

ХРАНИЛИЩЕ ДАННЫХ О ПОТРЕБЛЕНИИ КЛИЕНТА

Руководство по установке

Версия 4.3.0

nexign

Настоящая документация может быть использована только для поддержки работоспособности продуктов, установленных на основании договора с АО «Нэксайн». Документация может быть передана на основании договора, по которому производится (производилась или будет производиться) установка продуктов, или явно выраженного согласия АО «Нэксайн» на использование данной документации. Если данный экземпляр документации попал к вам каким-либо иным образом, пожалуйста, сообщите об этом в АО «Нэксайн» по адресу, приведенному ниже.

Все примеры, приведенные в документации (в том числе примеры отчетов и экранных форм), составлены на основании тестовой базы АО «Нэксайн». Любое совпадение имен, фамилий, названий компаний, банковских реквизитов и другой информации с реальными данными является случайным.

Все встречающиеся в тексте торговые знаки и зарегистрированные торговые знаки являются собственностью их владельцев и использованы исключительно для идентификации программного обеспечения или компаний.

Данная документация может не отражать некоторых модификаций программного обеспечения. Если вы заметили в документации ошибки или опечатки или предполагаете их наличие, пожалуйста, сообщите об этом в АО «Нэксайн».

Все имущественные авторские права сохраняются за АО «Нэксайн» в соответствии с действующим законодательством.

© АО «Нэксайн», 1992–2024

АО «Нэксайн»

Россия, 199155, Санкт-Петербург, ул. Уральская, д.4 лит.Б, помещение 22Н

Тел.: + 7 (812) 326-12-99; факс: + 7 (812) 326-12-98.

office@nexign.com; www.nexign.com

Содержание

1. Общие сведения	4
2. Предварительные требования	5
2.1. Предварительные настройки	5
2.2. Требования к внешнему окружению	5
3. Установка	6
3.1. Установка gus_admin_tools	6
3.2. Установка компонентов	6
4. Запуск и остановка продукта	7
4.1. Запуск gus_admin_tools	7
4.2. Запуск компонентов продукта	7
5. Проверка работоспособности	8
5.1. Проверка работоспособности сервисов	8
5.2. Статус работы сервисов	8
Приложение А: Конфигурация продукта	11

1. Общие сведения

Данное руководство представляет собой описание комплекса мероприятий, выполняемых для установки продукта «Хранилище данных о потреблении клиента» (GUS).

Дистрибутив продукта может быть размещён в Artifactory и/или на сервере FTP.

2. Предварительные требования

Для установки продукта «Хранилище данных о потреблении клиента» (GUS) выполните [предварительные настройки](#).

Для эксплуатации продукта выполните требования к [внешнему окружению](#).

2.1. Предварительные настройки

Перед установкой продукта выполните настройки, указанные в данном разделе.

Настройка Tomcat

Укажите путь к Tomcat в настройках компонента `oapi_chargesdb_api_backend` в файле `inventory/<схема_развертывания>/group_vars/oapi_chargesdb_backend.yml`.

Настройка ZooKeeper

Перед установкой продукта создайте в ZooKeeper параметры:

- для работы компонентов продукта;
- подключения к базе данных;
- подключения к RabbitMQ;
- включения функциональности.

Настройка конфигурации MBUS (RabbitMQ)

Взаимодействие с внешними системами осуществляется через шину данных MBUS. Для этого предварительно настройте точки обмена, очереди и правила маршрутизации сообщений.

Настройка Kafka

Создайте топики в Kafka для интеграций с внешними системами.

2.2. Требования к внешнему окружению

- РЕД ОС версии 7.3.1 или более новой.
- «Сервер сообщений RabbitMQ» (`RABBITMQ`) версии 3.6 или более новой.
- Apache Cassandra версии 3.11 или более новой, или ScyllaDB версии 1.1.2 или более новой.
- Брокер сообщений Kafka версии 2.13-3.1.0 или более новой.
- Java 8 или более новой.
- Apache ZooKeeper версии 3.5 или более новой.

При необходимости дополнительных интеграций: Apache Spark версии 2.2.0 или более новой.

3. Установка

В главе описывается последовательность действий, выполняемых при установке компонентов продукта.

Порядок установки GUS:

1. Установите компонент `gus_admin_tools`.
2. Установите остальные компоненты.

3.1. Установка `gus_admin_tools`

Для установки `gus_admin_tools` используйте команды консоли, передав аргументы скрипту `gus_console.sh`, например: `gus_console.sh "<command>"`.

Для установки компонента:

1. Создайте пространство ключей в базе данных с помощью команды:

```
create keyspace --name {{cassandra.keyspace}} --durable_writes true
--replication \"{{cassandra.replication}}\" --databases {{
ps.gus.gus_admin_tools.database_to_deploy }}
```

2. Создайте очереди в RabbitMQ:

```
rabbitmq init --host localhost --port 5672 --user admin --password
admin --vhost / --path <path> --scripts .*
```

3. Создайте таблицы и обновите схему до текущей версии:

```
up schema --from <from> --to <to>
```

3.2. Установка компонентов

Порядок установки остальных компонентов GUS:

1. Скачайте из Artifactory архивы:
 - `GUS_4/<version>/<component_name>/app/gus-<version>-<component_name>-<version>-app.zip` – содержит исполняемые файлы компонента;
 - `GUS_4/<version>/<component_name>/app/gus-<version>-<component_name>-<version>-conf.zip` – содержит файлы конфигурации компонента.
2. Распакуйте в каталог, где будет устанавливаться компонент.
3. Заполните файлы конфигурации.
4. Запустите компонент, выполнив команду `cd bin/; ./gusctl.sh start`.

4. Запуск и остановка продукта

В каталогах с дистрибутивами компонентов продукта находится папка `bin`, содержащая командные файлы для запуска, остановки и перезапуска компонента.

4.1. Запуск `gus_admin_tools`

Для запуска компонента `gus_admin_tools`:

- перейдите в подкаталог `bin` компонента `gus_admin_tools`;
- в командной строке запустите скрипт `gus_console.bat` для Windows или `gus_console.sh` для Linux.

4.2. Запуск компонентов продукта

Для запуска остальных компонентов продукта GUS:

- перейдите в подкаталог `bin` компонента;
- в командной строке запустите скрипт `gus_start.bat` для Windows или `gusctl.sh` для Linux.

Используйте следующий формат строки:

```
gusctl.sh <command>
```

где:

- `<command>` – команда для управления может принимать следующие значения:
 - `start` – запуск;
 - `stop` – остановка;
 - `restart` – перезагрузка;
 - `status` – получение информации о статусе работы GUS (запущен или остановлен).

5. Проверка работоспособности

Для определения работоспособности сервисов продукта GUS используется механизм Health Check.

5.1. Проверка работоспособности сервисов

Для проверки работоспособности сервиса используется URI /health или /<app_name>/health с типом GET.

Например, если сервис расположен по адресу http://host_name:8080, то допустимо для проверки использовать следующие варианты URI:

```
http://host_name:8080/health
http://host_name:8080/some-service/health
```

Если сервис доступен, то код возврата – 200.

В случае проблем в работе сервиса код возврата – 503.

5.2. Статус работы сервисов

Для проверки статуса работы сервиса используется URI /status или /<app_name>/status с типом GET.

Например, сервис, расположенный по адресу http://host_name:8080, должен в качестве статуса использовать следующие варианты URI:

```
http://host_name:8080/status
http://host_name:8080/some-service/status
```

Код возврата – 200.

Тело ответа содержит информацию о статусе работы сервиса и имеет следующую структуру:


```
{
  "startUpTime" : "${start.date}",
  "state" : "STARTING|UP|STOPPING|DOWN",
  "status" : "OK|FAIL",
  "product" : "${product.name}",
  "service" : "${service.name}",
  "serviceVersion" : "${service.version}",
  "serviceBuild" : "${service.build}",
  "resources" :
  [
    { "name": "resource-name-1", "connections": ["URI", "host1:port1", ...
    , "hostN:portN"], "status" : "OK|FAIL", "type" : "CONFIG|CACHE|DB|MBUS",
    "description" : "..."},
    ...
  ]
}
```

где:

- `startUpTime` – дата и время старта сервиса в формате `YYYY-MM-DD HH24:MI:SS [+hh[:mm]]`; временная зона указывается обязательно;
- `state` – состояние сервиса:
 - `STARTING` – начинает работать (запускается);
 - `UP` – работает;
 - `STOPPING` – останавливается;
 - `DOWN` – остановлен;
- `status` – статус сервиса:
 - `OK` – работает;
 - `FAIL` – не работает;



Примечание.

Статус `OK` может быть только одновременно с состоянием в статусе `UP`. Если сервис запущен (`UP`), но при этом имеет проблемы в работе (например, недоступны какие-то ресурсы), то статус должен быть `FAIL`. Если сервис стартует (`STARTING`), останавливается (`STOPPING`) или уже остановлен (`DOWN`), то статус должен быть `FAIL`.

- `product` – имя продукта в нижнем регистре;
- `service` – имя сервиса в нижнем регистре; в имени сервиса допускаются символы, цифры, знак "-" (не допускается использовать символ нижнего подчеркивания, так как сервис в будущем может использоваться в Service Discovery через DNS);
- `serviceVersion` – версия компонента, в который входит данный сервис (в один компонент может входить несколько сервисов);
- `serviceBuild` – уникальный идентификатор сборки; может включать в себя строковые символы и цифры;

- `resources` – подробное описание работоспособности ресурсов, от которых зависит данный сервис.

Атрибуты ресурсов, от которых зависит текущий сервис:

- `name` – строковое имя сервиса, от которого зависит вызываемый сервис;

Составное имя сервиса необходимо разделять дефисом (знак минус).

- `connections` – список соединений с сервисами в виде списка `host:port`;

Наличие порта в данном случае является необязательным, также допускается указывать вместо `host URI`, например, подключение к базе данных.

- `status` – статус;

Допустимые значения: `OK` – сервис работает, `FAIL` – сервис недоступен.

- `type` – тип сервиса;

Допускаются следующие варианты для зависимых ресурсов:

- `CONFIG` – сервис, являющийся конфигурацией (например, это может быть ZooKeeper или выделенный Config Server);
- `CACHE` – сервис, являющийся кешем (например, это может быть Couchbase);
- `DB` – сервис, являющийся базой данных (например, это может быть Scylla);
- `MBUS` – сервис, являющийся шиной данных MBUS (например, MBUS может использоваться для RabbitMQ или Kafka, если появится в стеке);
- `BACKEND` – сервис, являющийся backend'ом для текущего сервиса;
- `description` – текстовое поле, которое можно заполнить любой дополнительной информацией.

Приложение А: Конфигурация продукта

Конфигурация продукта задается в каталогах:

- gus-install/group_vars – настройки, которые не изменяются от стенда к стенду;
- gus-install/inventory/default/group_vars/gus – параметры установки продукта по умолчанию;
- install/inventory/<схема_развертывания>/group_vars – настройки, которые могут изменяться от окружения, и настройки, которые переопределяют параметры по умолчанию, указанные в файлах каталога gus-install/inventory/default/group_vars/gus.



Внимание!

В файлах gus-install/inventory/default/group_vars/gus запрещено изменение значений параметров.

При необходимости изменить параметр скопируйте его в соответствующий файл каталога gus-install/install/inventory/<схема_развертывания>/group_vars и задайте ему новое значение.