

ПЛАТФОРМА UNIDATA 3.5

РУКОВОДСТВО ПО УСТАНОВКЕ

ВЕРСИЯ 1.3 2016 ГОД Информация, содержащаяся в данном документе, может быть изменена без предварительного уведомления.

Никакая часть данного документа не может быть воспроизведена или передана в любой форме и любыми способами в каких-либо целях без письменного разрешения ООО «ЮниДата».

© Unidata, 2015-2016. Все права защищены.

СОДЕРЖАНИЕ

1	Общ	ие положения	6
	1.1	Краткое описание Платформы Unidata	6
	1.1.1	Назначение, функции и возможности	6
	1.1.2	Состав и структура	6
	1.1.3	Требования к программному и аппаратному обеспечению	9
	1.2	Дополнительные сведения	. 10
	1.2.1	Состав дистрибутива и назначение его частей	. 10
	1.2.2	Последовательность установки компонентов	. 11
2	Поді	отовка окружения	. 12
	2.1	Требования к установке окружения	. 12
	2.1.1	Подоготовительные действия	. 12
	2.2	Действия по установке	. 12
	2.2.1	Создание пользователя	. 12
	2.2.2	Установка Tomcat	. 12
	2.2.3	Установка Elasticsearch	. 12
	2.3	Проверка корректности установки	. 13
3	Уста	новка Хранилища данных. Unidata-store	. 14
	3.1	Частные Требования к установке	. 14
	3.1.1	Требования к программному обеспечению сервера	. 14
	3.1.2	Дополнительные требования к окружению	. 14
	3.2	Подготовительные действия	. 14
	3.3	Действия по установке	. 14
	3.3.1	Развертывание БД	. 14
	3.4	Проверка корректности установки	. 15
4	Уста	новка Поискового сервиса. Unidata-search	. 17
	4.1	Частные требования к установке	. 17
		Требования к программному обеспечению сервера	
		Подготовительные действия	
		Конфигурирование Elasticsearch	
		Создание пустого индекса в Elasticsearch	
		Проверка корректности установки	

5	Уста	новка Клиент-серверного приложения	. 19
	5.1	Частные требования к установке	. 19
	5.1.	1 Unidata-backend	. 19
	5.1.	2 Unidata-frontend	. 19
	5.2	Подготовительные действия	. 19
	5.3	Действия по установке	. 19
	5.3.	1 Развертывание Unidata на Tomcat	. 19
	5.3.	2 Базовая настройка Unidata на Tomcat	. 20
	5.3.	3 Создание модели по умолчанию	. 21
	5.4	Проверка корректности установки	. 21
6	Уста	новка Unidata в кластерной конфигурации	. 22
	6.1	Tomcat	. 22
	6.2	Elasticsearch	. 22
7	Уста	новка Внешней очереди сообщений	. 24
	7.1	Частные Требования к установке	. 24
	7.1.	1 Требования к аппаратной части	. 24
	7.1.	2 Требования к программному обеспечению сервера	. 24
	7.2	Действия по установке	. 24
8	Дал	ьнейшая настройка компонентов платформы	. 25
	8.1	Сводная таблица свойств компонентов	. 25
	8.2	Настройка Unidata-store	. 26
	8.3	Настройка Unidata-search	. 26
	8.4	Настройка Unidata-backend	. 26
	8.4.	1 Параметры настройки приложения	. 26
		2 Параметры настройки доступа к БД	
	8.4.	3 Параметры настройки доступа Tomcat	. 28
	8.5	Настройка Unidata-frontend	. 28
	8.5.	1 Параметры настройки приложения	. 28
	8.5.	2 Параметры настройки доступа Tomcat	. 28
	8.6	Параметры настройки ActiveMQ	. 29
	8.7	Настройка консольных утилит для управления платформой	. 29
9	Про	верка корректности установки	. 31
	9.1	Доступ к пользовательскому интерфейсу платформы Unidata	. 31

	9.2	Действия по устранению возможных неисправностей	31
1(0 Peu	шения по отказоустойчивости	32
	10.1	Настройка отказоустойчивого кластера для Unidata-store	32
	10.	1.1 Введение	32
	10.	1.2 Частные требования и Рекомендации	32
	10.2	Подготовка и настройка	34
	10.	2.1 Подготовка и настройка сетевых интерфесов	34
	10.	2.2 Подготовка и настройка операционной системы	35
	10.3	Установка и настройка отказоустойчивого кластера PostgreSQL	35
		3.1 Основная настройка отказоустойчивого кластера PostgreSQL в режиме аси гоковой репликации	•
	10.4	Проверка работоспособности	39
	10.	4.1 Проверка доступности PostgreSQL сервиса	39
	10.	4.2 Восстановление node1	40
	10.	4.3 Создание резервного файла конфигурации	41
	10.5	Замечания и дополнительная литература	42
1:	1 Пер	речень терминов и скоращений	43
П	рилож	кение А (справочное) Подготовка окружения	44
	A.1	Общие положения	44
	A.2	Дополнительные сведения	44
	A.3	Установка OpenJDK	44
	A.4	Установка PostgreSQL	45
	A.5	Установка ActiveMQ	46

1 ОБЩИЕ ПОЛОЖЕНИЯ

Руководство по установке платформы Unidata предназначено для системных администраторов (системных программистов) и содержит сведения, необходимые для установки платформы Unidata. В документе описан порядок действий по установке и настройке компонентов платформы, а также даны рекомендации по устранению возможных неисправностей и подготовке окружения.

1.1 КРАТКОЕ ОПИСАНИЕ ПЛАТФОРМЫ UNIDATA

1.1.1 НАЗНАЧЕНИЕ, ФУНКЦИИ И ВОЗМОЖНОСТИ

Платформа Unidata предназначена для построения систем централизованного управления информацией. К основным функциям, предоставляемым платформой следует отнести:

- **Функциональный блок «Администрирование».** В рамках данного блока платформа предоставляет инструментарий для управления пользователями и ролями.
- **Функциональный блок «Управление данными».** Предоставляет функциональность для:
 - Создания, просмотра и редактирования модели данных. Модель данных определяет сущности НСИ, их структуру и взаимосвязи. Также для каждой сущности задаются правила качества данных, привила поиска дубликатов и консолидации.
 - о Управления источниками данных.
 - о Просмотра библиотеки функций очистки и обогащения данных.
- **Функциональный блок «Обработка данных».** Поддержка функций поиска, создания, просмотра и редактирования записей.
- **Интеграционный блок.** Набор различных АРІ для интеграции с внешними системами. АРІ поддерживают все основные операции с данными.

1.1.2 СОСТАВ И СТРУКТУРА

❖ ЛОГИЧЕСКИЕ КОМПОНЕНТЫ ПЛАТФОРМЫ UNIDATA

Логическая структура платформы Unidata приведена на схеме 1.

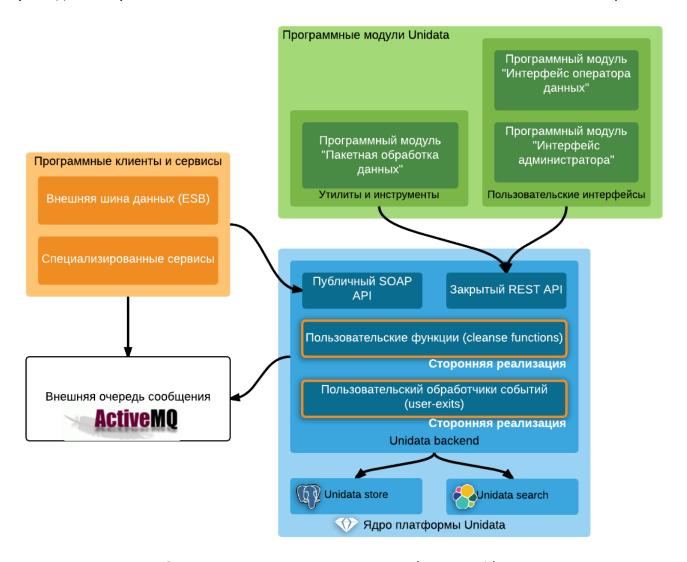


Схема 1 – Логическая структура платформы Unidata

1. Ядро платформы Unidata

- Серверное приложение. Unidata-backend. Серверное Java приложение реализует основную логику платформы, предоставляет API для доступа системам-источникам и системам-получателям к данным. Исполняется под управлением свободного контейнера сервлетов с открытым кодом Tomcat (http://tomcat.apache.org/). Приложение предоставляет механизмы для регистрации и исполнения сторонних Java библиотек реализующих:
 - о пользовательские функции;
 - о пользовательские обработчики событий;
- **Хранилище данных. Unidata-store**. Хранилище данных реализовано поверх реляционной базы данных под управлением свободной СУБД с открытым кодом PostgreSQL (http://www.postgresql.org/);
- Поисковый сервис. Unidata-search. Поисковый индекс для обеспечения функций корпоративного поиска. Разворачивается на базе свободного ПО с открытым кодом Elasticsearch (https://www.elastic.co/).

2. Программные модули

- интерфейсы. Unidata-frontend. Пользовательские Пользовательские интерфейсы "Интерфейс оператора данных" и "Интерфейс администратора" **WEB** приложениями, являются клиентскими которые предоставляет пользователям доступ ко всем основным функциям платформы: администрирование системы, управление данными и обработка данных. Данные приложения взаимодействуют с Unidata-backend посредством закрытого REST АРІ. Приложения исполняются под управлением свободного контейнера сервлетов с открытым кодом Tomcat (http://tomcat.apache.org/);
- Пакетная обработка данных. Набор консольных утилит, предназначенный для пакетной загрузки больших объемов данных. Утилиты написаны на языке программирования Java и взаимодействуют с Unidata-backend посредством закрытого REST API.

3. Механизмы асинхронного взаимодействия

- Внешняя очередь сообщений. Опциональный компонент, необходимый для построения асинхронный интеграции псевдо-реального времени со смежными информационными системами. Конкретное выбранное решение должно поддерживать спецификацию JMS (https://en.wikipedia.org/wiki/Java Message Service). В данное руководство входит описание настройки интеграции со свободным ПО с открытым кодом ActiveMQ (http://activemq.apache.org/).
- 4. Программные клиенты и сервисы. Специализированные программные компоненты, которые создаются в рамках внедрения платформы с учетом функциональной специфики, а также ИТ ландшафта заказчика. Компоненты взаимодействуют с публичным АРІ платформы, а также с очередями сообщений. Данные компоненты не являются частью ядра платформы или ее модулей. Они представлены на диаграмме для формирования целостной картины способов взаимодействия с платформой. Реализация данных компонент обычно возлагается на ИТ отдел заказчика или партнера по внедрению.

❖ ВАРИАНТЫ УСТАНОВКИ КОМПОНЕНТОВ ПЛАТФОРМЫ.

Возможны различные варианты установки компонентов платформы Unidata:

- Все компоненты платформы могут быть установлены на один физический или виртуальный сервер.
- Каждый компонент может быть развернут на отдельном физическом или виртуальном сервере, в зависимости от поставленных целей и требований к производительности и отказоустойчивости.

1.1.3 ТРЕБОВАНИЯ К ПРОГРАММНОМУ И АППАРАТНОМУ ОБЕСПЕЧЕНИЮ

Требования к программному и аппаратному обеспечению платформы Unidata включают в себя общие требования к системному ПО, к программному обеспечению сервера и к сетевому окружению платформы, а также частные требования к аппаратной части и к программному обеспечению серверов для отдельных компонентов платформы. Частные требования перечислены в таблице 1, дополнительные сведения приведены в соответствующих разделах. В случае установки нескольких компонентов платформы на одном физическом или виртуальном сервере частные требования к обеспечению компонентов объединяются.

◆ ОБЩИЕ ТРЕБОВАНИЯ

Общие требования данного раздела относятся к установке любого компонента платформы в равной степени.

• Требования к системному ПО

На сервере должна быть установлена одна из следующих операционных систем:

- Linux CentOS 7;
- Fedora Server 22;
- Ubuntu Server 14;
- Red Hat Enterprise Linux 7.0.

Данное руководство написано для Linux CentOS 7

• Требования к программному обеспечению сервера

На сервере должно быть установлено следующее ПО:

OpenJDK 1.8.

• Требования к настройке операционной системы

Со стороны сервера также должны быть выполнены следующие требования:

- На сервере опционально должна быть разрешена установка пакетов через yum;
- На сервере должны быть открыты ssh и порт 8080 для доступа снаружи.

Известны логины и пароли для пользователя: root.

• Требования к сетевому окружению

В случае разворачивания платформы на нескольких физических или виртуальных серверах, между ними должно быть обеспеченно взаимодействие на скорости 10 Гбайт/с.

❖ МИНИМАЛЬНЫЕ ЧАСТНЫЕ ТРЕБОВАНИЯ

В таблице 1 перечислены минимальные частные требования к аппаратной части и ПО сервера, на котором будет установлен компонент платформы Unidata.

Таблица 1 – Минимальные частные требования к обеспечению сервера

	МИНИМАЛЬНЫЕ ЧАСТНЫЕ ТРЕБОВАНИЯ	
КОМПОНЕНТ	К АППАРАТНОЙ ЧАСТИ	к по
ХРАНИЛИЩЕ ДАННЫХ. UNIDATA-STORE	 2 физических или виртуальных ядра процессора 4 Гбайт оперативной памяти 10 Гбайт свободного места на жестком диске после установки системного ПО 	PostgreSQL 9.4
ПОИСКОВЫЙ СЕРВИС. UNIDATA-SEARCH	 2 физических или виртуальных ядра процессора 8 Гбайт оперативной памяти 10 Гбайт свободного места на жестком диске после установки системного ПО 	Elasticsearch 1.4.4
КЛИЕНТ-СЕРВЕРНОЕ ПРИЛОЖЕНИЕ. UNIDATA-BACKEND И UNIDATA-FRONTEND	 4 физических или виртуальных ядра процессора 8 Гбайт оперативной памяти 8 Гбайт свободного места на жестком диске после установки системного ПО 	Apache Tomcat 7.0.61
ВНЕШНЯЯ ОЧЕРЕДЬ СООБЩЕНИЙ (НЕОБЯЗАТЕЛЬНЫЙ КОМПОНЕНТ)	 2 физических или виртуальных ядра процессора 2 Гбайт оперативной памяти 4 Гбайт свободного места на жестком диске после установки системного ПО 	ActiveMQ 5

1.2 ДОПОЛНИТЕЛЬНЫЕ СВЕДЕНИЯ

1.2.1 СОСТАВ ДИСТРИБУТИВА И НАЗНАЧЕНИЕ ЕГО ЧАСТЕЙ

В состав дистрибутива платформы Unidata 3.5 входит инсталляционный пакет: *Unidata-3.5.03.tar.gz*, который предназначен для установки платформы на предварительно подготовленное окружение (процедура подготовки окружения платформы Unidata описана в приложении A).

Структура инсталляционного пакета:

database - скрипты создания базы

Unidata-schema.sql

Tomcat - конфигурация, библиотеки, war и jar файлы

- bin
- conf
- lib
- shared
- webapps

util – Набор консольных утилит для управления платформой

version - файл с версией

ThirdParty – компоненты необходимые для стороннего ПО, используемого платформой

Elasticsearch

В состав дистрибутива платформы Unidata 3.5 не входят инсталляционные пакеты для установки окружения платформы.

1.2.2 ПОСЛЕДОВАТЕЛЬНОСТЬ УСТАНОВКИ КОМПОНЕНТОВ

Последовательность установки платформы Unidata и ее компонентов следующая:

- 1) установка окружения;
- 2) установка хранилища данных. Unidata-store;
- 3) установка поискового сервиса. Unidata-search;
- 4) установка клиент-серверного приложения. Unidata-frontend и Unidata-backend;
- 5) установка внешней очереди сообщений;
- 6) проверка корректности установки.

2 ПОДГОТОВКА ОКРУЖЕНИЯ

2.1 ТРЕБОВАНИЯ К УСТАНОВКЕ ОКРУЖЕНИЯ

Для своей работы платформа Unidata требует наличие следующего ПО:

- OpenJDK 1.8;
- PostgreSQL 9.4.

Прочие требования к установке окружения перечислены в подразделе 1.1.3.

2.1.1 ПОДОГОТОВИТЕЛЬНЫЕ ДЕЙСТВИЯ

Процедура подготовки окружения платформы Unidata описана в приложении A.

2.2 ДЕЙСТВИЯ ПО УСТАНОВКЕ

2.2.1 СОЗДАНИЕ ПОЛЬЗОВАТЕЛЯ

Чтобы создать пользователя unidata:

1. Подключитесь к серверу, как пользователь root:

```
ssh root@<SERVER NAME OR IP>
```

2. Создайте пользователя unidata:

```
adduser unidata passwd unidata
```

3. Создайте рабочую директорию:

```
mkdir /opt/unidata
chown unidata /opt/unidata
```

4. Переключиться в пользователя *unidata* и перейти в рабочую директорию:

```
ssh unidata@<SERVER_NAME_OR_IP>
su unidata
cd /opt/unidata
```

2.2.2 YCTAHOBKA TOMCAT

Чтобы установить Tomcat:

- 1. Скачайте и скопируйте apache-tomcat-7.0.61.tar.gz в папку /opt/unidata.
- 2. Распакуйте архив:

```
cd /opt/unidata
gunzip apache-tomcat-7.0.61.tar.gz
tar -xvf apache-tomcat-7.0.61.tar
```

2.2.3 YCTAHOBKA ELASTICSEARCH

Чтобы установить Elasticsearch:

- 1. Скачайте и скопируйте elasticsearch-1.4.4.tar.gz в папку /opt/unidata;
- 2. Распакуйте архив:

```
cd /opt/unidata
gunzip elasticsearch-1.4.4.tar.gz
tar -xvf elasticsearch-1.4.4.tar
```

2.3 ПРОВЕРКА КОРРЕКТНОСТИ УСТАНОВКИ

Процедура проверки корректности установки описана в разделе 8.

3 УСТАНОВКА ХРАНИЛИЩА ДАННЫХ. UNIDATA-STORE

3.1 ЧАСТНЫЕ ТРЕБОВАНИЯ К УСТАНОВКЕ

Общие требования к установке приведены в подразделе 1.1.3.

Минимальные частные требования к аппаратной части сервера для установки хранилища данных перечислены в таблице 1.

Частные требования к ПО сервера перечислены ниже.

3.1.1 ТРЕБОВАНИЯ К ПРОГРАММНОМУ ОБЕСПЕЧЕНИЮ СЕРВЕРА

На сервере должно быть установлено следующее ПО:

PostgreSQL 9.4.

3.1.2 ДОПОЛНИТЕЛЬНЫЕ ТРЕБОВАНИЯ К ОКРУЖЕНИЮ

На СУБД PostgreSQL разрешены подключения через JDBC по IP адресу, логину и паролю.

Известны логины и пароли для пользователя: postgres.

3.2 ПОДГОТОВИТЕЛЬНЫЕ ДЕЙСТВИЯ

Процедура подготовки окружения платформы Unidata 3.5 описана в разделе 2.

3.3 ДЕЙСТВИЯ ПО УСТАНОВКЕ

3.3.1 РАЗВЕРТЫВАНИЕ БД

Чтобы развернуть базу данных:

1. Подключитесь к серверу базы данных как пользователь ОС unidata:

ssh unidata@<SERVER NAME OR IP>

2. Переключитесь в ОС пользователя postgres:

su root su postgres

3. Создайте БД unidata:

psql -U postgres
 create database unidata;
\quit

4. Переключитесь в ОС как пользователь *unidata* и перейдите в рабочую директорию:

su unidata
cd /opt/unidata

- 5. Скопируйте папку "database" из дистрибутива в рабочую директорию
- 6. Отредактируйте файл init env.sh из директории database.

Содержимое файла:

```
export JAVA_HOME=/opt/jdk1.8.0_45
export UNISTORE_DB_HOST=localhost
export UNISTORE_DB_PORT=5432
export UNISTORE_DB_NAME=unidata
export UNISTORE_DB_USER_NAME=postgres
export UNISTORE_DB_USER_PASSWORD=postgres
export UNISTORE_DB_USER_PASSWORD=postgres
export UNISTORE_DB_SCHEMA=public
```

7. Запустите скрипт разворачивания базы:

```
./update_database.sh
```

3.4 ПРОВЕРКА КОРРЕКТНОСТИ УСТАНОВКИ

Для проверки корректности установки:

- 1. Подключитесь к серверу базы данных к БД unidata от имени пользователя postgres.
- 2. Выполните запрос:

```
select relname from pg_stat_user_tables where schemaname = 'public' order by relname;
```

Результат должен содержать как минимум следующие имена таблиц:

- "binary data"
- "character data"
- "dq errors"
- "duplicates"
- "etalons"
- "etalons_relations"
- "meta_model"
- "meta storage"
- "meta ui"
- "origins"
- "origins relations"
- "origins relations vistory"
- "origins vistory"
- "s password"
- "s resource"
- "s right"
- "s_right_s_resource"
- "s role"
- "s_token"
- "s user"
- "s user s role"
- "schema version"

- "system_auth_token"
- "user_logins"
- "user_roles"
- "user_token"

4 УСТАНОВКА ПОИСКОВОГО СЕРВИСА. UNIDATA-SEARCH

4.1 ЧАСТНЫЕ ТРЕБОВАНИЯ К УСТАНОВКЕ

Общие требования к установке приведены в подразделе 1.1.3.

Минимальные частные требования к аппаратной части сервера для установки поискового сервера перечислены в таблице 1.

Частные требования к ПО сервера перечислены ниже.

4.1.1 ТРЕБОВАНИЯ К ПРОГРАММНОМУ ОБЕСПЕЧЕНИЮ СЕРВЕРА

На сервере должно быть установлено следующее ПО:

Elasticsearch 1.4.4.

4.2 ПОДГОТОВИТЕЛЬНЫЕ ДЕЙСТВИЯ

Процедура подготовки окружения платформы Unidata 3.5 описана в разделе 2.

4.3 ДЕЙСТВИЯ ПО УСТАНОВКЕ

4.3.1 КОНФИГУРИРОВАНИЕ ELASTICSEARCH

Для конфигурирования Elasticsearch:

1. Подключитесь к серверу для запуска Elasticsearch как пользователь ОС *unidata* и перейдите в директорию Elasticsearch:

```
ssh unidata@<server name or ip>
cd /opt/unidata/elasticsearch-1.4.4
```

2. Измените config/elasticsearch.yml – свойства 'cluster.name: ' и 'node.name: ' должны быть не закомментированы и содержать уникальные значения:

```
cluster.name: <ELASTICSEARCH_CLUSTER_NAME>
node.name: <ELASTICSEARCH_NODE_NAME>
```

3. Добавьте поддержку русской морфологии (плагин) и "head" (плагин).

```
cp -R <UNIDATA PACKAGE DIR>/ThirdParty/Elasticsearch/*
```

Запустите Elasticsearch:

```
nohup ./bin/elasticsearch > elastic.output 2>&1 &
```

4.3.2 СОЗДАНИЕ ПУСТОГО ИНДЕКСА В ELASTICSEARCH

Чтобы создать пустой индекс:

- 1. Подключитесь к серверу с распакованным дистрибутивом платформы Unidata.
- 2. Перейдите в директорию с util:

```
cd <UNIDATA COMMAND LINE TOOL DIR>
```

3. Отредактируйте параметры в файле init.sh и сохраните изменения:

```
JAVA_HOME=<PATH_TO_JAVA_HOME>

SEARCH_INDEX_NAME=unidata
SEARCH_CLUSTER_NAME=<ELASTICSEARCH_CLUSTER_NAME>
SEARCH_NODE_NAME=<ELASTICSEARCH_NODE_NAME>
SEARCH_HOST=<ELASTICSEARCH_HOST_NAME>: 9300

UNISTORE_DB_HOST=<UNISTORE_HOST_NAME>
UNISTORE_DB_PORT=5432
UNISTORE_DB_NAME=unidata
UNISTORE_DB_USER_NAME=postgres
UNISTORE_DB_USER_PASSWORD=postgres
UNISTORE_DB_SCHEMA=public
```

4. Для добавления индекса выполните:

```
./create_index
```

4.4 ПРОВЕРКА КОРРЕКТНОСТИ УСТАНОВКИ

Для проверки корректности установки в браузере откройте: **Ошибка! Недопустимый объект гиперссылки.**

Установка прошла успешно, если на странице отображена информация об индексе Unidata.

5 УСТАНОВКА КЛИЕНТ-СЕРВЕРНОГО ПРИЛОЖЕНИЯ

5.1 ЧАСТНЫЕ ТРЕБОВАНИЯ К УСТАНОВКЕ

Общие требования к установке приведены в подразделе 1.1.3.

Частные требования к установке перечислены ниже.

5.1.1 UNIDATA-BACKEND

❖ МИНИМАЛЬНЫЕ ТРЕБОВАНИЯ К АППАРАТНОЙ ЧАСТИ

Требования к аппаратному обеспечению сервера:

- 2 физических или виртуальных ядра процессора;
- 4 Гбайт оперативной памяти;
- 4 Гбайт свободного места на жестком диске после установки системного ПО.

❖ ТРЕБОВАНИЯ К ПРОГРАММНОМУ ОБЕСПЕЧЕНИЮ СЕРВЕРА

На сервере должно быть установлено следующее ПО:

Apache Tomcat 7.0.61.

5.1.2 UNIDATA-FRONTEND

❖ МИНИМАЛЬНЫЕ ТРЕБОВАНИЯ К АППАРАТНОЙ ЧАСТИ

Требования к аппаратному обеспечению сервера:

- 2 физических или виртуальных ядра процессора;
- 4 Гбайт оперативной памяти;
- 4 Гбайт свободного места на жестком диске после установки системного ПО.

❖ ТРЕБОВАНИЯ К ПРОГРАММНОМУ ОБЕСПЕЧЕНИЮ СЕРВЕРА

На сервере должно быть установлено следующее ПО:

• Apache Tomcat 7.0.61.

5.2 ПОДГОТОВИТЕЛЬНЫЕ ДЕЙСТВИЯ

Процедура подготовки окружения платформы Unidata 3.5 описана в разделе 2.

5.3 ДЕЙСТВИЯ ПО УСТАНОВКЕ

5.3.1 РАЗВЕРТЫВАНИЕ UNIDATA НА ТОМСАТ

Чтобы развернуть Unidata на Tomcat:

1. Подключитесь к серверу, предназначенному для исполнения Tomcat, как пользователь ОС *unidata*:

```
ssh unidata@<SERVER_NAME_OR_IP>
```

2. Перейдите в директорию Tomcat:

cd /opt/unidata/apache-tomcat-7.0.61

- 3. Скопируйте из инсталляционного пакета содержимое папки Tomcat (конфигурационные пакеты, WAR файлы приложения, библиотеки, sh скрипты)
- 4. Чтобы предварительно распаковать дистрибутив платформы Unidata локально на данном сервере, выполните команду:

```
cp -R <UNIDATA_PACKAGE_DIR>/Tomcat/* .
```

5.3.2 БАЗОВАЯ НАСТРОЙКА UNIDATA НА ТОМСАТ

❖ НАСТРОЙКА UNIDATA-BACKEND

Чтобы настроить Unidata-backend:

1. Отредактируйте в файле *<TOMCAT_HOME>/conf/Unidata/backend.properties* параметры и сохраните изменения:

```
unidata.elasticsearch.cluster.name=<ELASTICSEARCH_CLUSTER_NAME>
unidata.elasticsearch.node.name=<ELASTICSEARCH_NODE_NAME>
unidata.elasticsearch.esnodes=<ELASTICSEARCH_HOST_NAME>:9300
```

2. Для настройки параметров доступа к базе данных отредактируйте файл <TOMCAT HOME>/conf/Catalina/localhost/context.xml.default и сохраните изменения:

```
factory="org.apache.tomcat.jdbc.pool.DataSourceFactory"
driverClassName="org.postgresql.Driver"
url="jdbc:postgresql://<UNISTORE_DB_HOST>:5432/unidata"
username="<UNISTORE_DB_USER_NAME>"
password="<UNISTORE_DB_USER_PASSWORD>"
```

- где <UNISTORE_DB_HOST> – имя или адрес хоста с сервером базы данных.

❖ НАСТРОЙКА UNIDATA-FRONTEND

Чтобы настроить Unidata-frontend:

1. Запустите Tomcat:

./bin/catalina start

2. Отредактируйте файл <TOMCAT HOME>/webapps/UniData-frontend/customer.json:

```
"serverUrl" = "http://<BACKEND SERVER NAME OR IP>:8080/unidata-backend/api/",
```

<BACKEND_SERVER_NAME_OR_IP> – имя или IP адрес хоста, на котором исполняется Tomcat с Unidata-backend приложением. Если HTTP порт отличен от 8080, его тоже надо поменять.

5.3.3 СОЗДАНИЕ МОДЕЛИ ПО УМОЛЧАНИЮ

Чтобы создать модель по умолчанию:

- 1. Подключитесь к серверу с распакованным дистрибутивом платформы Unidata.
- 2. Перейдите в директорию с util:

```
cd <UNIDATA COMMAND LINE TOOL DIR>
```

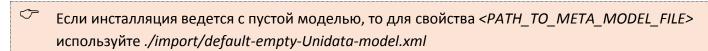
3. Отредактируйте в файле init.sh параметры и сохраните изменения:

```
JAVA_HOME=<PATH_TO_JAVA_HOME>

SEARCH_INDEX_NAME=unidata
SEARCH_CLUSTER_NAME=<ELASTICSEARCH_CLUSTER_NAME>
SEARCH_NODE_NAME=<ELASTICSEARCH_NODE_NAME>
SEARCH_HOST=<ELASTICSEARCH_HOST_NAME>: 9300

UNISTORE_DB_HOST=<UNISTORE_HOST_NAME>
UNISTORE_DB_PORT=5432
UNISTORE_DB_NAME=unidata
UNISTORE_DB_USER_NAME=postgres
UNISTORE_DB_USER_PASSWORD=postgres
UNISTORE_DB_SCHEMA=public
```

4. Отредактируйте свойство <PATH_TO_META_MODEL_FILE> в файле import_meta.sh и сохраните изменения.



5. Выполните *import_meta.sh*:

```
./import meta.sh
```

5.4 ПРОВЕРКА КОРРЕКТНОСТИ УСТАНОВКИ

Для проверки корректности установки после запуска Tomcat перейдите в директорию с *log* файлами командой:

```
cd /opt/unidata/apache-tomcat-7.0.61/logs
```

Установка прошла успешно, если результат содержит следующие строчки:

```
...
INFO: Deployment of web application archive /opt/unidata/apache-tomcat-7.0.61/webapps/unidata-backend.war has finished in 26,421 ms
...
INFO: Deployment of web application archive /opt/unidata/apache-tomcat-7.0.61/webapps/unidata-frontend.war has finished in 605 ms
...
```

а также, java exception stack trace. Единственное допустимое исключение на данный момент:

```
javax.jms.JMSException: Error while attempting to add new Connection to the pool
```

6 УСТАНОВКА UNIDATA В КЛАСТЕРНОЙ КОНФИГУРАЦИИ

Данный способ установки используется для обеспечения отказоустойчивости и масштабируемости системы. Кластеризация требуется только в Elasticsearch. Сервер приложений Tomcat использует балансировку нагрузки без репликации сессий. Перед узлами серверов приложений необходимо настроить и запустить балансировщик. И развернуть поисковый сервер Elasticsearch так, чтобы количество узлов было равным количеству узлов сервера приложений.

6.1 TOMCAT

Ноды рекомендуется разместить на отдельных машинах. В данном случае конфигурация сервера приложений будет такой же, что и для инсталляции из одного узла. Количество тредов и тип коннектора выбираются в зависимости от требований конкретного заказчика. Свойства для соединения приложения с поисковым сервером Elasticsearch, содержащиеся в файле backend.properties, описаны ниже:

• Список всех узлов кластера через запятую, к которым узел может обращаться:

unidata.elasticsearch.esnodes=192.168.10.1:9300,192.168.10.2:9300,192.168.10.3:9300

• Имя кластера, к которому нужно подключиться:

unidata.elasticsearch.cluster.name=Elasticsearch-mmi

• Свойство, используемое в специальных конфигурациях, должно существовать, но быть незаполненным:

unidata.elasticsearch.node.name=

6.2 ELASTICSEARCH

Для машин, на которых инсталлированы узлы поискового сервера, рекомендуется использовать дисковый носитель — SSD. Данные в кластере Elasticsearch хранятся в так называемых кусках (shards). Оптимальное количество кусков зависит от данных, однако, существует общее правило — количество кусков не должно быть меньше количества узлов. Ниже приведен пример настроек для кластера из четырех узлов.



Настройки должны быть прописаны на всех узлах кластера, так как с обычными настройками любой узел кластера может стать мастер-узлом.

Настройки, прописанные в файле elasticsearch.yml, могут быть следующими:

• Количество первичных осколков на один индекс:

index.number of shards: 4

Количество вторичных копий каждого из первичных кусков:

index.number of replicas: 1

• Для данных небольших объемов может быть использована инструкция для хранения всех данных только в памяти:

bootstrap.mlockall: true

• Запрет мультикаста:

discovery.zen.ping.multicast.enabled: false

• Использование юникаста с перечислением всех узлов явно:

discovery.zen.ping.unicast.hosts: ["192.168.10.1", "192.168.10.2", "192.168.10.3"]

7 УСТАНОВКА ВНЕШНЕЙ ОЧЕРЕДИ СООБЩЕНИЙ

7.1 ЧАСТНЫЕ ТРЕБОВАНИЯ К УСТАНОВКЕ

7.1.1 ТРЕБОВАНИЯ К АППАРАТНОЙ ЧАСТИ

Требования к аппаратному обеспечению сервера:

- 2 физических или виртуальных ядра процессора;
- 2 Гбайт оперативной памяти;
- 4 Гбайт свободного места на жестком диске после установки системного ПО.

7.1.2 ТРЕБОВАНИЯ К ПРОГРАММНОМУ ОБЕСПЕЧЕНИЮ СЕРВЕРА

На сервере должно быть установлено следующее ПО:

• ActiveMQ 5.

7.2 ДЕЙСТВИЯ ПО УСТАНОВКЕ

Дополнительные действия по установке не требуются.

8 ДАЛЬНЕЙШАЯ НАСТРОЙКА КОМПОНЕНТОВ ПЛАТФОРМЫ

В данном разделе собраны все основные параметры компонентов платформы, настройка которых может потребоваться в процессе эксплуатации.

8.1 СВОДНАЯ ТАБЛИЦА СВОЙСТВ КОМПОНЕНТОВ

В таблице 2 перечислены свойства, которые используются для настройки взаимодействия компонентов платформы Unidata между собой.

Таблица 2 – Сводная таблица свойств

Свойство	Значение по умолчанию	
Unidata-store		
Хост сервера базы данных	-	
Порт сервера базы данных	5432	
Имя базы данных	unidata	
Имя пользователя БД	postgresql	
Пароль пользователя БД	postgresql	
Схема БД	public	
Unidata-search		
Хост сервера Elasticsearch -		
Порт сервера Elasticsearch 9300		
Логическое имя поискового кластера	unidata_search	
Логическое имя поискового узла	unidata_search_1	
Логическое имя индекса	unidata	
Unidata-backend		
Хост сервера Tomcat	-	
HTTP порт сервера Tomcat	8080	
Unidata-frontend		
Хост сервера Tomcat	-	

Свойство	Значение по умолчанию
HTTP порт сервера Tomcat	8080
Внешняя очередь сообщений	
Хост сервера ActiveMQ	-
Порт сервера ActiveMQ	61616
Логическое имя очереди для публикации уведомлений	UN_NOTIFICATION
Имя пользователя ActiveMQ	admin
Пароль пользователя ActiveMQ	Admin
Максимально разрешенное количество соединений к очереди сообщений	5

8.2 HACTPOЙKA UNIDATA-STORE

Настройка компонента Unidata-store описана в разделе 9.

8.3 НАСТРОЙКА UNIDATA-SEARCH

Все параметры настройки поискового сервиса Elasticsearch расположены в <ELASTIC SEARCH HOME>/config/elasticsearch.yml.

Основные параметры описаны в таблице 3.

Таблица 3 – Параметры настройки поискового сервиса

Параметр	Описание
cluster.name	Логическое имя поискового кластера
node.name	Логическое имя поискового узла
transport.tcp.port	Порт сервера Elasticsearch

После изменения параметров перезапустите сервис.

8.4 HACTPOЙKA UNIDATA-BACKEND

8.4.1 ПАРАМЕТРЫ НАСТРОЙКИ ПРИЛОЖЕНИЯ

Файл: /opt/unidata/apache-tomcat-7.0.61/conf/unidata/backend.properties

Основные параметры настройки серверного приложения перечислены в таблице 4.

Таблица 4 – Параметры настройки серверного приложения

Параметр	Описание
Unidata.security.token.ttl	Время жизни пользовательской сессии в секундах
Unidata.security.token.cleanup	Расписание обновления информации о пользовательских сессиях
Unidata.security.password.expiration	Время жизни пароля пользователей в секундах
unidata.elasticsearch.esnodes	Хост и порт сервера Elasticsearch
unidata.elasticsearch.cluster.name	Логическое имя поискового кластера
unidata.elasticsearch.node.name	Логическое имя поискового узла
unidata.notification.activemq.broker.url	URL доступа к ActiveMQ, параметризируется хостом и портом
unidata.notification.activemq.broker.userName	Имя пользователя ActiveMQ
unidata.notification.activemq.broker.password	Пароль пользователя ActiveMQ
unidata.notification.queue	Логическое имя очереди для публикации уведомлений
unidata.notification.activemq.broker.maxConnections	Максимально разрешенное количество соединений к очереди сообщений

После изменения параметров перезапустите сервис.

8.4.2 ПАРАМЕТРЫ НАСТРОЙКИ ДОСТУПА К БД

Файл: /opt/unidata/apache-tomcat-7.0.61/conf/Catalina/localhost/context.xml.default

Основные параметры настройки доступа к БД перечислены в таблице 5.

Таблица 5 – Параметры настройки доступа к БД

Параметр	Описание
url	URL доступа к БД Unidata-store, параметризируется хостом, портом и именем БД
username	Имя пользователя БД

Параметр	Описание
password	Пароль пользователя БД
initialSize	Начальный размер пула соединений к БД
maxActive	Максимальный размер пула соединений к БД

После изменения параметров перезапустите сервис.

За описанием остальных свойств обратитесь к руководству администратора ПО Tomcat.

8.4.3 ПАРАМЕТРЫ НАСТРОЙКИ ДОСТУПА ТОМСАТ

Файл: /opt/unidata/apache-tomcat-7.0.61/conf/server.xml

Основные параметры настройки доступа Tomcat перечислены в таблице 6.

Таблица 6 – Параметры настройки доступа Tomcat

Параметр	Описание
Connector port="8080" protocol="HTTP/1.1"	HTTP порт доступа к Tomcat

После изменения параметров перезапустите сервис.

За описанием остальных свойств обратитесь к руководству администратора Tomcat.

8.5 HACTPOЙKA UNIDATA-FRONTEND

8.5.1 ПАРАМЕТРЫ НАСТРОЙКИ ПРИЛОЖЕНИЯ

Файл: /opt/unidata/apache-tomcat-7.0.61/webapps/unidata-frontend/index.html

Основные параметры настройки клиентского приложения перечислены в таблице 8.

Таблица 8 – Параметры настройки клиентского приложения

Параметр	Описание
serverUrl	URL доступа к Unidata-backend, параметризируется хостом и HTTP портом

После изменения параметров перезапустите сервис.

8.5.2 ПАРАМЕТРЫ НАСТРОЙКИ ДОСТУПА ТОМСАТ

Файл: /opt/unidata/apache-tomcat-7.0.61/conf/server.xml

Основные параметры настройки доступа Tomcat перечислены в таблице 9.

Таблица 9 – Параметры настройки доступа Tomcat

Параметр	Описание	
Connector port="8080" protocol="HTTP/1.1"	HTTP порт доступа к Tomcat	

После изменения параметров перезапустите сервис.

За описанием остальных свойств обратитесь к руководству администратора Tomcat.

8.6 ПАРАМЕТРЫ НАСТРОЙКИ ACTIVEMQ

Дополнительной настройки не требуется. В случае отсутствия очереди, платформа создаст очередь автоматически с именем указанным в свойстве *Unidata.notification.queue* в конфигурационном файле *<TOMCAT_HOME>/conf/unidata/backend.properties*.

8.7 НАСТРОЙКА КОНСОЛЬНЫХ УТИЛИТ ДЛЯ УПРАВЛЕНИЯ ПЛАТФОРМОЙ

В состав дистрибутива платформы Unidata входит ряд консольных утилит, для правильного функционирования которых требуется произвести настройку файла: <UNIDATA_COMMAND_LINE_TOOL_DIR>/init.sh

Основные параметры настройки консольных утилит перечислены в таблице 10.

Таблица 10 – Параметры настройки консольных утилит

Параметр	Описание	Значение по умолчанию
JAVA_HOME	Путь до расположения Java машины	/usr/lib/jvm/ openjdk-1.8.0/
SEARCH_INDEX_NAME	Логическое имя поискового индекса	unidata
SEARCH_CLUSTER_NAME	Логическое имя поискового кластера	unidata_search
SEARCH_NODE_NAME	Логическое имя поискового узла	unidata_search_1
SEARCH_HOST	Хост и порт сервера Elasticsearch	localhost:9300
UNISTORE_DB_HOST	Хост сервера БД Unidata-store	localhost
UNISTORE_DB_PORT	Порт сервера БД Unidata-store	5432

Параметр	Описание	Значение по умолчанию
UNISTORE_DB_NAME	Имя БД Unidata-store	unidata
UNISTORE_DB_USER_NAME	Имя пользователя БД Unidata-store	postgresql
UNISTORE_DB_USER_PASSWORD	Пароль пользователя БД Unidata-store	postgresql
UNISTORE_DB_SCHEMA	Схема БД Unidata-store	public
STAGING_DB_HOST	Хост сервера БД с исходными localhost данными	
STAGING_DB_PORT	Порт сервера БД с исходными данными	5432
STAGING_DB_NAME	Имя БД с исходными данными	staging
STAGING_DB_USER_NAME	Имя пользователя БД с исходными данными	postgresql
STAGING_DB_USER_PASSWORD	Пароль пользователя БД с исходными данными	postgresql
STAGING_DB_SCHEMA	Схема БД с исходными данными	public

9 ПРОВЕРКА КОРРЕКТНОСТИ УСТАНОВКИ

Чтобы проверить корректность установки выполните вход от имени администратора.

9.1 ДОСТУП К ПОЛЬЗОВАТЕЛЬСКОМУ ИНТЕРФЕЙСУ ПЛАТФОРМЫ UNIDATA

Для доступа к пользовательскому интерфейсу платформы Unidata откройте в браузере: **Ошибка! Недопустимый объект гиперссылки.**

Войдите в систему, используя следующие имя пользователя и пароль:

- Имя пользователя: admin;
- Пароль пользователя: admin.

Установка прошла успешно, если выполняется вход в систему.

9.2 ДЕЙСТВИЯ ПО УСТРАНЕНИЮ ВОЗМОЖНЫХ НЕИСПРАВНОСТЕЙ

Если вход в систему не выполняется, повторите действия подраздела 1.2.2 или обратитесь за помощью в службу технической поддержки.

10 РЕШЕНИЯ ПО ОТКАЗОУСТОЙЧИВОСТИ

Данный раздел описывает порядок действий по установке и конфигурированию Master-Slave отказоустойчивого кластера базы данных PostgreSQL.

10.1 НАСТРОЙКА ОТКАЗОУСТОЙЧИВОГО КЛАСТЕРА ДЛЯ UNIDATA-STORE

10.1.1 ВВЕДЕНИЕ

Список используемых утилит:

PACEMAKER – менеджер ресурсов кластера (Cluster Resource Manager), главная задача которого заключается в достижении максимальной доступности управляемых им ресурсов и их защиты от сбоев как на уровне самих ресурсов, так и на уровне целых узлов кластера.

Архитектура Pacemaker состоит из трех уровней:

КЛАСТЕРОНЕЗАВИСИМЫЙ УРОВЕНЬ— на этом уровне располагаются сами ресурсы и их скрипты, которыми они управляются, и локальный демон, который скрывает от других уровней различия в стандартах, использованных в скриптах;

МЕНЕДЖЕР РЕСУРСОВ (PACEMAKER), реагирует на события, происходящие в кластере: отказ или присоединение узлов, ресурсов, переход узлов в сервисный режим и другие административные действия. Расетакег, исходя из сложившейся ситуации, делает расчет наиболее оптимального состояния кластера и дает команды на выполнение действий для достижения этого состояния (остановка/перенос ресурсов или узлов);

ИНФОРМАЦИОННЫЙ УРОВЕНЬ – на этом уровне осуществляется сетевое взаимодействие узлов. На данном уровне работает Corosync, который отвечает за передачу сервисных команд, обмен информацией о полноте состава кластера.

PCS – набор инструментариев для работы с кластерной технологией (скрипты, утилиты обработчики).

10.1.2 ЧАСТНЫЕ ТРЕБОВАНИЯ И РЕКОМЕНДАЦИИ

В качестве примера, будут описаны действия по установке и настройке двухузлового, Master-Slave отказоустойчивого кластера в режиме асинхронной потоковой репликации. Системные сервисы *SELinux* и *iptables* должны быть выключены на всех элементах кластера. Сетевое оборудование должно быть настроено. Общие требования к ПО перечислены в п. 1.1.3. Частные требования к версиям ПО перечислены ниже:

- Pacemaker 1.0,
- Corosync-2.3,
- pcs-0.9.

❖ РЕКОМЕНДАЦИИ К ТОПОЛОГИИ СЕТИ

Ниже приведены рекомендации по настройке трех сетевых интерфейсов на обоих узлах для корректной работы кластерной технологии:

- 1) для сервиса PostgreSQL,
- 2) для кластерного оркестратора,
- 3) для асинхронной потоковой репликации.

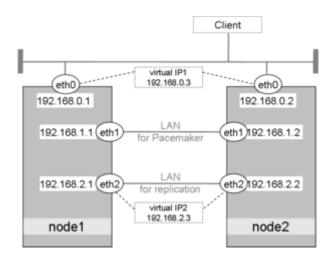
И двух виртуальных ІР адресов:

- 1) для подключения клиента БД к Master-узлу, на котором будет находиться сервис БД PostgreSQL,
- 2) для подключения Slave к асинхронной репликации.

Таблица 11 – Параметры и имена узлов сетевой конфигурации

Имена узлов	Сетевые интерфейсы	IP адреса	Описание
	eth0	192.168.0.1	LAN для сервиса
node1	eth1	192.168.1.1	LAN для Pacemaker
	eth2	192.168.2.1	LAN для репликации
	eth0	192.168.0.2	LAN для сервиса
node2	eth1	192.168.1.2	LAN для Pacemaker
	eth2	192.168.2.2	LAN для репликации
virtual IP	virtual IP1	192.168.0.3	vip для eth0 (клиент БД подключается к данному IP для доступа к PostgreSQL(Master))
(vip)	virtual IP2	192.168.2.3	vip для eth2 (Slave подключается к этому IP для репликации)

Логическая схема топологии сети кластера приведена на рисунке 1.



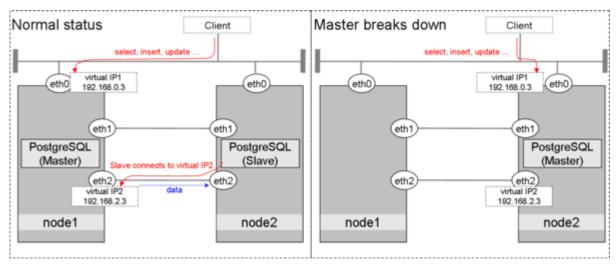


Рисунок 1 – Логическая схема топологии сети кластера

10.2 ПОДГОТОВКА И НАСТРОЙКА

10.2.1 ПОДГОТОВКА И НАСТРОЙКА СЕТЕВЫХ ИНТЕРФЕСОВ

Со стороны активного сетевого оборудования предварительно должна быть выполнена конфигурация. Для настройки сетевых интерфейсов выполните следующие действия:

- 1. Измените имена узлов кластера на "node1" и "node2":
 - о на первом узле:

```
cat node1 > /etc/hostname
hostname node1
```

о на втором узле:

```
cat node2 > /etc/hostname
hostname node2
```

- 2. На обоих узлах остановите и уберите из автозагрузки NetworkManager.
- 3. Сконфигурируйте сетевые интерфейсы как описано ниже:

```
systemctl stop NetworkManager.service
systemctl disable NetworkManager.service
```

- 4. Присвойте сетевым интерфейсам корректные настройки на обоих узлах кластера (см. «Рекомендации к топологии сети» в п. 10.1.2):
 - Пример конфигурации сетевого интерфейса eth0:

```
/etc/sysconfig/network-scripts/ifcfg-eth0
TYPE=Ethernet
BOOTPROTO=static
NAME=enp0s3
UUID=354f4719-fe36-4095-88ea-752b4e26e903
DEVICE=enp0s3
ONBOOT=yes
IPADDR=192.168.0.5
PREFIX=24
```

о Пример конфигурации сетевого интерфейса *eth0:0* с виртуальным IP-адресом:

/etc/sysconfig/network-scripts/ifcfg-eth0:0

```
TYPE=Ethernet
BOOTPROTO=static
NAME=enp0s3
UUID=354f4719-fe36-4095-88ea-752b4e26e903
DEVICE=enp0s3
ONBOOT=yes
IPADDR=192.168.0.3
PREFIX=24
```

о Пример конфигурации сетевого интерфейса eth2:0 с виртуальным IP-адресом:

```
/etc/sysconfig/network-scripts/ifcfg-eth2:0
TYPE=Ethernet
BOOTPROTO=static
NAME=enp0s3
UUID=354f4719-fe36-4095-88ea-752b4e26e903
DEVICE=enp0s3
ONBOOT=yes
IPADDR=192.168.2.3
PREFIX=24
```

- 5. После конфигурации на обоих узлах перезапустите сетевой сервис.
- 6. Проверьте доступность узлов командой ping:

```
systemctl restart network.service
ping 192.168.0.3
ping 192.168.2.3
```

10.2.2 ПОДГОТОВКА И НАСТРОЙКА ОПЕРАЦИОННОЙ СИСТЕМЫ

- 7. Отключите системные сервисы SELinux и iptables:
 - Замените SELINUX=disabled на обоих узлах в /etc/selinux/config:

```
systemctl stop firewalld.service
systemctl disable firewalld.service
```

10.3 УСТАНОВКА И НАСТРОЙКА ОТКАЗОУСТОЙЧИВОГО КЛАСТЕРА POSTGRESQL

Установите пакеты на обоих узлах из стандартного репозитория:

```
yum -y install postgresql-server pacemaker corosync pcs
yum -y install postgresql-server postgresql-devel postgresql-contrib
```

10.3.1 ОСНОВНАЯ НАСТРОЙКА ОТКАЗОУСТОЙЧИВОГО КЛАСТЕРА POSTGRESQL В РЕЖИМЕ АСИНХРОННОЙ ПОТОКОВОЙ РЕПЛИКАЦИИ

Чтобы настроить отказоустойчивый кластер PostgreSQL в режиме асинхронной потоковой репликации:

1. Замените скрипт оркестрации для агентов ресурсов на обоих узлах из поставляемого в комплекте дистрибутива:

```
cp /distr-Unidata/config/heardbeat/pgsql /usr/lib/ocf/resource.d/heartbeat/
chmod 755 /usr/lib/ocf/resource.d/heartbeat/pgsql
```

❖ НА ПЕРВОМ УЗЛЕ КЛАСТЕРА NODE1

```
su - postgres
$ mkdir /var/lib/pgsql/pg_archive
$ cd /var/lib/pgsql/data
```

```
$ initdb
```

2. Добавьте на первом узле кластера node1 в /var/lib/pgsql/data/postgresql.conf следующую конфигурацию:

```
listen addresses = '*'
wal level = hot standby
synchronous commit = on
archive mode = on
archive command = 'cp %p /var/lib/pgsql/pg archive/%f'
max_wal_senders=5
wal_keep_segments = 32
hot_standby = on
restart after crash = off
replication timeout = 5000
wal_receiver_status_interval = 2
max_standby_streaming_delay = -1
max standby archive delay = -1
synchronous commit = on
restart after crash = off
hot standby feedback = on
```

3. Добавить в /var/lib/pgsql/data/pg hba.conf следующую конфигурацию:

host	all	all	127.0.0.1/32	trust	
host	all	all	192.168.0.0/16	trust	
host	replication	all	192.168.0.0/16	trust	

4. Запустите PostgreSQL на node1:

```
pg_ctl -D /var/lib/pgsql/data start
```

❖ НА ВТОРОМ УЗЛЕ КЛАСТЕРА NODE2

5. Скопируйте данные с первого узла:

```
# su - postgres
$ rm -rf /var/lib/pgsql/data/*
$ pg_basebackup -h 192.168.2.1 -U postgres -D /var/lib/pgsql/data -X stream -P
$ mkdir /var/lib/pgsql/pg archive
```

6. Создать файл recovery.conf следующего содержания:

```
standby_mode = 'on'
primary_conninfo = 'host=192.168.2.1 port=5432 user=postgres application_name=node2'
restore_command = 'cp /var/lib/pgsql/pg_archive/%f %p'
recovery_target_timeline = 'latest'
```

7. Запустите PostgreSQL на node2:

```
pg_ctl -D /var/lib/pgsql/data start
```

8. Проверьте статус репликации на *node1*:

```
# su - postgres
$ psql -c "select client_addr,sync_state from pg_stat_replication;"
    client_addr | sync_state
```

➤ Вывод:

9. Остановите сервис на обоих узлах:

```
$ pg_ctl -D /var/lib/pgsql/data stop
$ exit
```

10. Создайте файл конфигурации на обоих узлах сети и запустите Corosync:

```
/etc/corosync/corosync.conf
quorum {
    provider: corosync votequorum
    expected votes: 2
aisexec {
    user: root
    group: root
}
service {
    name: pacemaker
    ver: 0
totem {
    version: 2
    secauth: off
    interface {
        ringnumber: 0
        bindnetaddr: 192.168.1.0
        mcastaddr: 239.255.1.1
logging {
    to_syslog: yes
systemctl start corosync.service
```

Создайте файл конфигурации на обоих узлах сети и запустите Pacemaker:

```
rm -f /var/lib/pacemaker/cib/cib*
systemctl start pacemaker.service
```

12. Для создания файла конфигурации config.pcs для pcmaker выполните команды:

```
pcs cluster cib pgsql_cfg
pcs -f pgsql_cfg property set no-quorum-policy="ignore"
pcs -f pgsql_cfg property set stonith-enabled="false"
pcs -f pgsql cfg resource defaults resource-stickiness="INFINITY"
pcs -f pgsql_cfg resource defaults migration-threshold="1"
pcs -f pgsql cfg resource create vip-master IPaddr2 \
   ip="192.168.0.3" \
   nic="enp0s3" \
   cidr netmask="24" \
   op start timeout="60s" interval="0s" on-fail="restart" \
   op monitor timeout="60s" interval="10s" on-fail="restart" \
              timeout="60s" interval="0s" on-fail="block"
pcs -f pgsql cfg resource create vip-rep IPaddr2 \
   ip="192.168.2.3" \
   nic="enp0s9" \
   cidr_netmask="24" \setminus
   meta migration-threshold="0" \setminus
   op start timeout="60s" interval="0s" on-fail="stop" \
   op monitor timeout="60s" interval="10s" on-fail="restart" \
op stop timeout="60s" interval="0s" on-fail="ignore"
pcs -f pgsql cfg resource create pgsql pgsql \
   pgctl="/usr/bin/pg ctl" \
   psql="/usr/bin/psql" \
   pgdata="/var/lib/pgsql/data/" \
   rep mode="sync" \
   node list="node1 node2" \
   restore_command="cp /var/lib/pgsql/pg_archive/%f %p" \
   primary_conninfo_opt="keepalives_idle=60 keepalives_interval=5 keepalives_count=5" \
```

```
master ip="192.168.2.3" \
   restart_on_promote='true' \
   op monitor timeout="60s" interval="4s" on-fail="restart" \
   op monitor timeout="60s" interval="3s" on-fail="restart" role="Master" \ op promote timeout="60s" interval="0s" on-fail="restart" \
   op demote timeout="60s" interval="0s" on-fail="stop" \
             timeout="60s" interval="0s" on-fail="block" \
   op stop
   op notify timeout="60s" interval="0s"
pcs -f pgsql cfg resource master msPostgresql pgsql \
   master-max=1 master-node-max=1 clone-max=2 clone-node-max=1 notify=true
pcs -f pgsql cfg resource group add master-group vip-master vip-rep
pcs -f pgsql cfg constraint colocation add master-group with Master msPostgresql INFINITY
pcs -f pgsql_cfg constraint order promote msPostgresql then start master-group symmetrical=false
score=INFINITY
pcs -f pgsql cfg constraint order demote msPostgresql then stop master-group symmetrical=false
score=0
pcs cluster cib-push pgsql_cfg
```

о Либо воспользуйтесь скриптом из дистрибутивного комплекта:

/distr-unidata/config/pacemaker /pacemaker-config-pcs.sh

13. Выполните проверку статуса кластера:

```
crm_mon -Afr -1
```

➤ Вывод:

```
Last updated: Thu Oct 22 20:24:21 2015
Last change: Mon Jul 8 10:22:14 2013 via crm attribute on node1
Stack: corosvnc
Current DC: node2 (419473600) - partition with quorum
Version: 1.1.9-3.fc19-781a388
2 Nodes configured, unknown expected votes
4 Resources configured.
Online: [ node1 node2 ]
Full list of resources:
Resource Group: master-group
                                            Started node1
Started node2
    vip-master (ocf::heartbeat:IPaddr2):
     vip-rep (ocf::heartbeat:IPaddr2):
Master/Slave Set: msPostgresql [pgsql]
     Masters: [ node1 ]
     Slaves: [ node2 ]
Node Attributes:
* Node node1:
   + master-pgsql
                                         : 1000
    + pgsql-data-status
                                        : LATEST
                                         : 000000009000080
   + pgsql-master-baseline
                                         : PRI
   + pgsql-status
* Node node2:
    + master-pgsql
                                        : 100
    + pgsql-data-status
                                        : STREAMING | SYNC
    + pgsql-status
                                         : HS:sync
Migration summary:
* Node node1:
* Node node2:
```

10.4 ПРОВЕРКА РАБОТОСПОСОБНОСТИ

Чтобы проверить работоспособность сервиса на node1 выполните команду:

```
killall -9 postgres
```

Затем проверьте статус кластера командой:

```
crm_mon -Afr -1
```

➤ Вывод:

```
Last updated: Thu Oct 22 25:24:21 2015
Last change: Mon Aug 26 19:12:32 2013 via crm attribute on node2
Stack: corosync
Current DC: node2 (2566957248) - partition with quorum
Version: 1.1.9-3.fc19-781a388
2 Nodes configured, unknown expected votes
4 Resources configured.
Online: [ node1 node2 ]
Full list of resources:
 Master/Slave Set: msPostgresql [pgsql]
    Masters: [ node2 ]
    Stopped: [ node1 ]
 Resource Group: master-group
               vip-master (ocf::heartbeat:IPaddr2):
     vip-rep
Node Attributes:
* Node node1:
    + master-pgsql
                                      : -INFINITY
    + pgsql-data-status
                                      : DISCONNECT
    + pgsql-status
                                      : STOP
* Node node2:
    + master-pgsql
                                      : 1000
    + pgsql-data-status
    + pgsql-master-baseline
                                      : 00000000F000000
    + pgsql-status
                                     : PRI
Migration summary:
* Node node1:
   pgsql: migration-threshold=1000000 fail-count=1000000 last-failure='Mon Aug 26 19:12:26 2013'
* Node node2:
Failed actions:
   pgsql_start_0 (node=node1, call=157, rc=1, status=complete, last-rc-change=Mon Aug 26
19:12:27 2015
, queued=402ms, exec=0ms
): unknown error
```

В результате корректного выполнения описанных выше действий Master-узлом должен стать *node2* и переключить на него VIP.

10.4.1 ПРОВЕРКА ДОСТУПНОСТИ POSTGRESQL СЕРВИСА

Для проверки работоспособности сервиса PostgreSQL и самой базы данных со стороны клиента отправьте запрос в БД на виртуальный IP-адрес (VIP). Для примера ниже указаны команды для основных запросов.

Запись

Команда:

```
psql -h 192.168.0.3 -d Unidata -U postgres -c 'INSERT INTO rep_test VALUES ('\''data 9'\'');'
```

Вывод:

```
-----
INSERT 0 1
```

Чтение

Команда:

```
psql -h 192.168.0.3 -d Unidata -U postgres -c "SELECT * FROM rep test;"
```

➤ Вывод:

```
data 22
data 3
data 4
data 5
data 6
data 7
data 8
data 9
data 34
(9 rows)
```

10.4.2 BOCCTAHOBЛЕНИЕ NODE1

Для восстановления node1 выполните команды:

```
# su - postgres
$ rm -rf /var/lib/pgsql/data/
$ pg_basebackup -h 192.168.2.3 -U postgres -D /var/lib/pgsql/data -X stream -P
$ rm /var/lib/pgsql/tmp/PGSQL.lock
$ exit
# pcs resource cleanup msPostgresql
```

Проверьте статус кластера командой:

```
crm_mon -Afr -1
```

➤ Вывод:

```
Last updated: Thu Oct 22 25:24:21 2015
Last change: Mon Aug 26 19:21:17 2013 via crm attribute on node2
Stack: corosync
Current DC: node2 (2566957248) - partition with quorum
Version: 1.1.9-3.fc19-781a388
2 Nodes configured, unknown expected votes
4 Resources configured.
Online: [ node1 node2 ]
Full list of resources:
Master/Slave Set: msPostgresql [pgsql]
    Masters: [ node2 ]
     Slaves: [ node1 ]
 Resource Group: master-group
    vip-master (ocf::heartbeat:IPaddr2):
                                                Started node2
               (ocf::heartbeat:IPaddr2):
                                                Started node2
```

```
Node Attributes:
* Node node1:
    + master-pgsql
                                        : 100
                                        : STREAMING | SYNC
    + pgsql-data-status
                                        : HS:sync
    + pgsql-status
* Node node2:
                                        : 1000
   + master-pgsql
    + pgsql-data-status
                                       : LATEST
                                      : 00000000F000000
    + pgsql-master-baseline
    + pgsql-status
                                        : PRI
Migration summary:
* Node node1:
* Node node2:
```

10.4.3 СОЗДАНИЕ РЕЗЕРВНОГО ФАЙЛА КОНФИГУРАЦИИ

Чтобы создать резервный файл конфигурации и загрузить его в конфигурацию кластера выполните:

```
property \
    no-quorum-policy="ignore" \
     stonith-enabled="false" \
    crmd-transition-delay="0s"
rsc_defaults \
    resource-stickiness="INFINITY" \
    migration-threshold="1"
ms msPostgresql pgsql \
    meta \
        master-max="1" \
        master-node-max="1" \
        clone-max="2" \
        clone-node-max="1" \
         notify="true"
 clone clnPingCheck pingCheck
group master-group \
       vip-master \
       vip-rep
primitive vip-master ocf:heartbeat:IPaddr2 \
    params \
         ip="192.168.0.3" \
         nic="eth0" \
        cidr netmask="24" \
     op start timeout="60s" interval="0s" on-fail="stop" \
     op monitor timeout="60s" interval="10s" on-fail="restart" \
               timeout="60s" interval="0s" on-fail="block"
    op stop
primitive vip-rep ocf:heartbeat:IPaddr2 \
    params \
         ip="192.168.2.3" \
        nic="eth2" \
        cidr_netmask="24" \
    meta \
            migration-threshold="0" \
     op start timeout="60s" interval="0s" on-fail="restart"
     op monitor timeout="60s" interval="10s" on-fail="restart" \
    op stop
               timeout="60s" interval="0s" on-fail="block"
primitive pgsql ocf:heartbeat:pgsql \setminus
         pgctl="/usr/bin/pg ctl" \
         psql="/usr/bin/psql" \
         pgdata="/var/lib/pgsql/data/" \
         start_opt="-p 5432" \
         rep_mode="sync" \
```

```
node list="node1 node2" \
       restore_command="cp /var/lib/pgsql/pg_archive/%f %p" \
       primary conninfo opt="keepalives idle=60 keepalives interval=5 keepalives count=5" \
       master_ip="192.168.2.3" \
       restart_on_promote="true" \
   op start timeout="60s" interval="0s" on-fail="restart"
    op monitor timeout="60s" interval="4s" on-fail="restart" \
   op monitor timeout="60s" interval="3s" on-fail="restart" role="Master" \
   op promote timeout="60s" interval="0s" on-fail="restart" \
   op demote timeout="60s" interval="0s" on-fail="stop" \
              timeout="60s" interval="0s" on-fail="block" \
   op stop
   op notify timeout="60s" interval="0s"
colocation rsc colocation-1 inf: master-group msPostgresql:Master
order rsc order-1 0: msPostgresql:promote master-group:start symmetrical=false
order rsc order-2 0: msPostgresql:demote
                                         master-group:stop symmetrical=false
```

Затем выполните команду:

crm configure load update <FILE NAME>

10.5 ЗАМЕЧАНИЯ И ДОПОЛНИТЕЛЬНАЯ ЛИТЕРАТУРА

❖ ЗАМЕЧАНИЯ

Чтобы выключить оба узла кластера, сначала остановите Slave, затем – Master.

Все скрипты и файлы конфигурации можно найти в дистрибутиве, поставляемом в комплекте:

config.zip

❖ ДОПОЛНИТЕЛЬНАЕ ССЫЛКИ

http://clusterlabs.org/wiki/PgSQL Replicated Cluster https://wiki.postgresql.org/wiki/Streaming Replication

11 ПЕРЕЧЕНЬ ТЕРМИНОВ И СКОРАЩЕНИЙ

ОБОЗНАЧЕНИЕ (СОКРАЩЕНИЕ)	ОПРЕДЕЛЕНИЕ (РАСШИФРОВКА)		
БД	База данных		
ос	Операционная Система		
по	Программное Обеспечение		
СУБД	Система Управления Базами Данных		
API	Application Programming Interface. Программный интерфейс взаимодействия		
IP	Internet Protocol. Маршрутизируемый протокол сетевого уровня стека TCP/IP		
JDBC	Java DataBase Connectivity. Платформенно-независимый промышленный стандарт взаимодействия Java-приложений с различными СУБД		
JMS	Java Message Service. Стандарт рассылки сообщений		
SSH	Secure Shell. Сетевой протокол прикладного уровня, позволяющий производить удаленное управление ОС. Схож по функциональности с протоколами telnet и rlogin, но, в отличие от них, шифрует весь трафик, включая и передаваемые пароли		
БД	База данных		
ос	Операционная Система		
ОРКЕСТРАЦИЯ	Централизованное управление разнесенными компонентами		

ПРИЛОЖЕНИЕ А (СПРАВОЧНОЕ) ПОДГОТОВКА ОКРУЖЕНИЯ

А.1 ОБЩИЕ ПОЛОЖЕНИЯ

Данное приложение к Руководству по установке платформы Unidata содержит описание процедуры подготовки окружения платформы Unidata 3.5 для справки.

Рекомендованной операционной системой является Linux CentOS 7.0.

Для своей работы, продукт Unidata требует наличие следующих компонентов:

- OpenJDK 1.8;
- PostgreSQL 9.4;
- ActiveMQ 5 (опционально).

А.2 ДОПОЛНИТЕЛЬНЫЕ СВЕДЕНИЯ

ПОСЛЕДОВАТЕЛЬНОСТЬ УСТАНОВКИ КОМПОНЕНТОВ ОКРУЖЕНИЯ

- 1) Установка ОрепJDК
- 2) Установка PostgreSQL
- 3) Установка ActiveMQ 5 (опционально)
- 4) Настройка возможности удаленного подключения

❖ ВОЗМОЖНЫЙ СОСТАВ ДИСТРИБУТИВОВ

При отсутствии возможности подключения к сети Интернет, необходимо иметь дистрибутивы устанавливаемого ПО.

Дистрибутивы устанавливаемого ПО могут называться так:

OpenJDK:

Java-1.8.0-openjdk.rpm

PostgreSQL:

```
postgresq194-server-9.4.4-1PGDG.rhel7.x86_64.rpm
postgresq194-9.4.4-1PGDG.rhel7.x86_64.rpm
postgresq194-contrib-9.4.4-1PGDG.rhel7.x86_64.rpm
postgresq194-libs-9.4.4-1PGDG.rhel7.x86_64.rpm
```

ActiveMQ:

apache-activemq-5.12.0-bin.tar.gz

❖ ОБЩИЕ ТРЕБОВАНИЯ К УСТАНОВКЕ

На сервере должна быть установлена ОС CentOS 7.0.

Должен быть обеспечен доступ к сети Интернет или доступ к дистрибутивам, описанным выше.

A.3 YCTAHOBKA OPENJDK

❖ ПОДГОТОВИТЕЛЬНЫЕ ДЕЙСТВИЯ

Дистрибутив OpenJDK скачать можно здесь:

ftp://fr2.rpmfind.net/linux/fedora/linux/updates/testing/23/x86 64/j/java-1.8.0-openjdk-devel1.8.0.60-14.b27.fc23.x86 64.rpm



При отсутствии возможности подключения к сети Интернет, необходимо иметь дистрибутив OpenJDK, описанный в разделе A.2.

Скопируйте дистрибутив OpenJDK на сервер. Перед выполнением установки убедитесь, что других окружений JDK в системе не установлено:

rpm -qa

◆ ДЕЙСТВИЯ ПО УСТАНОВКЕ

Чтобы установить OpenJDk 1.8 на CentOS, выполните команду:

yum install java-1.8.0-openjdk

ПРОВЕРКА КОРРЕКТНОСТИ УСТАНОВКИ

Проверьте корректность установки OpenJDk:

yum info java-1.8.0-openjdk

❖ ДЕЙСТВИЯ ПО УСТРАНЕНИЮ ВОЗМОЖНЫХ НЕИСПРАВНОСТЕЙ

В случае обнаружения неисправностей повторите действия данного раздела сначала или обратитесь в службу технической поддержки Unidata.

A.4 YCTAHOBKA POSTGRESQL

❖ ПОДГОТОВИТЕЛЬНЫЕ ДЕЙСТВИЯ

Для PostgreSQL дистрибутив скачать можно здесь:

ftp://ftp.postgresql.org/pub/repos/yum/9.4/redhat/rhel-7-x86 64/



При отсутствии возможности подключения к сети Интернет, необходимо иметь дистрибутив PostgreSQL, описанный в разделе A.2.

1. Скопируйте дистрибутив PostgreSQL на сервер и выполните команду установки:

#yum install anypacket.rpm

2. Установите Yum репозиторий:

yum install http://yum.postgresql.org/9.4/redhat/rhel-6-x86_64/pgdg-redhat94-9.4-1.noarch.rpm

❖ ДЕЙСТВИЯ ПО УСТАНОВКЕ POSTGRESQL

3. Установите PostgreSQL командой:

yum -y update && yum -y install postgresq194 postgresq194-server postgresq194-contrib postgresq194-libs

СУБД PostgreSQL в настоящий момент должна быть установлена.

Добавьте ссылку в автозагрузку:

systemctl enable postgresql-9.4



По умолчанию PostgreSQL принимает подключения только с локального компьютера.

- 1. Чтобы иметь возможность удаленного подключения, отредактируйте переменную "listen_addresses" в файле postgresql.conf (адрес по умолчанию: /var/lib/pgsql/9.x/data).
- 2. Присвойте значение "*", вместо "localhost", для возможности подключения с любого компьютера в сети.
- 3. Убедитесь в корректности данных в файле $pg_hba.conf$ (адрес по умолчанию /var/lib/pgsql/9.x/data).
- 4. Чтобы разрешить подключения из локальной сети, строка конфигурации в этом файле, с заголовком "*IPv4 local connections*" будет выглядеть так:

host all all all trust



Для удаленного подключения к серверу, пользователь должен иметь пароль.

НАСТРОЙКА ВОЗМОЖНОСТИ УДАЛЕННОГО ПОДКЛЮЧЕНИЯ

Чтобы настроить фаервол на сервере, откройте порты следующей командой:

firewall-cmd --permanent --add-port=8080/tcp
firewall-cmd --permanent --add-port=8443/tcp

ПРОВЕРКА КОРРЕКТНОСТИ УСТАНОВКИ

Перезагрузите фаервол командой:

firewall-cmd -reload

❖ ДЕЙСТВИЯ ПО УСТРАНЕНИЮ ВОЗМОЖНЫХ НЕИСПРАВНОСТЕЙ

В случае обнаружения неисправностей повторите действия данного раздела сначала или обратитесь в службу технической поддержки Unidata.

A.5 YCTAHOBKA ACTIVEMQ

❖ ПОДГОТОВИТЕЛЬНЫЕ ДЕЙСТВИЯ

Для OpenJDK дистрибутив скачать можно здесь:

#wget http://activemq.apache.org/path/tofile/apache-activemq-5.12.0-bin.tar.gz



При отсутствии возможности подключения к сети Интернет, необходимо иметь дистрибутив ActiveMQ, описанный в разделе A.2.

Дистрибутив ActiveMQ необходимо скопировать на сервер.

◆ ДЕЙСТВИЯ ПО УСТАНОВКЕ

- 1. Для установки скачайте и скопируйте ActiveMQ в папку /opt/unidata.
- 2. Распакуйте архив:

```
cd /opt/unidata
gunzip apache-activemq-5.12.0-bin.tar.gz
tar -xvf apache-activemq-5.12.0-bin.tar
```

❖ СОЗДАНИЕ ПОЛЬЗОВАТЕЛЯ UNIDATA

3. Отредактируйте в файле *apache-activemq-5.12.0/conf/user.properties* параметры и сохраните изменения:

unidata=unidata

❖ ЗАПУСК АСТІVЕМО

Для запуска ActiveMQ выполните команду:

./bin/activemq start

❖ ПРОВЕРКА КОРРЕКТНОСТИ УСТАНОВКИ

Чтобы проверить корректность установки, откройте в браузере: **Ошибка! Недопустимый объект гиперссылки.**

Войдите в систему, используя следующие имя пользователя и пароль:

- Имя пользователя: admin;
- Пароль пользователя: admin.

Установка прошла успешно, если выполняется вход в систему.

❖ ДЕЙСТВИЯ ПО УСТРАНЕНИЮ ВОЗМОЖНЫХ НЕИСПРАВНОСТЕЙ

В случае обнаружения неисправностей повторите действия данного раздела сначала или обратитесь в службу технической поддержки Unidata.