "/usr/bin/ssh -l nfssync -i /home/nfssync/.ssh/id\_rsa -o StrictHostKeyChecking=no",

\_extra = { "--omit-dir-times"}

},

delay=1

}

где:

* ${NFS\_SERVER} – ip-адрес или доменное имя сервера-репликации;
* /var/lib/alphabi\_file\_storage/ – каталог, созданный в шаге 1) в п. 4.5.6.2.

#### Запуск Lsyncd

Запустите lsyncd:

systemctl start lsyncd

systemctl enable lsyncd

#### Первоначальная установка Keepalived

Примечание – Необходимо, чтобы серверы, на которых будет развернута служба keepalived, были в одном сегменте сети и чтобы в данном сегменте сети был хотя бы один незарезервированный ip-адрес, который будет выполнять роль плавающего ip-адреса.

Установите keepalived:

yum -y install keepalived

#### Конфигурирование Keepalived

Для конфигурирования Keepalived выполните следующие действия:

1. создайте пользователя keepalived:

useradd keepalived

1. удалите содержимое файла **/etc/keepalived/keepalived.conf** и откройте для редактирования на сервере, выполняющем роль master-сервера, и приведите к виду:

global\_defs {

vrrp\_garp\_master\_refresh 5

script\_user keepalived

enable\_script\_security

}

vrrp\_script chk\_nfs\_service {

script "/usr/sbin/rpcinfo -t ${LAN\_IP} nfs '4'"

timeout 1

interval 1

fall 3

rise 3

}

vrrp\_instance nfs\_failover {

state MASTER

interface ${LAN\_INTERFACE}

virtual\_router\_id 55

priority 255

virtual\_ipaddress {

${VIRTUAL\_IP}/32 dev ${LAN\_INTERFACE}

}

track\_interface {

${LAN\_INTERFACE}

}

track\_script {

chk\_nfs\_service

}

notify\_fault "/usr/bin/systemctl restart nfs-server" root

}

где:

* ${LAN\_IP} – локальный ip-адрес сервера, который будет использоваться для коммуникации с остальными серверами;
* ${LAN\_INTERFACE} – имя сетевого интерфейса, за которым закреплен ${LAN\_IP};
* ${VIRTUAL\_IP} – плавающий ip-адрес, незанятый ip-адрес в том же сегменте сети, который будет назначаться серверам, где будет использоваться служба keepalived. Плавающий ip-адрес един для экземпляров с одинаковым значением virtual\_router\_id.

1. удалите содержимое файла **/etc/keepalived/keepalived.conf** и откройте для редактирования на сервере, выполняющем роль backup-сервера, и приведите к виду:

global\_defs {

vrrp\_garp\_master\_refresh 5

script\_user keepalived

enable\_script\_security

}

vrrp\_script chk\_nfs\_service {

script "/usr/sbin/rpcinfo -t ${LAN\_IP} nfs '4'"

timeout 1

interval 1

fall 3

rise 3

}

vrrp\_instance nfs\_failover {

state BACKUP

interface ${LAN\_INTERFACE}

virtual\_router\_id 55

priority 100

virtual\_ipaddress {

${VIRTUAL\_IP}/32 dev ${LAN\_INTERFACE}

}

track\_interface {

${LAN\_INTERFACE}

}

track\_script {

chk\_nfs\_service

}

notify\_fault "/usr/bin/systemctl restart nfs-server" root

}

где:

* ${LAN\_IP} – локальный ip-адрес сервера, который будет использоваться для коммуникации с остальными серверами;
* ${LAN\_INTERFACE} – имя сетевого интерфейса, за которым закреплен ${LAN\_IP};
* ${VIRTUAL\_IP} – плавающий ip-адрес, незанятый ip-адрес в том же сегменте сети, который будет назначаться серверам, где будет использоваться служба keepalived. Плавающий ip-адрес един для экземпляров с одинаковым значением virtual\_router\_id.

1. выполните команду:

echo "net.ipv4.ip\_forward = 1" >> /etc/sysctl.conf

sysctl -p

1. примените правила для firewall:

firewall-cmd --direct --permanent --add-rule ipv4 filter INPUT 0 --in-interface ${LAN\_INTERFACE} --destination 224.0.0.18 --protocol vrrp -j ACCEPT

firewall-cmd --direct --permanent --add-rule ipv4 filter OUTPUT 0 --out-interface ${LAN\_INTERFACE} --destination 224.0.0.18 --protocol vrrp -j ACCEPT

firewall-cmd --reload

* где ${LAN\_INTERFACE} – наименование сетевого интерфейса для входящих и исходящих соединений.

#### Запуск Keepalived

Запустите keepalived:

systemctl start keepalived

systemctl enable keepalived

## Запуск Системы

Начало работы с Системой содержит следующую последовательность действий:

* запустите web-браузер двойным нажатием левой кнопки мыши по его ярлыку на рабочем столе или нажмите на кнопку «Пуск» и в открывшемся меню выберите пункт, соответствующий используемому web-браузеру;
* в адресной строке введите адрес Системы;
* в окне идентификации пользователя введите логин и пароль, выданные пользователю администратором Системы, нажмите на кнопку «Войти» (Рисунок 21).

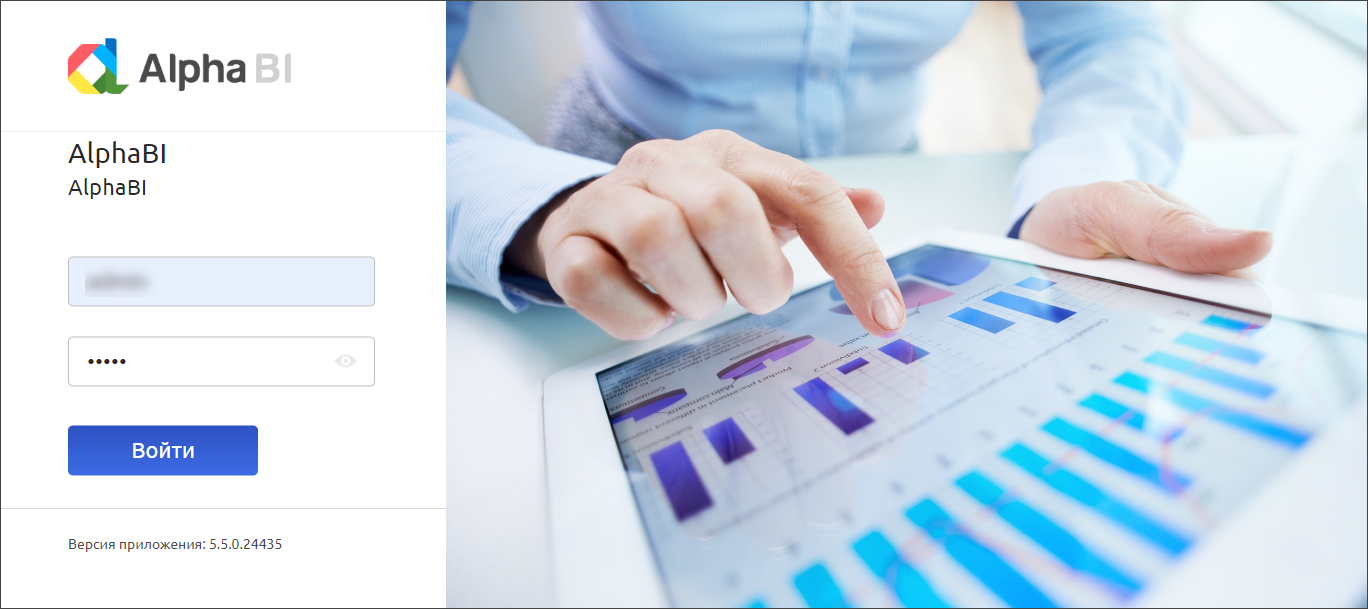


Рисунок  – Ввод идентификационных данных

После чего откроется главное окно Системы, отображающее (Рисунок 22):

* главное меню Системы (1), предназначенное для открытия разделов Системы. Панель главного меню будет отображаться на каждой странице Системы;
* краткую информацию о пользователе (ФИО) (2);
* кнопку «Выйти» (3), предназначенную для закрытия Системы и возвращения пользователя к окну идентификации;
* информационное полотно (4), которое может быть использовано для отображения аналитической панели, последних изменений платформы, на которой построена Система, и других сведений.



Рисунок  – Главное окно Системы

Примечание – Главное окно и главное меню Системы могут отличаться от приведенного выше рисунка и зависят от настроек Системы и прав доступа пользователей.

## Настройка Системы для авторизации и аутентификации

В Системе есть возможность настроить авторизацию через внешний сервис аутентификации по стандартам SAML, OpenID Connect, а также синхронизироваться со службой Active Directory.

Подробное описание настройки Системы для авторизации и аутентификации приведено разделе 5 документа «Платформа бизнес-аналитики «Alpha BI». Руководство пользователя. АБСМ.506100.001.И3».

## Порядок проверки работоспособности

Последовательность действий администратора при работе с Системой соответствует следующему алгоритму:

* запустите web-браузер, в случае возникновения ошибки проверьте Интернет-соединение;
* введите в адресной строке web-браузера ссылку для входа в контрольную панель Системы и авторизуйтесь (см. п. 5.1);
* произведите необходимые настройки в контрольной панели;
* просмотрите протокол ошибок;
* выполните выход из контрольной панели;
* войдите в Систему (см. п. 4.6);
* выполните выход из Системы.

### Просмотр и настройка логов на серверах приложений

#### Просмотр и настройка логов Alpha BI

##### Размещение лог-файлов и настройка уровней логирования

В процессе работы Alpha BI формируются журналы работы Системы, которые находятся в каталоге **/opt/alphabi/.logs/** на серверах приложений. На каждую дату создается отдельный лог-файл (с именем log-YYYY-MM-DD.json), что позволяет архивировать логи за предыдущие даты. По умолчанию логируются ошибки и важные сообщения.

Для настройки каталога выполните следующие действия:

1. для настройки альтернативного каталога хранения лог-файлов укажите соответствующий путь до каталога для ключа LogDirectory. По умолчанию (пустое значение) директория с логами будет указывать на .logs в рабочей директории приложения:

"LogConfig": {

"FileLogging": "info",

"DbLogging": "info",

// Путь до директории с логами. По умолчанию директория .logs в рабочей директории приложения

"LogDirectory": "/var/log/alphabi"

},

Создайте каталог и назначьте соответствующие права на него:

mkdir -p /var/log/alphabi

chown alphabi:adm /var/log/alphabi

1. для настройки уровня логирования отредактируйте файл **/opt/alphabi/alpha.config** на сервере приложения. В параметре LogConfig.FileLogging можно указать значение «debug», что приведет к записи более детальных логов (с уровнем логирования debug). Значения могут быть следующими: fatal, error, warn, info, debug, trace.

##### Структура лог-файлов

Файлы логов Alpha BI – это файлы в формате Newline-delimited JSON. Каждое сообщение журнала располагается в отдельной строке и записано в виде JSON-объекта:

{

"@ts": "2020-07-15T09:38:14.359682+03:00",

"@l": "Error",

"@src": "Bars.Alpha.UI.OlapClient.Mondrian.MondrianXmlaClient",

"@msg": "Не удалось выполнить XMLA-запрос к OLAP серверу",

"@exc": "System.Net.Http.HttpRequestException: An error occurred while sending the request. ---> System.IO.IOException: The server returned an invalid or unrecognized response.\n at System.Net.Http.HttpConnection.SendAsyncCore (System.Net.Http.HttpRequestMessage request, System.Threading.CancellationToken cancellationToken) [0x00cf5] in <c413547ad6b24c62b21f1da55944ca24>:0 \n --- End of inner exception stack trace ---\n at System.Net.Http.HttpConnection.SendAsyncCore (System.Net.Http.HttpRequestMessage request, System.Threading.CancellationToken cancellationToken) [0x012e5] in <c413547ad6b24c62b21f1da55944ca24>:0 \n at System.Net.Http.HttpConnectionPool.SendWithNtConnectionAuthAsync (System.Net.Http.HttpConnection connection, System.Net.Http.HttpRequestMessage request, System.Boolean doRequestAuth, System.Threading.CancellationToken cancellationToken) [0x0012b] in <c413547ad6b24c62b21f1da55944ca24>:0 \n at System.Net.Http.HttpConnectionPool.SendWithRetryAsync (System.Net.Http.HttpRequestMessage request, System.Boolean doRequestAuth, System.Threading.CancellationToken cancellationToken) [0x0014b] in <c413547ad6b24c62b21f1da55944ca24>:0 \n at System.Net.Http.RedirectHandler.SendAsync (System.Net.Http.HttpRequestMessage request, System.Threading.CancellationToken cancellationToken) [0x000ba] in <c413547ad6b24c62b21f1da55944ca24>:0 \n at Microsoft.Extensions.Http.Logging.LoggingHttpMessageHandler.SendAsync (System.Net.Http.HttpRequestMessage request, System.Threading.CancellationToken cancellationToken) [0x000a0] in <a1e13f54f7de43399f5ed249a523faff>:0 \n at Microsoft.Extensions.Http.Logging.LoggingScopeHttpMessageHandler.SendAsync (System.Net.Http.HttpRequestMessage request, System.Threading.CancellationToken cancellationToken) [0x000be] in <a1e13f54f7de43399f5ed249a523faff>:0 \n at System.Net.Http.HttpClient.FinishSendAsyncBuffered (System.Threading.Tasks.Task`1[TResult] sendTask, System.Net.Http.HttpRequestMessage request, System.Threading.CancellationTokenSource cts, System.Boolean disposeCts) [0x0017e] in <c413547ad6b24c62b21f1da55944ca24>:0 \n at Bars.Alpha.UI.OlapClient.Mondrian.MondrianXmlaClient.ExecuteXmla (Microsoft.AspNetCore.Http.HttpRequest request) [0x000a4] in <f809614959cc49b59c61a6b333696532>:0 ",

"@ctx": {

"cid": "1068b326",

"user": "admin",

"WebUserIp": "192.168.228.135",

"MvcController": "Xmla",

"MvcAction": "Http"

}

}

Структура записи лог-файла следующая:

* @ts – дата и время;
* @l – уровень события. Возможные значения:
  + «Fatal» – ошибки, приводящие к прекращению работы приложения;
  + «Error» – единичные ошибки, вызывающие сбой одной операции;
  + «Warn» – ошибки или подозрительные ситуации, не вызывающие сбой какой-либо операции;
  + «Info» – сообщения о нормальной работе приложения;
  + «Debug» – информация для отладки;
  + «Trace» – информация трассировки (более подробно, чем Debug).
* @src – источник записи лога (как правило, содержит имя класса, в котором произошло событие);
* @msg – текст или шаблон сообщения. Шаблон может содержать подстановочные места вида {value}, которые соответствуют значению соответствующего ключа JSON-объекта;
* @exc – детали исключительной ситуации;
* @ctx – контекст записи лога:
  + «cid» – id контекста;
  + «user» – имя пользователя;
  + «WebUserIp» – ip-адрес пользователя;
  + «MvcController» – имя web-контроллера;
  + «MvcAction» – имя действия web-контроллера.

#### Логирование Mondrian

##### Общая информация

По умолчанию в логи выводится только общая информация о работе Mondrian уровня INFO. Файлы логов находятся в том же каталоге, где и логи Tomcat (обычно по пути **/var/log/tomcat** или **/opt/tomcat/logs**).

Логирование Mondrian разделяется на три вида:

* общее функционирование Mondrian (логируется по умолчанию), файл mondrian.log;
* логирование MDX-запросов, файл mondrian-mdx.log;
* логирование SQL-запросов, файл mondrian-sql.log.

##### Настройка расширенного логирования

Примечание – Включение более подробного логирования приведет к увеличению занимаемого места файлами логов. Перед проведением настройки убедитесь в том, что свободного места в разделе достаточно.

В файлах со структурой XML строки и группы строк считаются закомментированными, если они начинаются с символов <!-- и заканчиваются символами -->. Если необходимо раскомментировать строку или группу строк, то удалите соответствующие символы, в которые они заключены.

Изменения, описанные в данном разделе, придется вносить вновь при обновлении Mondrian.

Все пути относительные, файл с настройками находится в **mondrian.war**. Работа с данным файлом описана в п. 4.2.2.

###### Изменение уровня логирования Mondrian

В файле **WEB-INF/classes/log4j.xml** найдите следующий блок:

<category name="mondrian">

<priority value="INFO"/>

</category>

Вместо INFO укажите необходимый уровень логирования, введя одно из следующих значений: FATAL, ERROR, WARN, INFO, DEBUG, TRACE.

###### Включение логирования MDX-запросов

В файле **WEB-INF/classes/log4j.xml** раскомментируйте следующий блок:

<!--

<category name="mondrian.mdx" additivity="false">

<priority value="DEBUG"/>

<appender-ref ref="MDXLOG"/>

</category>

-->

Примечание – В текущем примере уровень логирования MDX-запросов будет в режиме DEBUG. При необходимости укажите одно из следующих значений: FATAL, ERROR, WARN, INFO, DEBUG, TRACE.

###### Включение логирования SQL-запросов

В файле **WEB-INF/classes/log4j.xml** раскомментируйте следующий блок:

<!--

<category name="mondrian.mdx" additivity="false">

<priority value="DEBUG"/>

<appender-ref ref="SQLLOG"/>

</category>

-->

Примечание – В текущем примере уровень логирования SQL-запросов будет в режиме DEBUG. При необходимости укажите одно из следующих значений: FATAL, ERROR, WARN, INFO, DEBUG, TRACE.

###### Изменение размера и количества файлов логов Mondrian

Для каждого вида логирования можно отдельно настроить количество и размер сохраняемых файлов логов. Все настройки производятся в файле WEB-INF/classes/log4j.xml. Выполните следующие действия:

1. для изменения mondrian.log найдите следующий раздел:

<appender name="FILE" class="org.apache.log4j.RollingFileAppender">

<param name="File" value="${catalina.base}/logs/mondrian.log" />

<param name="MaxFileSize" value="50MB" />

<param name="MaxBackupIndex" value="10" />

<layout class="org.apache.log4j.PatternLayout">

<param name="ConversionPattern" value="%d %-5p [%c] %m%n"/>

</layout>

</appender>

Параметр MaxFileSize отвечает за максимальный размер файла, параметр MaxBackupIndex отвечает за количество файлов. Измените значения при необходимости.

1. для изменения mondrian-mdx.log найдите следующий раздел:

<appender name="MDXLOG" class="org.apache.log4j.RollingFileAppender">

<param name="File" value="${catalina.base}/logs/mondrian-mdx.log" />

<param name="MaxFileSize" value="50MB" />

<param name="MaxBackupIndex" value="10" />

<layout class="org.apache.log4j.PatternLayout">

<param name="ConversionPattern" value="%d %-5p [%c] %m%n"/>

</layout>

</appender>

Параметр MaxFileSize отвечает за максимальный размер файла, параметр MaxBackupIndex отвечает за количество файлов. Измените значения при необходимости.

1. для изменения mondrian-sql.log найдите следующий раздел:

<appender name="SQLLOG" class="org.apache.log4j.RollingFileAppender">

<param name="File" value="${catalina.base}/logs/mondrian-sql.log" />

<param name="MaxFileSize" value="50MB" />

<param name="MaxBackupIndex" value="10" />

<layout class="org.apache.log4j.PatternLayout">

<param name="ConversionPattern" value="%d %-5p [%c] %m%n"/>

</layout>

</appender>

Параметр MaxFileSize отвечает за максимальный размер файла, параметр MaxBackupIndex отвечает за количество файлов. Измените значения при необходимости.

#### Логирование PostgreSQL

##### Общая информация

По умолчанию в логах PostgreSQL фиксируются только ошибки. Порой этого бывает недостаточно для анализа работы Alpha BI с целью выявления причин медленной работы процессов.

##### Настройка расширенного логирования

Примечание – Существует несколько способов конфигурирования PostgreSQL. В данном разделе рассматривается вариант внесения изменений в основной конфигурационный файл PostgreSQL 15 в Ubuntu 22.04, находящийся по пути **/etc/postgresql/15/main/postgresql.conf**. После внесения изменений в файл конфигурации перезапустите PostgreSQL.

###### Включение логирования подключений, выполнения контрольных точек, блокировок

Для включения логирования подключений, выполнения контрольных точек, блокировок укажите следующие значения параметров:

log\_checkpoints = on

log\_connections = on

log\_disconnections = on

log\_lock\_waits = on

###### Включение логирования длительности запросов и создания временных файлов

Для включения логирования длительности запросов и создания временных файлов измените следующие параметры:

log\_min\_duration\_statement = 1000

log\_temp\_files = 10MB

Примечание – Указанные значения означают, что будут логироваться все запросы, длительность выполнения которых превышает 1 секунду, а также зафиксированы факты создания временных файлов размером более 10 МБ.

###### Использование pgbadger для анализа логов PostgreSQL

Примечание – Не рекомендуется запускать pgbadger на сервере БД в промышленной среде. Следует копировать файлы логов на отдельный сервер, где уже генерировать отчеты. В примере, описанном ниже, отчеты собираются на сервере БД.

Для использования pgbadger для анализа логов PostgreSQL выполните следующие действия:

1. предварительно измените следующие параметры:

lc\_messages = 'C'

log\_line\_prefix = '%t [%p]: '

1. установите pgbadger и создайте каталог для результатов его работы:

apt update

apt -y install pgbadger

mkdir /tmp/pgbadger

Пример – Пример команды pgbadger для генерации отчета:

/usr/bin/pgbadger --prefix '%t [%p]: ' -q -X --nocomment --incremental --last-parsed /tmp/pgbadger/database.last --top 100 --title database /var/log/postgresql/postgresql-15-main.log --pid-dir /tmp/pgbadger/ -O /tmp/pgbadger/

Указанная команда сгенерирует отчет с основными показателями работы БД.

### Просмотр и настройка логов в Системе

#### Просмотр лога действий

Для перехода к просмотру лога действий нажмите на кнопку «DBLOG» в главном меню Системы, если эта кнопка настроена. Если нет, то для ее настройки выполните следующую последовательность действий:

1. нажмите в главном меню Системы кнопку «Система»;
2. во вкладке «Система» перейдите в раздел «Администрирование»;
3. выберите пункт «Главное меню системы»;
4. включите режим редактирования;
5. нажмите на кнопку «Добавить» и выберите пункт «Пустой пункт меню»;
6. выделите добавленный пункт, укажите для него наименование «DBLOG»;
7. в поле «Действие» откройте встроенный справочник и нажмите на кнопку «Создать». Откроется окно создания действия. Выберите тип действия «Просмотр данных», присвойте создаваемому действию наименование «Просмотр данных», в выпадающем списке поля «Метаданные» выберите значение «Лог действий (dblog)», нажмите на кнопку «Сохранить и закрыть»;
8. выберите созданное действие для добавляемого пункта главного меню Системы;
9. выйдите из режима редактирования, сохранив настройки по созданию пункта главного меню Системы (Рисунок 23);
10. обновите Систему нажатием на клавишу <F5>. Добавленный пункт отобразится в главном меню Системы.

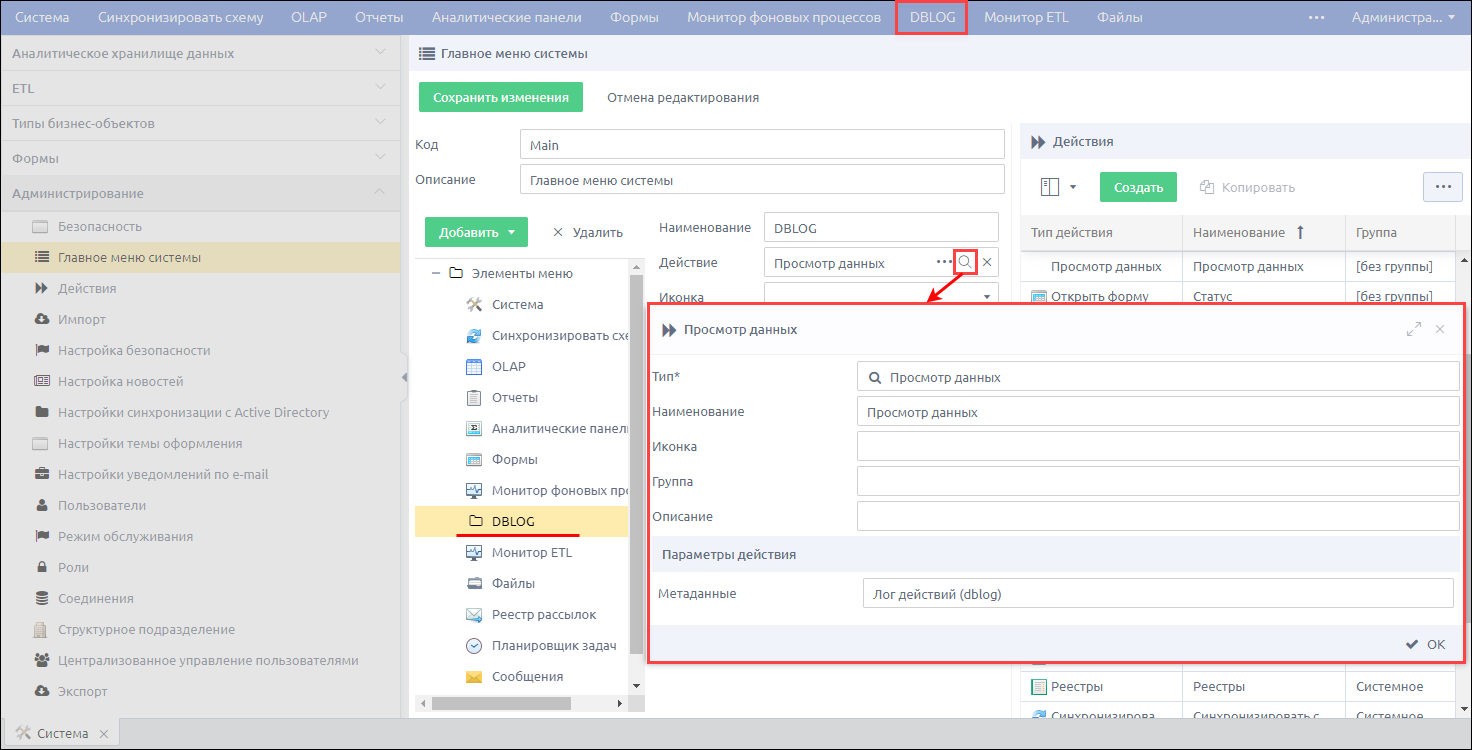


Рисунок  – Настройки пункта «DBLOG» главного меню Системы

1. нажмите на пункт меню «DBLOG» (Рисунок 24), откроется окно, содержащие следующую информацию:

* дата и время ошибки;
* логин пользователя;
* IP пользователя;
* Host;
* контроллер;
* действие контроллера;
* текст (описывающий действие);
* текст ошибки (стек ошибки);
* тип лога:
  + Debug;
  + Info (информационный лог);
  + Warning (предупреждение);
  + Error (ошибка);
  + Fatal (фатальная ошибка).
* уникальный идентификатор потока.

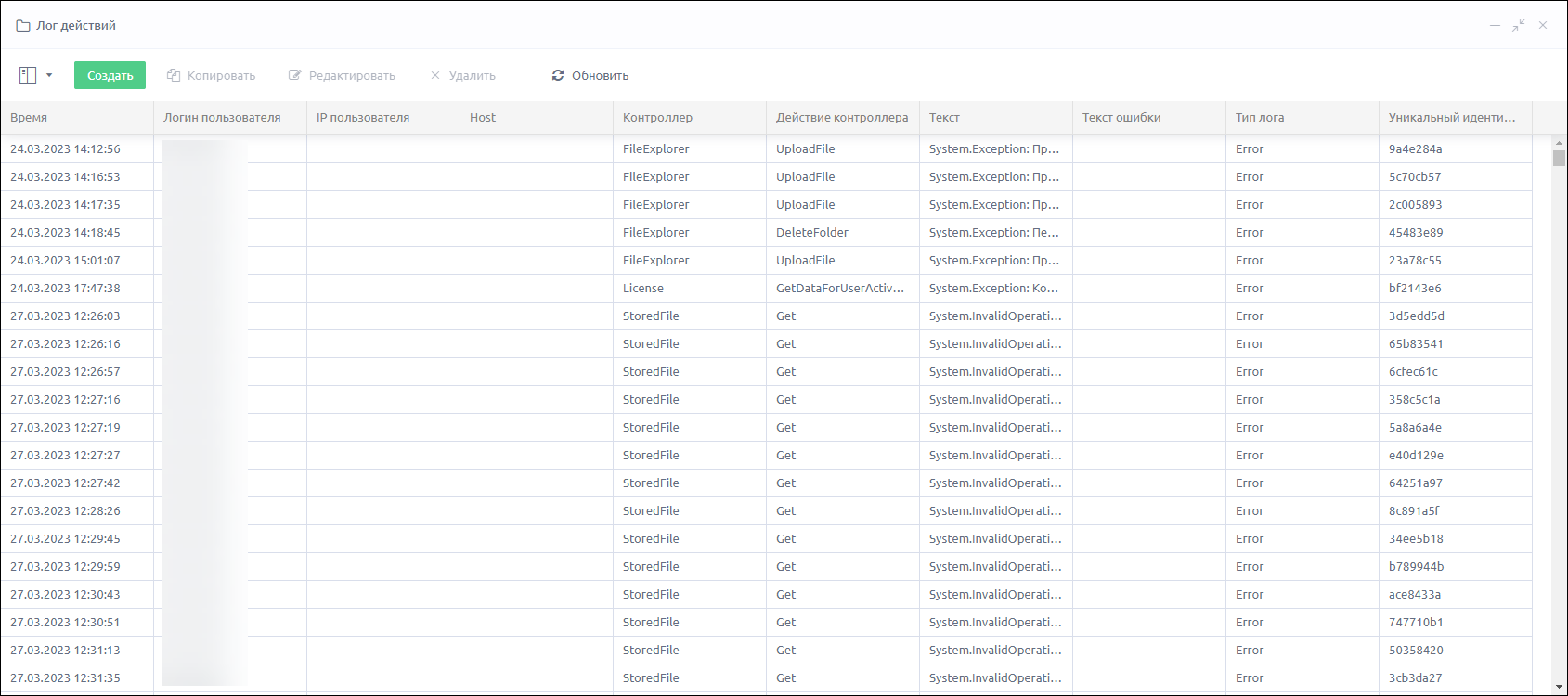


Рисунок  – Окно «Лог действий»

1. для детального просмотра лога дважды нажмите на строку, после чего откроется окно (Рисунок 25).

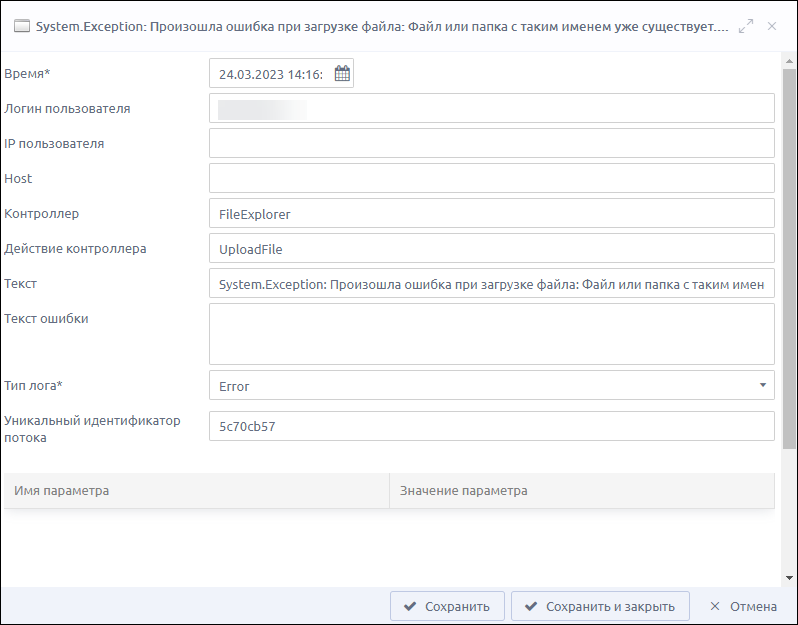


Рисунок  – Просмотр лога действий

#### Просмотр лога авторизации

Логи авторизации хранятся в служебных метаданных «security\_sessionopenclosefact» в БД, управляемой СУБД PostgreSQL.

Для просмотра лога авторизации создайте БО с типом «SQL-запрос». В поле «Текст SQL запроса» введите значение «select \* from runtime.security\_sessionopenclosefact». Создайте поля автозаполнением из полей SQL-запроса. Опубликуйте БО и синхронизируйте схему. Подробное описание создания БО приведено в разделе 4 документа «Платформа бизнес-аналитики «Alpha BI». Руководство пользователя. АБСМ.506100.001.И3».

Нажмите на кнопку «Просмотр данных» в БО. Откроется окно, содержащее следующую информацию (Рисунок 26):

* логин пользователя;
* имя пользователя;
* тип: вход/выход;
* дата и время.

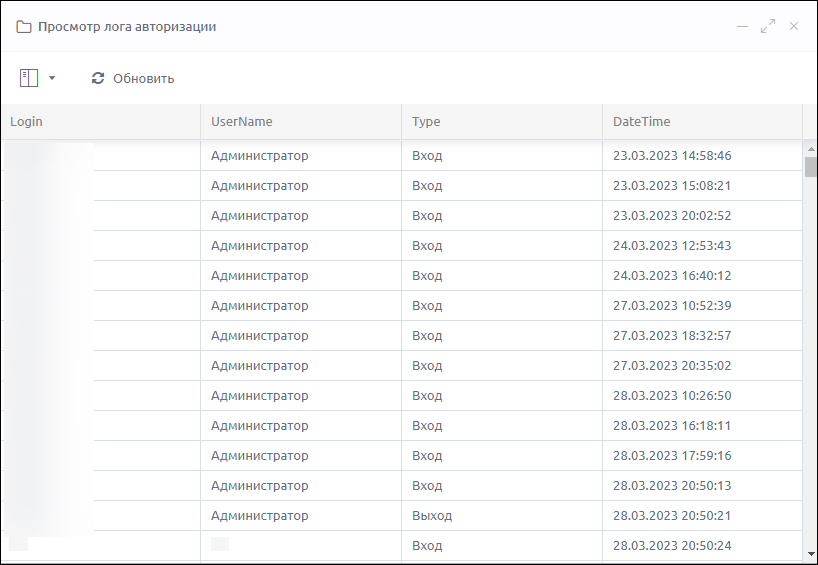


Рисунок  – Логирование авторизации

#### Монитор сессий пользователей

В Системе реализована возможность просмотра информации об активных на текущий момент пользовательских сессиях. Для открытия монитора сессий нажмите на кнопку «Монитор сессий» в главном меню Системы (Рисунок 27). Если данной кнопки в главном меню нет, настройте ее, привязав к ней действие «Монитор сессий» (Рисунок 28) (процедура настройки главного меню Системы подробно описана в разделе 5 документа «Платформа бизнес-аналитики «Alpha BI». Руководство пользователя. АБСМ.506100.001.И3»).



Рисунок  – Выбор пункта «Монитор сессий»

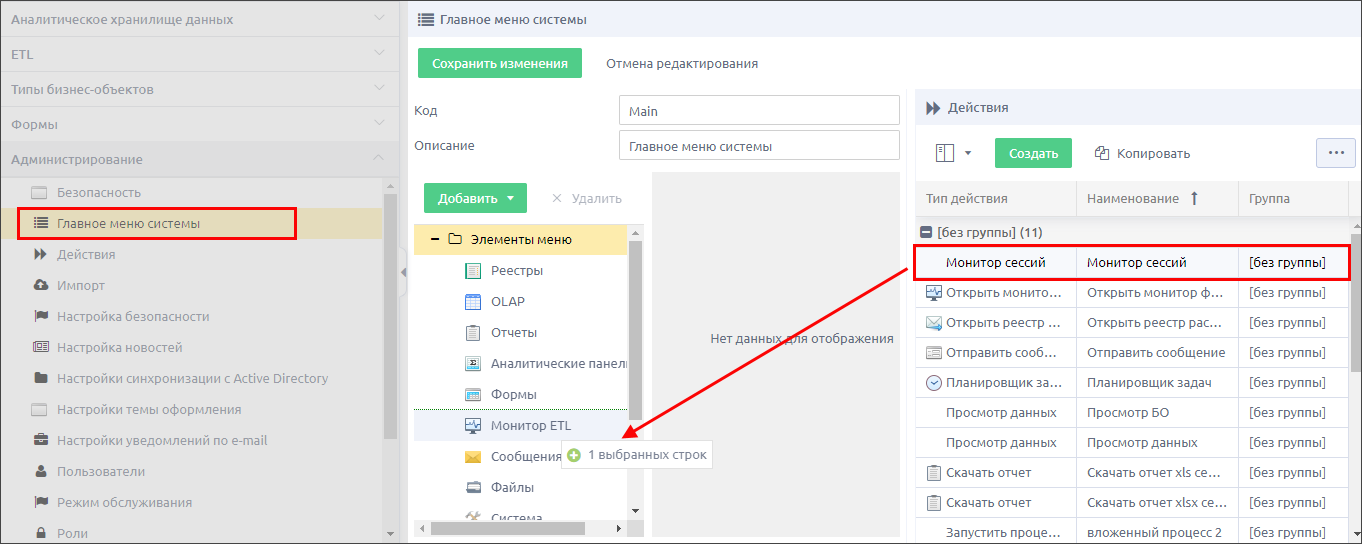


Рисунок  – Добавление пункта в главное меню Системы

Откроется окно «Монитор сессий», отображающее следующую информацию об активных на текущий момент сессиях (Рисунок 29):

* «Пользователь» – логин пользователя, инициировавшего сессию;
* «IP пользователя» – IP пользователя, инициировавшего сессию;
* «Создана» – дата начала сессии;
* «Истекает» – дата окончания сессии;
* «Длительность» – длительность активности сессии;
* «Таймаут простоя (мин.)» – временной период (в минутах), по истечении которого сессия удалится, если пользователь не станет активным.

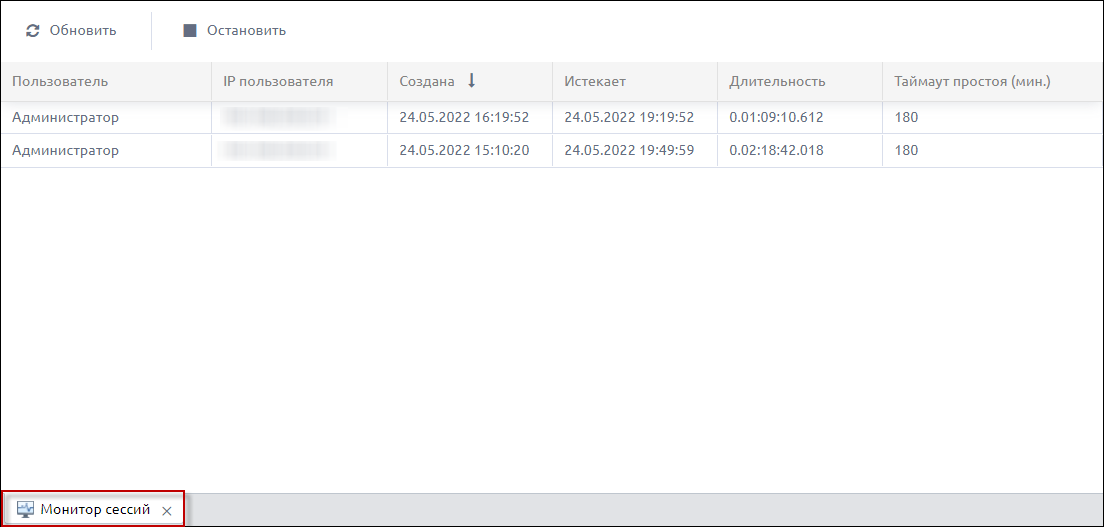


Рисунок  – Окно «Монитор сессий»

Для обновления отображаемой информации нажмите на кнопку «Обновить». Чтобы остановить сессию пользователя, выберите необходимую запись в таблице и нажмите на кнопку «Остановить».

Пользователю будет доступен просмотр монитора сессий всех пользователей, если ему даны права доступа к функции управления пользователями и к просмотру и выполнению действия «Монитор сессий» (Рисунок 30). Подробнее процессы наделения пользователей/ролей правами доступа в Системе описаны в разделе 4 документа «Платформа бизнес-аналитики «Alpha BI». Руководство пользователя. АБСМ.506100.001.И3».

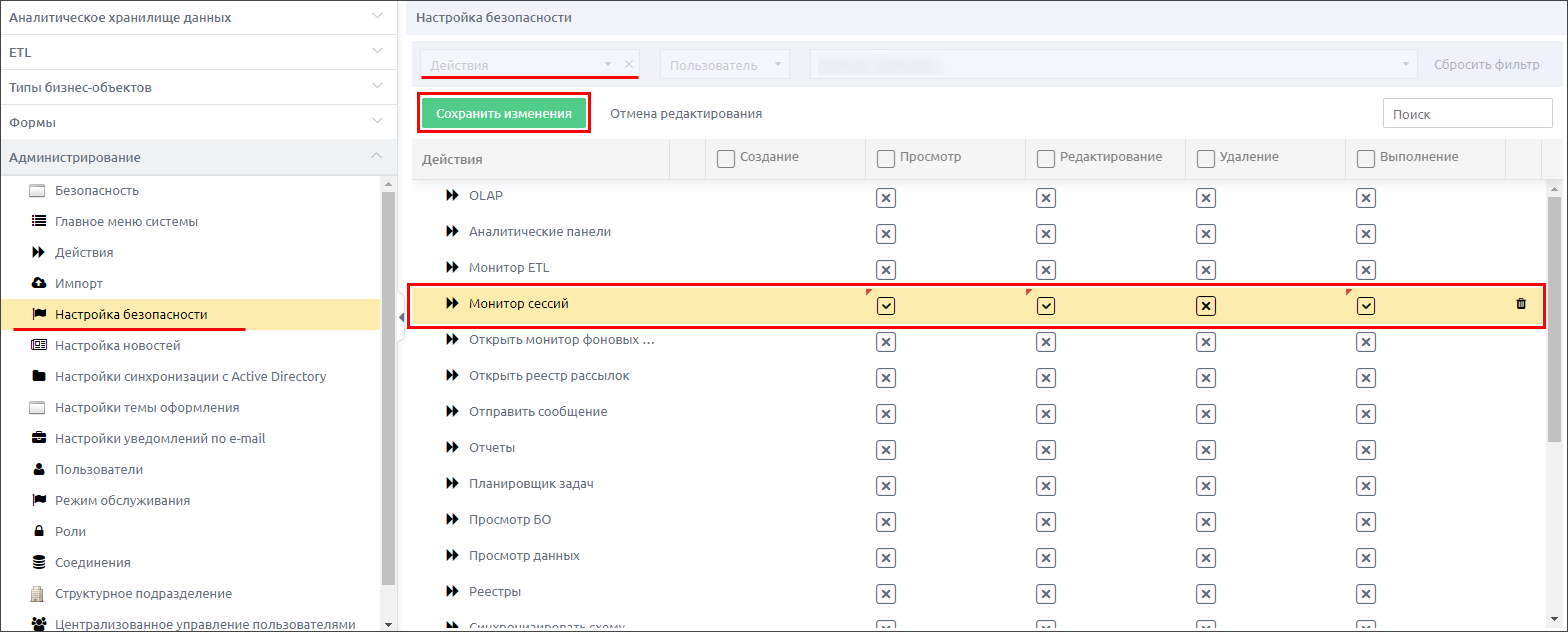


Рисунок  – Определение прав доступа роли/пользователя к просмотру информации о сессиях всех пользователей Системы

# Описание операций

## Работа в контрольной панели

Для входа в контрольную панель:

1. введите в адресной строке web-браузера ссылку на Систему;
2. допишите на английской раскладке буквы «cc» (Рисунок 31);

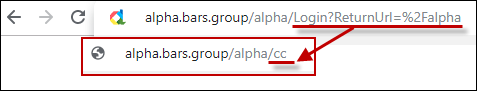


Рисунок  – Переход в контрольную панель

1. введите пароль из соответствующей секции файла «alpha.config» (Рисунок 32):

" ControlCenterConfig ": { " Password ": "123" },



Рисунок  – Окно авторизации

Если пароль не задан, то вход в центр управления невозможен.

После этого откроется главное окно контрольной панели (Рисунок 33).

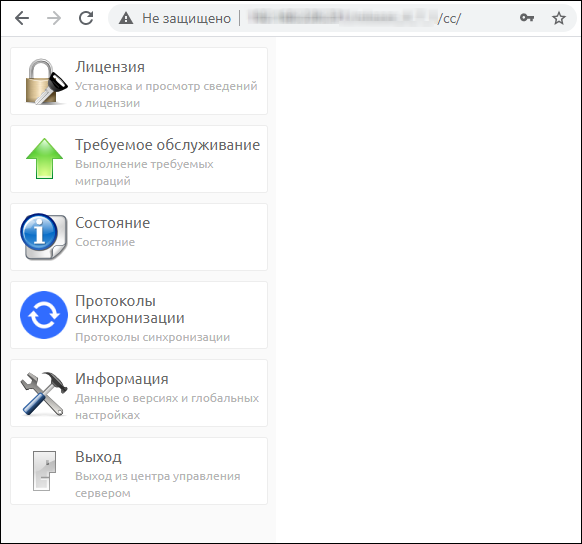


Рисунок  – Контрольная панель

### Лицензия

В данном разделе осуществляется установка и просмотр сведений о лицензии (Рисунок 34).

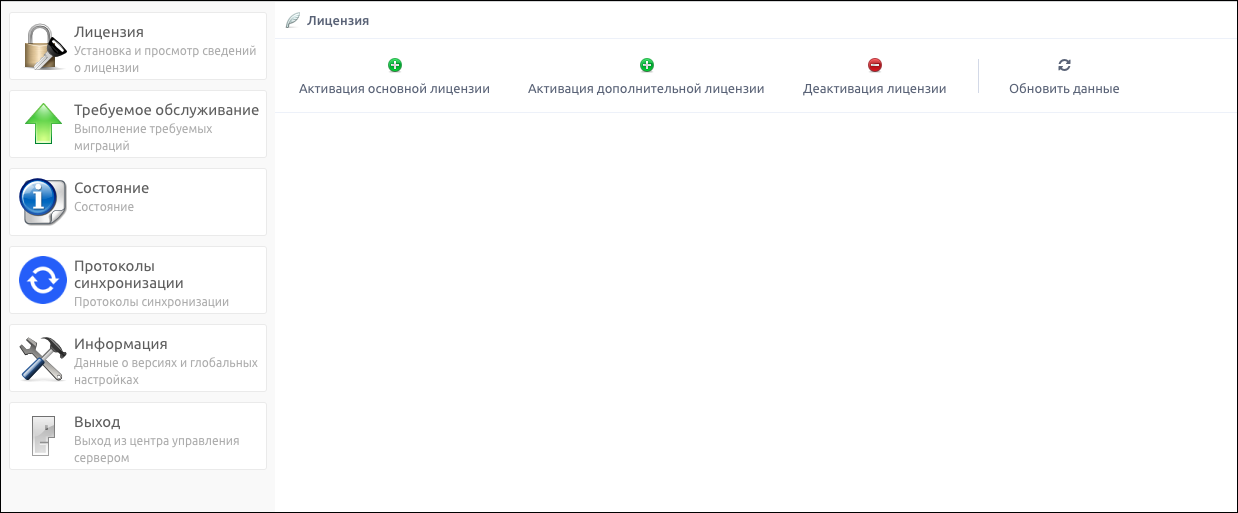


Рисунок  – Окно «Лицензия»

#### Объекты лицензии

Объектами заключения лицензионного договора являются:

* число одновременно созданных учетных записей пользователей;
* серверное программное обеспечение в составе базовых модулей;
* дополнительные модули.

#### Виды лицензий в Системе

Предусмотрены следующие виды лицензий:

* коммерческая – устанавливается на «боевые» приложения. После завершения срока действия лицензии приложение остается доступным для работы пользователей, однако блокируется возможность провести обновление. Данная лицензия выписывается сотрудниками коммерческого блока:
  + «Основная» – выдается на платформу и определяет срок действия лицензии, модули, входящие в состав платформы, и первоначальное количество пользователей Системы. Такая лицензия может единовременно быть активирована только на одном приложении, на одном сервере. Для повторной активации лицензии, например, в случае изменения ЦОД, требуется провести процедуру деактивации;
  + «Дополнительная» – позволяет закупить дополнительные рабочие места пользователей для последующей активации на разных приложениях. Дополнительную лицензию невозможно активировать на приложении без основной лицензии. Такая лицензия может быть активирована частично, например, 5 пользователей из 10 возможных, и единовременно на нескольких приложениях, например, 5 пользователей на одном приложении и 5 пользователей на другом. Лицензию можно активировать до тех пор, пока не будет исчерпан лимит лицензии. В результате активации дополнительной лицензии число возможных пользователей Системы увеличивается. Деактивация на приложении, где были активированы и основная, и дополнительные лицензии единая. При повторной активации такой деактивированной лицензии количество пользователей суммируется.
* демонстрационная (тестовая) – устанавливается только на тестовые серверы АО «БАРС Груп». Приложение доступно только на период действия лицензии. По истечении указанного срока приложение блокируется и необходимо обновление лицензии. Выписывается менеджером продукта Alpha BI.

#### Ключи

Лицензионный ключ (далее – ключ) – техническое средство защиты авторских прав Правообладателя (ст. 1299 ГК РФ), которое устанавливается в Системе и дает доступ к ее функциям и обновлениям Системы.

Каждый ключ является именным и выписывается на имя покупателя или наименование организации покупателя.

Каждый ключ подлежит обязательной активации, которая происходит по следующему алгоритму:

* клиент устанавливает ключ;
* Система генерирует код активации;
* клиент сообщает код активации менеджеру по продажам;
* менеджер по продажам с помощью кода активации генерирует ответный код активации и сообщает его клиенту;
* клиент вводит в Систему ответный код активации;
* Система активирована.

Описание порядка активации основной и дополнительной лицензий в Системе приведено в п. 5.1.1.4 и 5.1.1.5 соответственно.

#### Описание порядка активации основной лицензии

Для активации лицензии выполните следующие действия:

1. нажмите на кнопку «Активация основной лицензии»;
2. в открывшемся окне введите серийный номер продукта (Рисунок 35), полученный от представителей АО «БАРС Груп»;

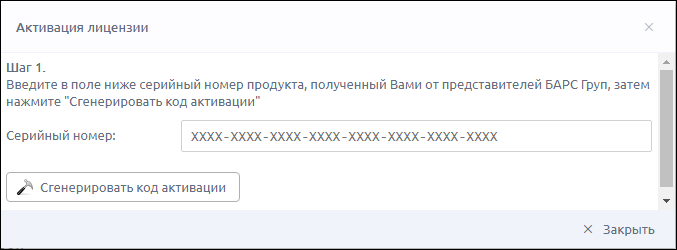


Рисунок  – Окно «Активация основной лицензии», шаг 1

1. нажмите на кнопку «Сгенерировать код активации» для генерации кода активации;
2. передайте сгенерированный код представителям АО «БАРС Груп»;
3. получите ключ активации и введите его в поле «Код активации» (Рисунок 36).

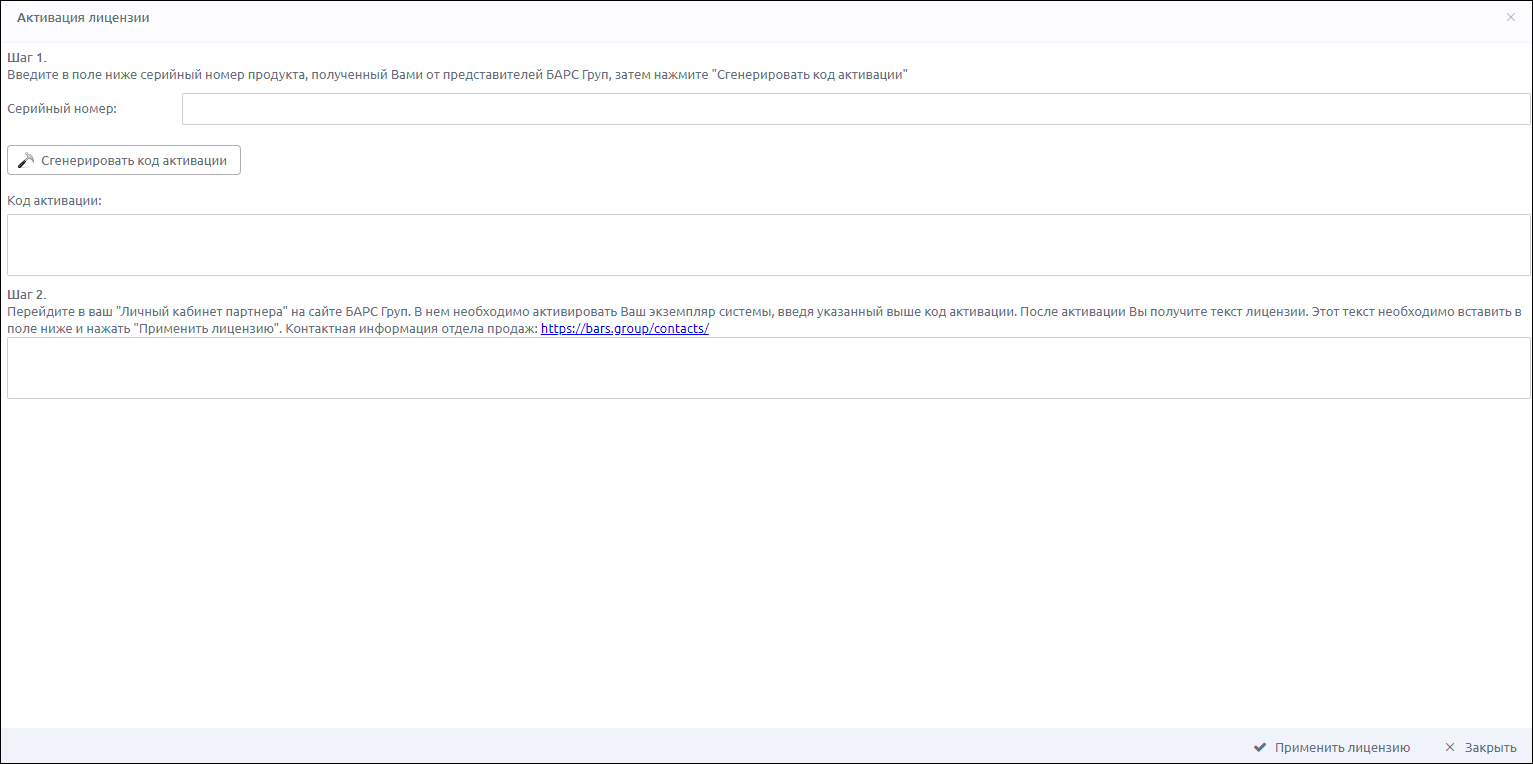


Рисунок  – Окно «Активация основной лицензии», шаг 2

#### Описание порядка активации дополнительной лицензии

Для активации дополнительной лицензии выполните следующие действия:

1. нажмите на кнопку «Активация дополнительной лицензии»;
2. в открывшемся окне введите серийный номер продукта (Рисунок 37), полученный от представителей АО «БАРС Груп»;

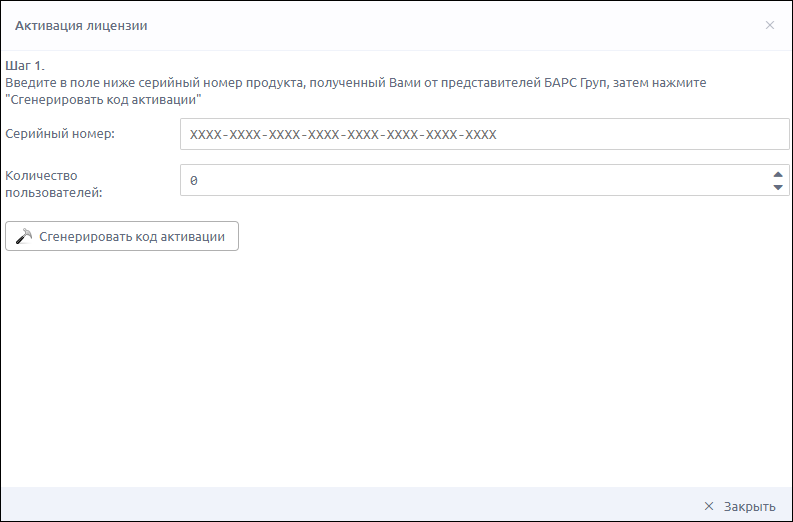


Рисунок  – Окно «Активация дополнительной лицензии»

1. в поле «Количество пользователей» введите количество пользователей, которое требуется дополнительно активировать на данном приложении (введенное значение должно быть больше 0);
2. нажмите на кнопку «Сгенерировать код активации» для генерации кода активации;
3. передайте сгенерированный код представителям АО «БАРС Груп»;
4. если количество запрашиваемых пользователей доступно к активации, то получите ключ активации и введите его в поле «Код активации».

#### Виды ключей

В Системе предусмотрены два вида ключей:

* ключ активации, дающий возможность активировать Систему, использовать ее функции (бессрочно) и получать обновления (в течение одного года);
* ключ обновления, дающий возможность обновить Систему на второй и последующие годы ее использования (имеет срок действия один год, шесть месяцев или три месяца и покупается в рамках гарантированной поддержки).

#### Состав программного продукта

Программный продукт состоит из стандартных модулей (которые входят стоимость лицензии) и дополнительных модулей (которые покупаются отдельно).

Состав базовых модулей:

* модуль реестров;
* модуль «Аналитическое хранилище данных»;
* модуль OLAP-анализа;
* модуль «Формы вывода»;
* модуль отчетов;
* модуль визуализации;
* модуль файлового хранилища;
* модуль интеграции и обработки данных;
* модуль сообщений;
* модуль администрирования;
* модуль экспорта-импорта настроек экземпляра Системы;
* модуль «Центр управления сервером».

#### Коды ключей

Коды ключей приведены в таблице ниже (Таблица 4).

Таблица  – Коды ключей

| Код модуля | Модуль | Примечание |
| --- | --- | --- |
| ETL | Модуль интеграции и обработки данных |  |
| DWH | Модуль «Аналитическое хранилище данных», подсистема OLAP-анализа |  |
| DASHBOARDS | Модуль визуализации |  |
| REPORTS | Модуль отчетов |  |
| FORMS | Модуль «Формы вывода» |  |

#### Форма и состав стандартной поставки программного продукта

Программное обеспечение поставляется в виде дистрибутива на электронном цифровом носителе либо путем предоставления доступа к скачиванию дистрибутива.

Стандартная поставка программного продукта включает:

* одну лицензию на серверное ПО;
* пять лицензий пользователей. Каждая лицензия пользователя дает возможность доступа ко всем функциям Системы.

#### Обновление ПО

Обновление Системы происходит в ручном режиме с помощью установки патча и ключа обновления (за исключением первого периода использования).

#### Прочие условия

При нехватке числа учетных записей пользователей, входящих в состав стандартной поставки, возможно приобретение дополнительных учетных записей в соответствии с Прайс-листом.

Приобретение дополнительных учетных записей осуществляется путем перевыписки ключа активации через менеджера по продажам. Лицензии администраторов входят в число пользовательских лицензий стандартной поставки.

Под числом созданных учетных записей пользователей понимается лимит пользователей и запрет на добавочное число учетных записей пользователей Системы.

Приобретение дополнительных модулей осуществляется путем перевыписки ключа активации через менеджера по продажам.

### Требуемое обслуживание

После обновления и ввода ключа активации Системы иногда требуется миграция. Для проведения обслуживания в разделе «Требуемое обслуживание» нажмите на кнопку «Провести обслуживание» (Рисунок 38).

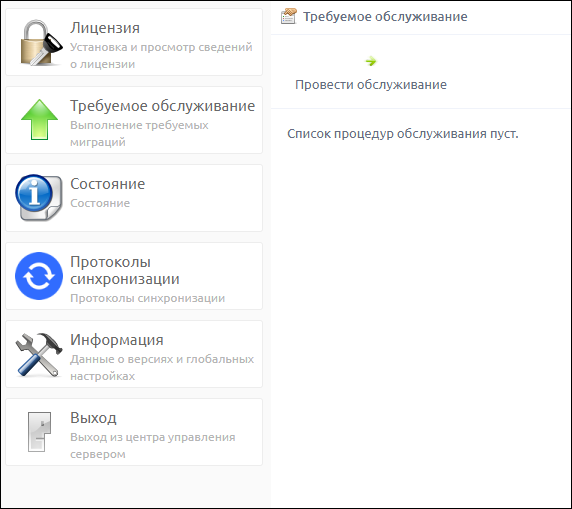


Рисунок  – Раздел «Требуемое обслуживание»

Результат выполнения обслуживания отобразится в окне.

В случае выполнения обслуживания с ошибкой отобразится описание причины ее появления (Рисунок 39). При возникновении ошибки обратитесь в службу технической поддержки.

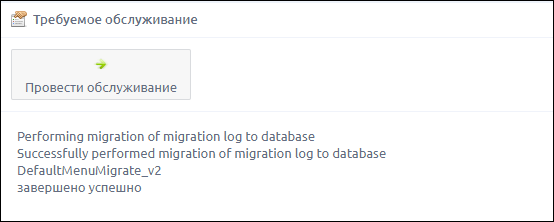


Рисунок  – Описание ошибки миграции

### Состояние

В разделе «Состояние» отображается актуальная информация о состоянии кластера (Рисунок 40).

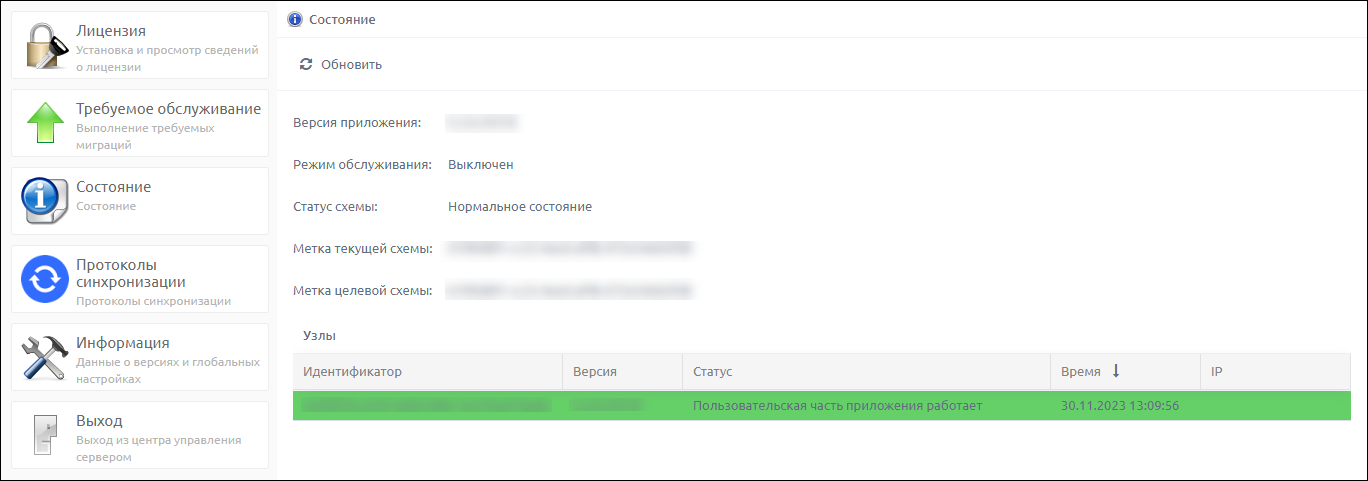


Рисунок  – Раздел «Состояние»

### Протоколы синхронизации

Для просмотра протоколов синхронизации схемы перейдите в раздел «Протоколы синхронизации» (Рисунок 41).

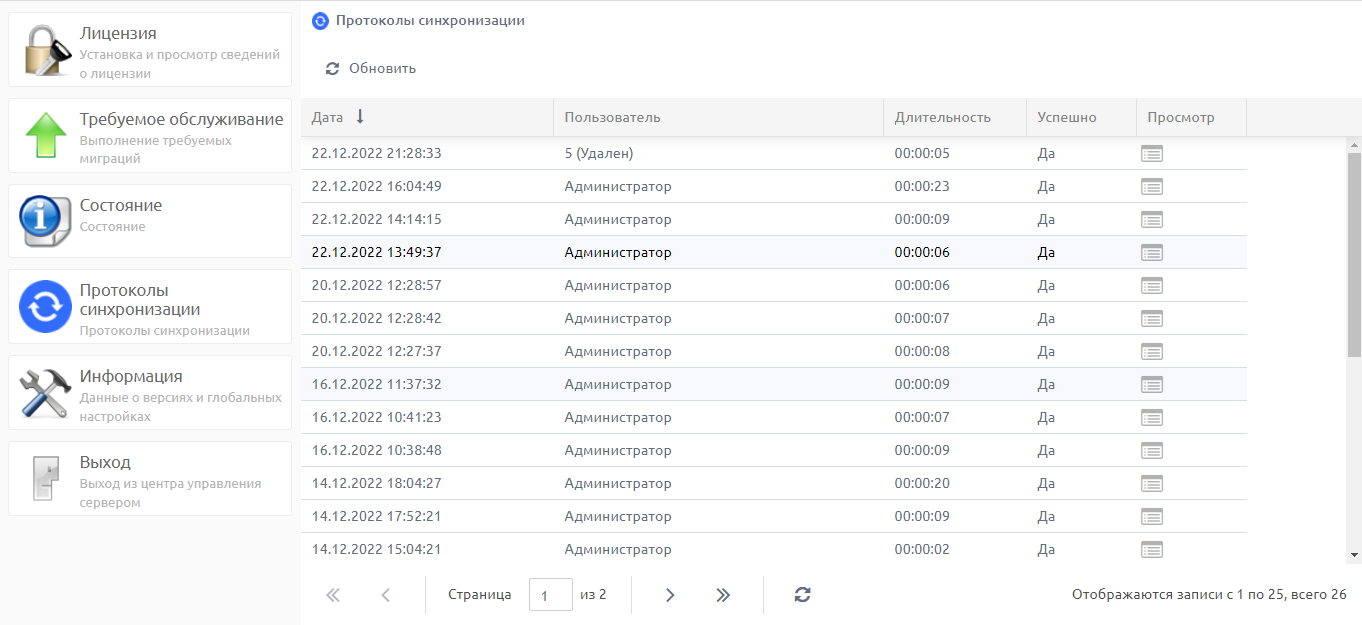


Рисунок  – Раздел «Протоколы синхронизации»

В данном разделе отображается следующая информация о проведенных синхронизациях схемы:

* дата синхронизации;
* имя пользователя, выполнившего синхронизацию;

Примечание – Наименование пользователя, который выполнил синхронизацию схемы и был удален из Системы, отображается с постфиксом «(Удален)» (см. Рисунок 41).

* длительность выполнения синхронизации;
* результат синхронизации.

Для просмотра протокола синхронизации нажмите на кнопку  в столбце «Просмотр» (Рисунок 42).

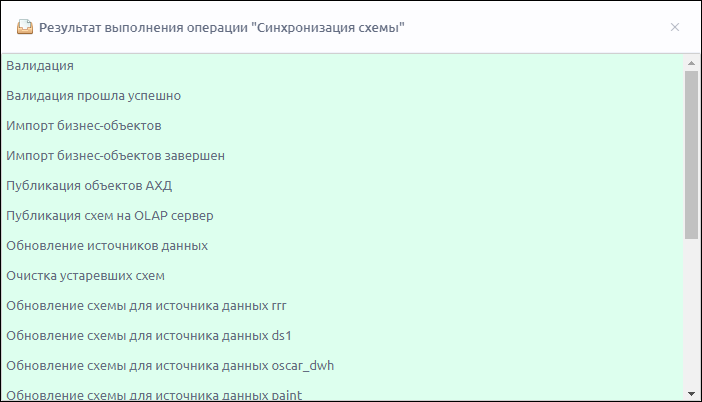


Рисунок  – Результат выполнения операции «Синхронизация схемы»

### Информация

В разделе «Информация» отображается информация о версиях и глобальных настройках приложения (Рисунок 43).

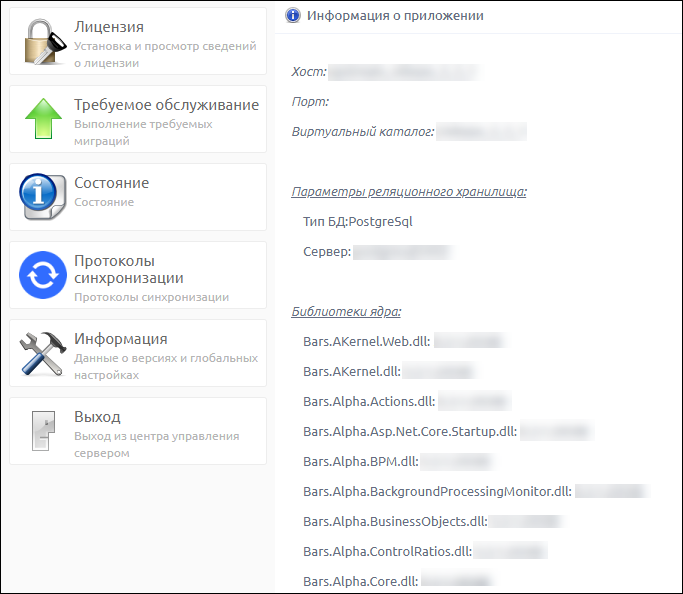


Рисунок  – Раздел «Информация»

### Выход

При выборе пункта «Выход» откроется окно для подтверждения выхода из центра управления сервером. Нажмите на кнопку «Да», чтобы выйти, или на кнопку «Нет» для отмены выхода.

# Аварийные ситуации

## Нештатные ситуации

Возникающие при работе с Системой нештатные ситуации и способы их решения описаны в таблице ниже (Таблица 5).

Таблица  – Описание аварийных ситуаций

| Сообщение/ название ошибки | Действия администратора/ способы устранения |
| --- | --- |
| Страница не найдена (Рисунок 44, Рисунок 45) | Проверьте правильность ввода ссылки в адресной строке web-браузера |
| Ошибка авторизации. Пользователь с указанным логином не существует (Рисунок 46) | В окне идентификации пользователя заново заполните поля «Логин» и «Пароль», предварительно проверив, не включена ли клавиша <Caps Lock>, и правильность выбора раскладки языка |
| Остановлена служба БД (Рисунок 47) | Изучите логи приложения Alpha BI.  Изучите логи службы БД |
| Ошибка при запуске. Приложение временно недоступно (Рисунок 48) | Перейдите в контрольную панель. Проверьте статус схемы в разделе «Состояние», проверьте наличие актуальной лицензии в разделе «Лицензии», изучите логи приложения Alpha BI |
| Сообщение «Время сессии истекло» | Нажмите на клавишу <F5>, при необходимости авторизуйтесь. Для пула приложений, который обрабатывает запросы, должен быть указан один рабочий процесс, так как web-формы на базе первого пула не имеют общей памяти для сессий |
| Увеличение времени открытия страниц | Остановите процесс, занимающий большую часть времени процессора.  Остановите/запустите пул приложения в IIS |
| OutOfMemoryException | При появлении такой ошибки на компьютере с операционной системой 64бит (клиент, на котором проводится сборка) убедитесь, что процесс запущен в 64битной среде, т.е. в запущенных процессах не отображается x32bit. Появление такой ошибки в x64 среде может свидетельствовать о том, что недостает виртуальной памяти вследствие неправильно настроенного SWAP-файла |
| Некорректное отображение стартовой страницы (отображаются только поля ввода логина/пароля, без изображений. После авторизации откроется пустое окно) | Проверьте, включен ли в IIS статический контент, был ли он выбран при установке IIS |
| Постоянный разрыв сессии | Настройте служебный перезапуск пула IIS на время наименьшей активности пользователей. На рисунках (Рисунок 49, Рисунок 50) показаны настройки для 6 и 7 версии IIS |
| Ошибка при работе с блоком «Заполнение Excel файла»:  «Unable to cast object of type ‘System.Int32’ to type ‘System.String’» | Выполните очистку и автозаполнение блока. Типы записываемых данных не совпадают с типом данных на входе в блок |
| Ошибка при работе с блоком «Чтение файла из файлового хранилища»:  «Exception of type ‘System.OutOfMemoryException’ was thrown» | Проверьте наличие файла и возможность обращения к нему через файловое хранилище |
| Ошибка при работе с блоком «Excel (\*.xlsx)»:  «A disk error occurred during a write operation. (Exception from HRESULT: 0x8003001D (STG\_E\_WRITEFAULT))» | Проверьте доступность и наличие свободного места на диске. Если проблем с диском нет, перезапустите инстанс на сервере |
| Ошибка при работе с блоком «Создание бизнес-объектов»:  «неверный порядок полей Строка данных:.» | Перезаполнение хранилища может решить проблему |
| Ошибка при работе с блоком «Калькулятор»:  «The given key was not present in the dictionary» | Проверьте все коды обращения к полям через текстовое отображение – поля данных.  Проверьте, что синтаксис обращения к переменной или полю верный «@код\_обращения\_к переменной» |
| Невозможно добавить новые поля в бизнес-объект. При сохранении метаописания бизнес-объекта возникает ошибка 404 (Not Found) | Длина запроса превысила заданное число maxAllowedContentLength в «WebConfig». Для устранения ошибки увеличьте значения maxAllowedContentLength в файле «Web.config» развернутого приложения, например, до значения 1048576000. Это означает, что можно отправлять запрос на сервер размером до 1 ГБ |
| При попытке зайти в Систему до окна авторизации появляется ошибка: Request is not available in this context | Проверьте наличие свободной оперативной памяти и свободного места на диске |
| При выгрузке шаблона формы из БО возникает сообщение об исключительной ситуации сервера: The type initializer for 'Gdip' threw an exception (Рисунок 51), и форма не выгружается | Для устранения ошибки установите пакет libgdiplus на сервер с приложением Alpha BI и перезапустите приложение Alpha BI |
| Ошибка при синхронизации схемы или выполнении запроса к OLAP-серверу:   * Ошибка обновления схемы: Не удалось выполнить запрос к OLAP серверу: Response status code does not indicate success: 500 ()…   или   * Ошибка при загрузке схемы: The Mondrian XML: Connection refused | 1. Если синхронизация схемы проводится впервые с момента развертывания: убедитесь, что настройки для сервера приложений в файле alpha.config и в файле web.xml для OLAP-сервера корректные.  1.1. На сервере приложения Alpha BI в файле конфигурации alpha.config в секции OlapConfig проверьте значения параметров:   * XmlaEndpoint – URL для подключения к Mondrian; * ApplicationBaseUrl – URL приложения Alpha BI, по которому оно доступно для Mondrian.   1.2. На сервере OLAP в файле Mondrian web.xml проверьте значения параметров:   * DataSourcesConfig; * AlphaApiKey.   2. Если синхронизация схемы ранее проводилась успешно, но перестала работать, то проанализируйте логи Tomcat |
| При настройке соединения с СУБД Oracle проверка подключения завершается ошибкой:  ORA-00604: error occurred at recursive SQL level 1 ORA-01882: timezone region not found | Один из вариантов решения – определите на сервере приложения переменную окружения TZ со значением, известным серверу БД Oracle. Для этого в скрипт запуска Alpha BI добавьте переменную TZ с указанием в ней временной зоны. Обычно скрипт находится по пути **/etc/systemd/system/alphabi.service**.  Добавьте строчку:   * Environment=TZ=Europe/Moscow.   После этого перечитайте скрипт запуска и перезапустите приложение |
| Ошибка при выполнении запроса к OLAP-серверу:  «Не удалось выполнить запрос к OLAP серверу: Подключение не установлено, т.к. конечный компьютер отверг запрос на подключение» | 1. Проверьте, включен ли OLAP-сервер.  2. Проверьте, доступен ли OLAP-сервер с сервера приложения Alpha BI по порту, который использует Tomcat (например, командой curl -v telnet://{MondrianHost}:{MondrianPort}), адреса можно посмотреть в конфигурационном файле Alpha BI alpha.config, параметр XmlaEndpoint.  3. Проверьте правильность указания адресов серверов в конфигурационных файлах – возможно, допущена ошибка при настройке |

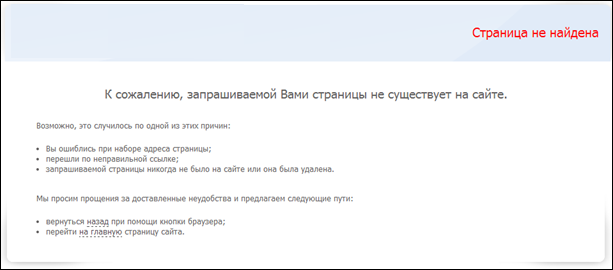


Рисунок  – Сообщение «Страница не найдена»

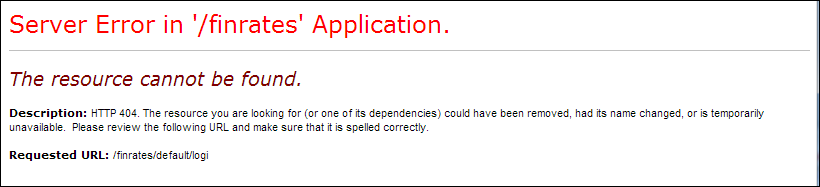


Рисунок  – Ошибка в запрашиваемом URL

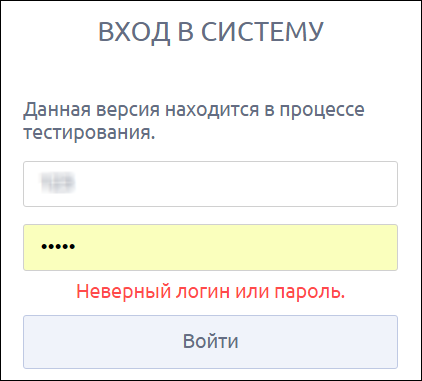


Рисунок  – Сообщение «Ошибка авторизации»

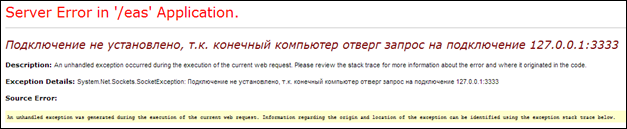


Рисунок  – Сообщение «Остановлена служба БД»

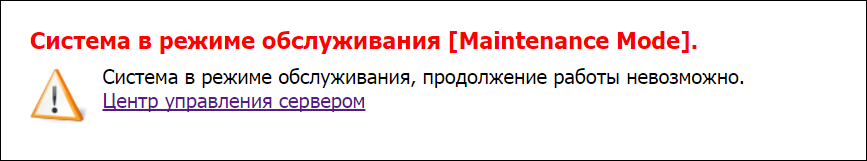


Рисунок  – Сообщение о режиме обслуживания

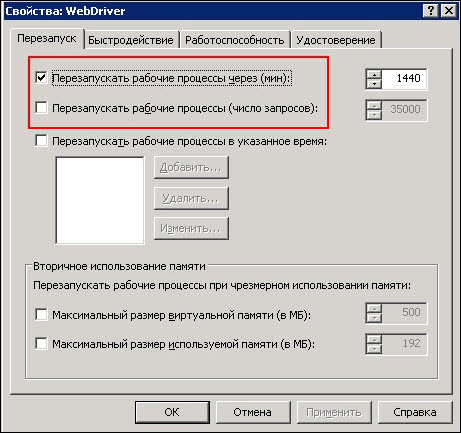


Рисунок  – Настройки IIS. 6 версия

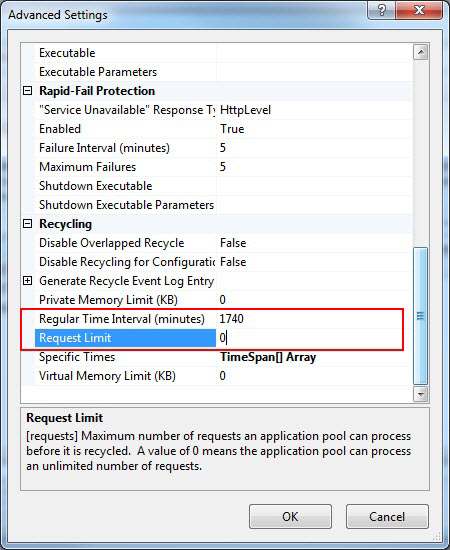


Рисунок  – Настройки IIS. 7 версия

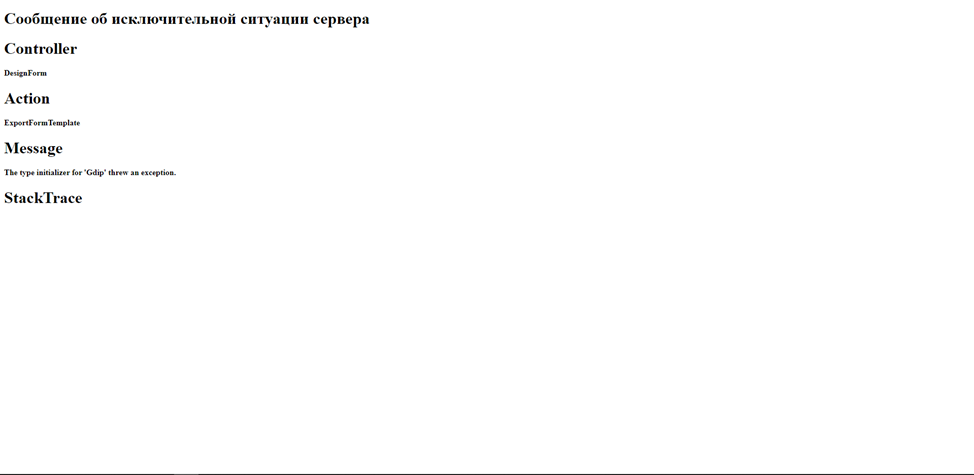


Рисунок  – Сообщение об исключительной ситуации сервера

# Рекомендации по освоению

## Последовательность действий при работе с Системой

Последовательность действий администратора при работе с Системой соответствует следующему алгоритму:

* запустите web-браузер, в случае возникновения ошибки проверьте Интернет-соединение;
* введите в адресной строке web-браузера ссылку для входа в контрольную панель Системы и авторизуйтесь (см. п. 5.1);
* произведите необходимые настройки в контрольной панели;
* просмотрите протокол ошибок;
* выполните выход из контрольной панели;
* войдите в Систему (см. п. 4.6);
* проведите мониторинг сессий пользователей (см. п. 4.8.2.3);
* выполните выход из Системы;
* по истечении установленного в настройках автоматического резервного копирования времени проверьте наличие снятых резервных копий.

Лист регистрации изменений

| Изм. | Версия Системы | Версия документа | Дата внесения изменений | Автор изменений | Краткое описание изменений |
| --- | --- | --- | --- | --- | --- |
| 1 | 2.2 | 01 | 30.04.2015 | Е.В. Фроловичева | Создан документ |
| 2 | 2.5 | 01 | 15.04.2016 | Е.В. Фроловичева | Добавлен пп. «Создание пула приложений», обновлены пп. «Создание узла «alpha», «Установка и настройка Internet Information Service 8.5» |
| 3 | 2.6 | 01 | 09.08.2016 | Е.В. Фроловичева | Добавлен п. «Настройка Системы для аутентификации через Active Directory», обновлен п. «Установка Системы и ее компонентов вручную» |
| 4 | 2.8 | 01 | 25.08.2016 | Я.С. Красько | Добавлен п. «Синхронизация пользователей Системы с Active Directory».  Внесены изменения в п.п. о настройке Системы для авторизации и аутентификации через Active Directory, о нередактируемости учетной записи default, о пояснении причины ошибки обслуживания |
| 5 | 3.0 | 01 | 20.10.2016 | Я.С. Красько | Добавлены п. 4.3, п. 4.8.2.3, п. «Настройка ведения в Системе история хранения метаописания бизнес-объектов». Удален п. «Установка IIS 6» |
| 6 | 3.0 | 01 | 15.12.2016 | Е.В. Фроловичева | Добавлено описание установки Cистемы на Ubuntu и CentOS, установка IIS 8.5 |
| 7 | 3.1 | 01 | 09.01.2017 | В.М. Власова | Форматирование документа. Дополнен п. 5.1.1. Обновлен п. 2.2.2 |
| 8 | 3.2 | 01 | 29.03.2017 | Я. Красько,  Т. Валиуллина | Обновлен п. 2.2.2. Добавлены п. 2.2.3, Организация информационной базы, 3, 4.8.1, приложение «Перечень системных бизнес-объектов Системы», «Настройка отображения полной информации об ошибке в окне детализации», дополнен список аварийных ситуаций, приложение «Тестирование функций Системы» |
| 9 | 4.0 | 01 | 13.04.2018 | Я.С. Красько | Документ скорректирован под версию Alpha 4.0 |
| 10 | 4.1 | 01 | 26.12.2018 | С.Г. Гильманшина | Актуализированы п. «Настройка Apache Tomcat», «Первое разворачивание Системы», «Установка Системы на CentOS 7», 5.1 |
| 11 | 4.2 | 01 | 27.02.2018 | С.Г. Гильманшина | Добавлен п. 3.1.10 |
| 12 | 4.3 | 01 | 19.06.2019 | Н.С. Насритдинова,  С.Р. Гильманшина | Актуализированы п. 2.2.2, 3.1.3, 3.1.6, 3.1.7, «Установка Системы на CentOS 7», 5.1, 5.1.1, 5.1.1.2, 5.1.1.4, 5.1.1.7.  Добавлены п. «Создание дополнительных приложений», 5.1.1.5 |
| 13 | 4.4 | 01 | 27.12.2019 | Гильманшина С.Р. | Изменена версия Системы |
| 14 | 4.4.5 | 01 | 29.06.2020 | Гильманшина С.Р. | Изменена версия Системы |
| 15 | 4.4.8 | 01 | 19.10.2020 | Гильманшина С.Р. | Добавлены п. 5.1.3, 5.1.4 |
| 16 | 4.6 | 01 | 15.12.2020 | Гильманшина С.Р. | Обновлена версия Системы |
| 17 | 4.7.0 | 01 | 14.05.2021 | Жукова А.И. | Актуализированы п. 5.1, 5.1.1, 5.1.2, 5.1.3, 5.1.4, 5.1.5 |
| 18 | 4.8.0 | 01 | 20.10.2021 | Жукова А.И. | Актуализированы рисунки: Рисунок 11, Рисунок 30 |
| 19 | 4.9.0 | 01 | 21.12.2021 | Гильманшина С.Р. | Удалены пункты про бизнес-процессы и упоминание модуля управления бизнес-процессами |
| 20 | 4.10.0 | 01 | 25.01.2022 | Закирова И.И. | Актуализирован п. «Первоначальная установка» |
| 21 | 4.11.0 | 01 | 01.03.2022 | Закирова И.И. | Актуализированы п. 2.2.2, 4.2.6.2.  Актуализированы рисунки: Рисунок 23, Рисунок 28, Рисунок 30.  Добавлены п.: 4.1.2, 4.5.5, 4.2.6.1.3, 4.4 |
| 22 | 4.12.0 | 01 | 30.03.2022 | Закирова И.И. | Обновлена версия Системы |
| 23 | 4.12.1 | 01 | 25.04.2022 | Закирова И.И., Гильманшина С.Р. | Внесены изменения в п. 3.1.5, 4.6, 5.1.1.8 |
| 24 | 4.13.0 | 01 | 26.05.2022 | Закирова И.И. | Актуализирован п. 4.7, 4.8.2.3.  Актуализированы рисунки: Рисунок 11, Рисунок 21, Рисунок 22, Рисунок 27, Рисунок 28, Рисунок 29, Рисунок 30.  Удален п. 4.6 «Настройка Системы для авторизации и аутентификации через Active Directory».  Удален Рисунок «Определение прав доступа пользователю к просмотру информации о сессиях всех пользователей Системы» |
| 25 | 4.14.0 | 01 | 27.06.2022 | Закирова И.И. | Актуализированы рисунки: Рисунок 21, Рисунок 22, Рисунок 27 |
| 26 | 4.15.0 | 01 | 16.12.2022 | Закирова И.И. | Актуализированы п.: 4.1.1, 4.2.6.3.  Актуализированы рисунки: Рисунок 21, Рисунок 22 |
| 27 | 5.0.0 | 01 | 26.12.2022 | Закирова И.И. | Актуализированы п.: 4.4.1, 5.1.4.  Актуализированы рисунки: Рисунок 21, Рисунок 22, Рисунок 41.  Добавлены п.: 4.4.2, 4.4.3 |
| 28 | 5.1.0 | 01 | 01.08.2023 | Жукова А.И. | Добавлен рисунок: Рисунок 36.  Актуализированы п.: 2.2.1, 2.2.2.  Полностью актуализировано описание п. 4.1 – 4.5 |
| 29 | 5.2.0 | 01 | 06.12.2023 | Жукова А.И. | Обновлена версия Системы |
| 30 | 5.3.0 | 01 | 08.12.2023 | Жукова А.И. | Актуализированы п.: 2.2.1, 2.2.2, 4.1, 4.1.1, 4.2.1, 4.2.2, 4.2.2.1, 4.2.3, 4.2.4, 4.2.5, 4.2.6, 4.3, 4.8.1.1.1.  Актуализированы рисунки: Рисунок 17, Рисунок 21, Рисунок 22, Рисунок 27, Рисунок 32, Рисунок 34, Рисунок 40.  Добавлены п.: 4.2.2.2, 4.8.1.2, 4.8.1.3.  Удалены п. «Настройка отображения полной информации об ошибке в окне детализации», «Действия при аварийном отключении электропитания» |
| 31 | 5.3.1 | 01 | 29.02.2024 | Жукова А.И. | Актуализирован п. 2.2.2 |
| 32 | 5.4.0 | 01 | 08.05.2024 | Жукова А.И. | Актуализированы п.: 2.2.1.1, 2.2.2, 4.1, 4.1.1, 4.2.1, 4.2.1.3, 4.2.2.1, 4.2.2.1.2.3, 4.2.2.1.3.2, 4.2.2.2, 4.2.2.2.2.3, 4.2.2.2.3.2, 4.2.3, 4.2.4, 4.2.4.3.1, 4.2.4.3.2, 4.2.4.4, 4.2.5, 4.2.5.2, 4.2.5.3, 4.2.6, 4.2.6.3.3, 4.2.6.3.4, 4.3.1, 4.3.2, 4.3.3 |
| 33 | 5.5.0 | 01 | 14.08.2024 | Жукова А.И. | Актуализированы п.: 2.2.2, 4.1.1, 4.2.1.3, 4.2.2.1.2.3, 4.2.4.3.1, 4.2.5.3, 4.4.1.  Актуализированы рисунки: Рисунок 17, Рисунок 21, Рисунок 28, Рисунок 30 |