GS LABS
GS GROUP

Публичный - L0

Система генерации ключей Keys Generation System (KGS)

Руководство по установке

Индекс	KGS-IG
Конфиденциальность	Публичный - L0
Ревизия	1.0
Статус	Согласован

Содержание

1. Аннотация

Данный документ является руководством по установке и первоначальной настройке "Системы генерации ключей Keys Generation System (KGS)" (далее по тексту - KGS или Система). Руководство содержит общие сведения о программе, основные группы задач, решаемых системой, требования к аппаратному и программному обеспечению, процедуры установки, настройки и удаления программы, обязанности и задачи администратора, процедуры настройки программы, управления учетными записями, загрузкой и выгрузкой данных, а также описание основных проблем и способов их устранения.

Документ предназначен для пользователей, осуществляющих обслуживание KGS. Руководство ориентировано на администраторов, имеющих навыки практической работы с СУБД PostgreSQL и ОС семейства Linux (в первую очередь, ОС Debian 11.4), обладающих базовыми знаниями по структуре БД KGS.

⚠ **Данный документ опубликован исключительно с целью изучения системных требований для установки продукта, а также ознакомления с последовательностью и деталями процесса установки. Реальная установка продукта производится с использованием внутренних репозиториев ООО "ПЦТ", доступ к которым предоставляется заказчику по запросу.**

2. Термины и сокращения

Термин	Определение
API (Application Programming Interface, Интерфейс программирования приложений)	Набор готовых классов, процедур, функций, структур и констант, предоставляемых приложением (библиотекой, сервисом) для использования во внешних программных продуктах. Используется программистами для написания всевозможных приложений.
DAL (Database Abstraction Layer)	Один из компонентов общей инфраструктуры, обеспечивающий интерфейс для доступа к единой базе данных всех прочих компонентов и приложений.
DMZ (Demilitarized Zone, демилитаризованная зона)	Сегмент сети, содержащий общедоступные сервисы и отделяющий их от частных. В качестве общедоступного сервиса может выступать, например, веб-сервис: обеспечивающий его сервер, который физически размещён в локальной сети (Интранет), должен отвечать на любые запросы из внешней сети (Интернет), при этом другие локальные ресурсы (например, файловые серверы, рабочие станции) необходимо изолировать от внешнего доступа. Цель DMZ – добавить дополнительный уровень безопасности в локальной сети, позволяющий минимизировать ущерб в случае атаки на один из общедоступных сервисов: внешний злоумышленник имеет прямой доступ только к оборудованию в DMZ.
Framework	Структура программной системы; программное обеспечение, облегчающее разработку и объединение разных компонентов большого программного проекта. Framework может включать вспомогательные программы, библиотеки кода, язык сценариев и другое ПО, облегчающее разработку и объединение разных компонентов большого программного проекта. Обычно объединение происходит за счёт использования единого API.
FTP (File Transfer Protocol)	Протокол передачи файлов – стандартный протокол, предназначенный для передачи файлов по TCP-сетям (например, Интернет). Протокол построен на архитектуре "клиент-сервер" и использует разные сетевые соединения для передачи команд и данных между клиентом и сервером.
NFS (Network File System)	Протокол сетевого доступа к файловым системам. Позволяет подключать (монтировать) удалённые файловые системы через сеть.
Репозиторий, хранилище	Место, где хранятся и поддерживаются какие-либо данные. Чаще всего данные в репозитории хранятся в виде файлов, доступных для дальнейшего распространения по сети.

Сокращение	Расшифровка
API	Application Programming Interface
DAL	Database Abstraction Layer

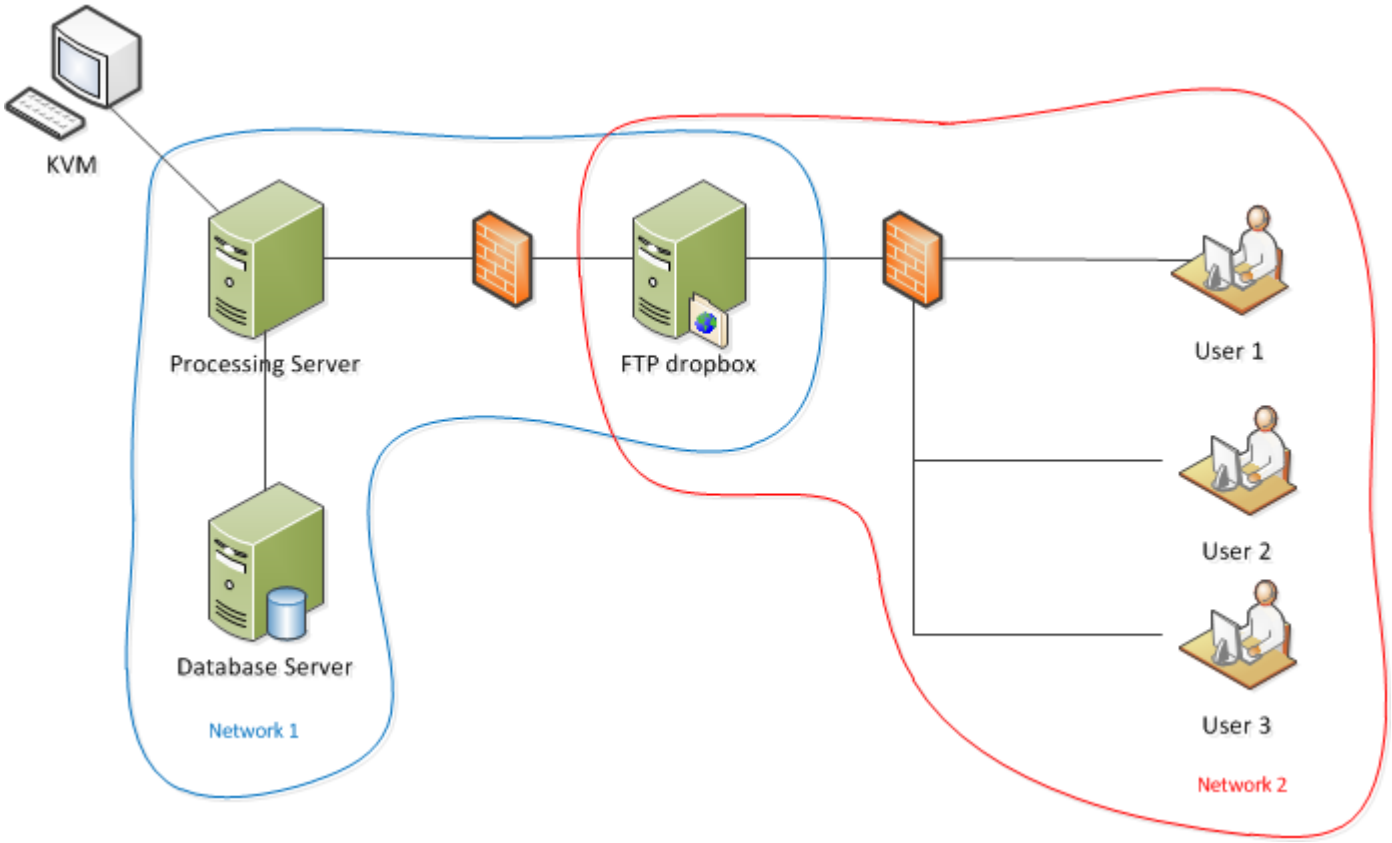
DB	Database
DBMK	Database Master Key
FW	Framework
HWRK	Hardware Root Key
БД	База Данных
СУБД	Система Управления Базами Данных

3. Общие сведения

3.1. Назначение

Система предназначена для работы с ключами, прошиваемыми в однократно программируемую область чипа в процессе его персонализации. Программа предоставляет инфраструктуру, необходимую разработчикам систем, использующих персонализированные ключи. Программа реализует механизмы генерации, безопасного хранения и экспорта ключей для возможности дальнейшего их использования в процессе персонализации чипов на производственной линии.

3.2. Схема развертывания компонентов



На схеме используются следующие обозначения:

- **Database Server** – физический сервер (ОС Debian, x64), на котором развернута база данных (PostgreSQL 13.7 (последней версии))) и модуль Backup. Входит в выделенную физическую сеть “Network 1”.
- **Processing Server** – физический сервер (ОС Debian, x64), на котором развернуты все компоненты Framework (за исключением Backup) и Workflows. Сервер входит в выделенную физическую сеть “Network 1”.
  Оба сервера (Database Server и Processing Server) расположены в отдельном защищенном помещении, доступ в которое ограничен.

> ⚠ На всех серверах используется ОС Debian или аналоги.

- **KVM** (keyboard, video, mouse) - **Терминал** – физическое оборудование, подсоединенное к Processing Server, используемое пользователем для выполнения определенных работ (workflows) в KGS.

> ⓘ Фактически это ЭВМ/терминал, с которой(которого) пользователь управляет KGS (работает в KMI_CONSOLE).
>
> Предполагается, что пользователь имеет доступ к KGS только посредством KVM, который подключен к серверу в защищенном помещении.

> ⚠ **Обратите внимание!** Здесь и далее используются внутренние системные обозначения компонентов (например, KMI_CONSOLE). Соответствие между названиями компонентов (подсистем KGS) и внутренними обозначениями приведено в документе "Система генерации ключей Keys Generation System (KGS). Общее описание", в разделе "Архитектура".

- **FTP dropbox** – выделенный сервер для обмена информацией по FTP. Входит в две физические сети “Network 1” и “Network 2”. Сервер осуществляет соединения в рамках сети “Network 1” только с Processing Server и только по протоколу FTP (firewall). Сервер осуществляет соединения в рамках сети “Network 2” только с фиксированным набором рабочих станций и только по протоколу FTP (firewall).
- **User 1/2/3** – рабочие станции в рамках сети “Network 2”, которым разрешен доступ на FTP dropbox.

Передача данных в / из KGS осуществляется только посредством файлов. Все файлы, в свою очередь, пересылаются только через FTP dropbox.

3.3. Системные требования

3.3.1. FTP server

3.3.1.1. Программное обеспечение

- Эталонный образ Debian 11.4, 64bit (см. гл.4 Рекомендации по установке операционных систем).

3.3.1.2. Аппаратное обеспечение

- Требований нет.

3.3.2. Database Server

3.3.2.1. Программное обеспечение

- Эталонный образ Debian 11.4, 64bit (см. гл.4 Рекомендации по установке операционных систем).
- СУБД PostgreSQL 13.7 (последней версии)).
- Python 3 64bit.

> ⚠ Python3 входит в состав эталонного образа ("устанавливается из коробки").

3.3.2.2. Аппаратное обеспечение

- Требований нет.

3.3.3. Processing Server

3.3.3.1. Программное обеспечение

- Эталонный образ Debian 11.4, 64bit (см. гл.4 Рекомендации по установке операционных систем).
- Python 3 64bit.

> ⚠ Python3 входит в состав эталонного образа ("устанавливается из коробки").

- программно-аппаратная система защиты HASP (HASP-driver и HASP-ключ).

3.3.3.2. Аппаратное обеспечение

- HASP-USB.

3.3.4. Требования по безопасности

Защита системы от несанкционированного доступа обеспечивается с помощью специальной системы развертывания, при которой компоненты KGS и база данных находятся в одной физической сети, а пользователи системы – в другой. Доступ из одной сети в другую осуществляется с помощью сервера FTP dropbox. Сервер осуществляет соединения только с фиксированным набором рабочих станций и только по протоколу FTP (firewall).

Database Server и Processing Server должны находиться в закрытом помещении с системой контроля доступа. Пользователь имеет доступ к KGS только с помощью KVM, который подключен к Processing Server в закрытом помещении.

3.3.5. Требования к квалификации обслуживающего персонала

Администратор KGS должен:

- обладать теоретическими знаниями и практическим опытом работы с ОС Debian;
- обладать теоретическими знаниями и практическим опытом работы с СУБД PostgreSQL (язык plsql);
- знать структуру БД KGS (KMI_DB_SCH);
- иметь общее представление о системе KGS.

Перейти к Содержанию...

4. Рекомендации по установке операционных систем

4.1. DB Server

Для DB Server при инсталляции ОС рекомендуется разбить HDD на 3 раздела: 1-й – для ОС, 2-й – для файлов postrges, 3-й – для файлов с бекапами, wal-файлов и tablespace базы. Размер 1-го раздела следует выбирать достаточным для ОС, исходя из особенностей ОС Debian, размер 2-го раздела (для файлов СУБД PostgreSQL) – не менее 100 ГБ, объем 3-го раздела – все остальное пространство, но не менее 400 ГБ.

4.2. Processing Server

При установке Processing Server количество разделов на HDD – не имеет значения (достаточно использовать один раздел).

Перейти к Содержанию...

5. Предварительные действия

Введение

Процедуры установки и настройки системы KGS приведены ниже.

Здесь и далее предполагается, что система KGS будет развернута на трёх серверах:

- FTP
- DB Server
- Processing Server

В связи с этим установка и настройка компонентов KGS описана аналогичным образом.

Действия рекомендуется выполнять в указанном порядке, тем не менее, некоторые процедуры могут быть выполнены в относительно любой момент времени.

Указанные особенности приведены в соответствующих подразделах.

Предварительные действия

До установки KGS на сервера в общем случае нужно выполнить следующие действия:

1. Получить IP-адреса машин (серверов), на которых будет развернут KGS.
2. Завести пользователей.
3. Настроить firewalls на серверах.

Перейти к Содержанию...

6. Настройка FTP server

6.1. Настройка операционной системы

Самым простым способом установки ОС является её установка из специально подготовленного образа *Debian **X**.iso*, который содержит дистрибутив самой ОС Debian, а также многие системные пакеты. Подробнее см. по **ссылке** .

6.1.1. Расширение репозиториев

Расширение репозиториев необходимо выполнить на ВСЕХ серверах (Database Server, Processing Server, FTP-Server).

Все операции выполнять под *sudo*.

Последовательность действий:

1. На ВСЕ сервера, которые будут использоваться KGS, необходимо предварительно установить ОС Debian 11.4 x64.
2. Подключить репозиторий производителя Системы debian.gs-labs.tv, содержащий необходимые системные пакеты (доступ к репозиторию предоставляется по запросу).
   Для подключения репозитория:
   a. Откройте файл *sources.list* для редактирования, например, в редакторе *nano*:

   ```
   sudo nano /etc/apt/sources.list
   ```

   b. В конце файла добавьте строку:

   ```
   deb http://debian.gs-labs.tv/ amd64/
   ```

   c. Сохраните изменения в файле (CTRL+O) и закройте редактор (CTRL+X).
3. Установить необходимые пакеты с помощью команды вида:

   ```
   sudo apt-get install [packet_name]
   ```

В процессе эксплуатации системы возможно обновление установленных системных пакетов. Для этого необходимо последовательно выполнить две команды:

```
sudo apt-get update
sudo apt-get upgrade
```

6.1.2. Установка дополнительных утилит

⚠ Процедура выполняется на всех серверах.

Данные программные пакеты устанавливаются для удобства установщика. Их перечень может быть изменен.

Утилиты, которые должны быть установлены на FTP server:

- sudo curl iptables ssh
- openssh-client=1:8.4p1-5 zlib1g=1:1.2.11.dfsg-2 libc6=2.31-13+deb11u2
- openssh-server libarchive13 libpython3.9
- nano wget mc ntpdate

Последовательность действий:

1. (Под root пользователем) установить sudo:

```
apt-get install sudo
```

2. Установить (от sudo) утилиты:

```
sudo apt-get install curl iptables ssh
```

3. Понизить версии пакетов (они включены в эталонный образ, но для установки других пакетов необходимо сделать даунгрейд):

```
sudo apt-get install openssh-client=1:8.4p1-5 zlib1g=1:1.2.11.dfsg-2 libc6=2.31-13+deb11u2
```

⚠ Процедура даунгрейда должна быть выполнена перед установкой других пакетов.

4. (**Обязательно**) после даунгрейда клиента установить openssh-server:

```
sudo apt-get install openssh-server
```

5. Установить пакеты:

```
sudo apt-get install libarchive13 libpython3.9
```

6. Установить nano editor, wget (на запросы системы нажимать y):

```
sudo apt-get install nano wget mc ntpdate
```

6.1.3. Настройка времени/часовых поясов на серверах

⚠ Процедура выполняется на BCEX серверах: Processing Server, DB Server, FTP-Server.

Все операции выполнять под *sudo*.

Последовательность действий:

1. Проверить текущий часовой пояс, установленный на машине, выполнив команду:

```
date
```

2. На экране появится строка вида:

```
Tue Feb 17 23:31:00 CST 2009
```

3. Сделать резервную копию существующего файла временной зоны (часового пояса):

```
sudo mv /etc/localtime /etc/localtime.bak
```

4. Создать ссылку на необходимую временную зону:

```
sudo ln -s /usr/share/zoneinfo/Europe/Moscow /etc/localtime
```

5. Выполнить команду:

```
sudo dpkg-reconfigure tzdata
```

6. Команда вызывает псевдографический интерфейс, в котором выбрать регион (Европа) и город. После их выбора интерфейс автоматически закроется.

При необходимости можно установить значение времени вручную (MM – месяц, DD – день, hh – час, mm – минуты):

```
sudo date MMDDhhmm
```

6.1.4. Настройка NTPDATE

⚠ Процедура выполняется на BCEX серверах: Processing Server, DB Server, FTP-Server.

Цель – синхронизация времени по расписанию, каждые 12 часов.

Все операции выполнять под *sudo*.

Последовательность действий:

1. Установить ntpdate:

```
sudo apt-get install ntpdate
```

2. Добавить в расписание (cron) для ежедневной синхронизации:

> После выполнения команды **crontab -e** вы окажетесь в текстовом редакторе (редактор задан по умолчанию, в данном случае - **vi**), где сможете вводить текст сценария для cron. Краткая справка по редактору **vi**:
>
> - для вставки текста нажмите **i**, затем вводите текст
> - для удаления символов нажмите **ESC**, а затем наберите **x**
> - для выхода из vi без сохранения изменений нажмите **ESC**, а затем наберите **:q!**
> - для сохранения и выхода нажмите **ESC**, а затем наберите **:wq**
>
> Задания для cron пишутся по одному в строке. После каждой строки, в том числе после последней или единственной, обязательно нужно нажать Enter - иначе задания работать не будут.
>
> Задание для cron выглядит как строка, в начале находятся пять обязательных полей для указания периодичности задания, а далее следует команда, которую нужно запускать:
>
> *поле1 поле2 поле3 поле4 поле5 команда*
>
> Значения первых пяти полей:
>
> a. минуты — число от 0 до 59
> b. часы — число от 0 до 23
> c. день месяца — число от 1 до 31
> d. номер месяца в году — число от 1 до 12
> e. день недели — число от 0 до 7 (0-Вс,1-Пн,2-Вт,3-Ср,4-Чт,5-Пт,6-Сб,7-Вс)
>
> Для каждого конкретного параметра можно задать несколько значений через запятую. Например, если в поле "часы" написать **1,4,22**, то задание будет запущено в 1 час ночи, в 4 часа утра и в 22 часа. Можно задать интервал - **4-9** будет означать, что программу нужно запускать каждый час в период с 4 до 9 часов включительно. Символ '*****' означает "все возможные значения". Например, указание '*****' в поле "часы" будет означать "запускать каждый час". Символ '**/**' служит для указания дополнительной периодичности задания. Например, '***/3**' в поле "часы" означает "каждые три часа".

- запустить crontab:

```
sudo crontab -e
```

- вставить в crontab строку:

```
* */12 * * * /etc/updatetime.sh
```

3. Создать файл */etc/updatetime.sh* с правом на исполнение:

```
sudo touch /etc/updatetime.sh
sudo chmod +x /etc/updatetime.sh
```

4. Открыть файл */ etc/updatetime.sh* на редактирование:

```
sudo nano /etc/updatetime.sh
```

5. Добавить строку в созданный файл (в строке указан IP-адрес сервера, с которым осуществляется синхронизация времени в компании производителя Системы):

```
/usr/sbin/ntpdate -s 192.168.12.129
```

6.1.5. Задание имени сервера

Рекомендуется задать серверу понятное имя hostname, например: kgs-ftp.

⚠ Процедура выполняется на всех серверах.

Все операции выполнять под *sudo*.

В общем случае шаги по изменению имени хоста следующие:

1. Узнать текущее имя сервера можно, выполнив команду:

```
cat /etc/hostname
```

2. Отредактировать файл */etc/hostname*, заменив в нем имя сервера (hostname) на нужное.

```
sudo nano /etc/hostname
```

3. Прочитанное имя устанавливается во время работы init-скрипта */etc/hostname* , а в некоторых версиях - */etc/hostname.sh* . Поэтому необходимо перезапустить скрипт:

```
sudo /etc/hostname
```

либо:

```
sudo /etc/hostname.sh
```

4. Активировать демона **hostname**:

```
/etc/hostname.sh start
```

> Для того чтобы изменения вступили в силу, можно вместо выполнения команд, описанных на шагах 3 и 4, выполнить перезагрузку сервера.

5. Проверить файл */etc/hosts* на предмет упоминания в нем прежнего имени. Если там присутствует прежнее имя сервера, то:
    a. Заменить в */etc/hosts* прежнее имя сервера на новое.
    b. Переподнять сеть:

```
sudo /etc/init.d/networking restart
```

Перейти к Содержанию...

6.2. Установка и настройка proftp

> Процедура выполняется на FTP-server, в любое время, но до запуска системы.

> Настройку FTP-сервера должна осуществляться системным администратором. Здесь и далее приведены общие **рекомендации** по установке FTP демона.

Последовательность действий:

1. Установить proftp:

```
sudo apt-get install proftpd
```

2. Открыть для редактирования файл:

```
nano /etc/proftpd/proftpd.conf
```

3. Внести в файл следующие данные:

```
# Includes DSO modules
Include /etc/proftpd/modules.conf

# If set on you can experience a longer connection delay in many cases.
IdentLookups off

ServerName "KMI_FTP"
ServerType standalone
DeferWelcome off

MultilineRFC2228 on
DefaultServer on
ShowSymlinks on

TimeoutNoTransfer 600
TimeoutStalled 600
TimeoutIdle 1200
```

```
DisplayLogin welcome.msg
DisplayChdir .message true
ListOptions "-l"

DenyFilter \*.*/

# Use this to jail all users in their homes
DefaultRoot ~
RequireValidShell off
AuthUserFile /etc/proftpd/ftpd.passwd

Port 21

# This is useful for masquerading address with dynamic IPs:
# refresh any configured MasqueradeAddress directives every 8 hours
<IfModule mod_dynmasq.c>
# DynMasqRefresh 28800
</IfModule>

MaxInstances 30

# Set the user and group that the server normally runs at.
User proftpd
Group nogroup

# Umask 022 is a good standard umask to prevent new files and dirs
# (second parm) from being group and world writable.
Umask 000 000
# Normally, we want files to be overwriteable.
AllowOverwrite yes

TransferLog /var/log/proftpd/xferlog
SystemLog /var/log/proftpd/proftpd.log

<IfModule mod_quotatab.c>
QuotaEngine off
</IfModule>

<IfModule mod_ratio.c>
Ratios off
</IfModule>


# Delay engine reduces impact of the so-called Timing Attack described in
# http://www.securityfocus.com/bid/11430/discuss
# It is on by default.
<IfModule mod_delay.c>
DelayEngine on
</IfModule>

<IfModule mod_ctrls.c>
ControlsEngine off
ControlsMaxClients 2
ControlsLog /var/log/proftpd/controls.log
ControlsInterval 5
ControlsSocket /var/run/proftpd/proftpd.sock
</IfModule>

<IfModule mod_ctrls_admin.c>
AdminControlsEngine off
</IfModule>

Include /etc/proftpd/conf.d/
```

4. Создать логины (пример для *user2*):

```
sudo touch /etc/proftpd/ftpd.passwd
sudo ftpasswd -passwd --file=/etc/proftpd/ftpd.passwd --name=user2 --shell=/bin/false --home /opt/kmi_ftp
/user2/ --uid=119 --gid=65500
```

Перейти к Содержанию...

6.3. Создание пользователей и каталогов

⚠ Процедура выполняется на Processing Server, DB Server и FTP-Server.

6.3.1. Общая структура папок, используемая в KGS

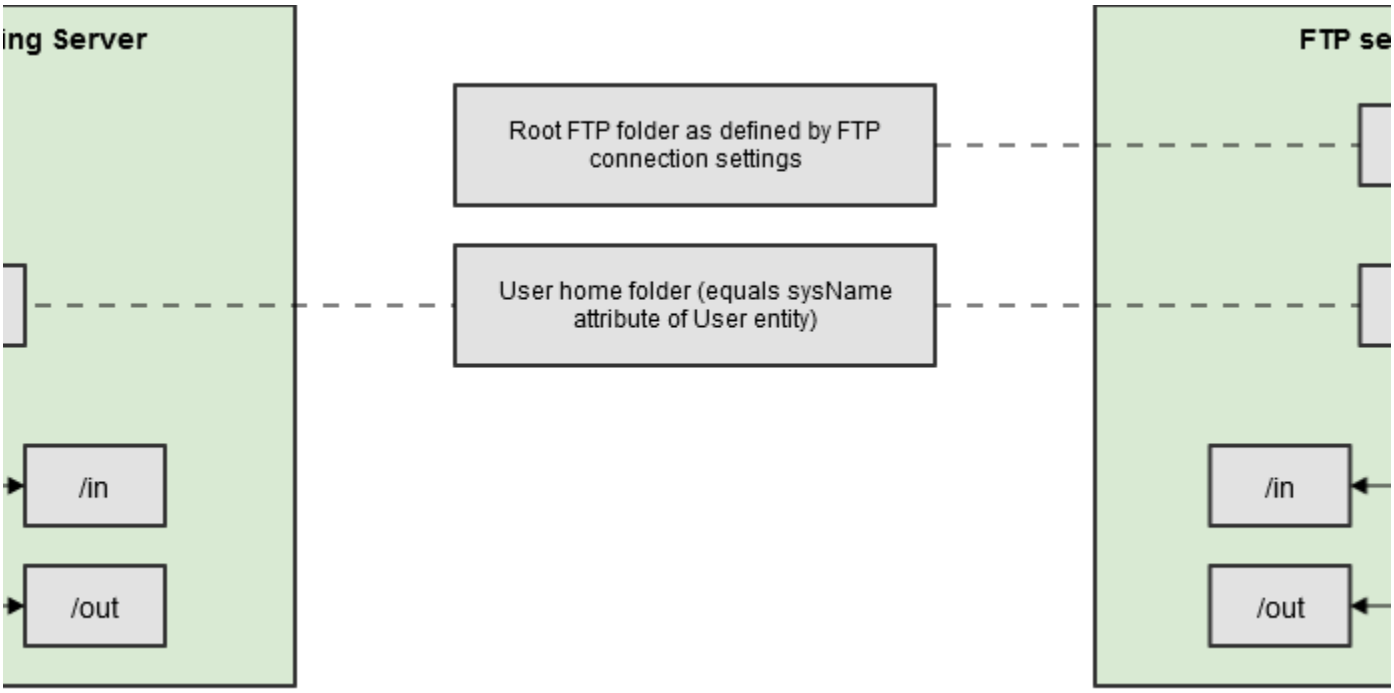
Предполагается следующая структура каталогов при разворачивании системы:

- Все файлы компонентов системы KGS (исполняемые файлы, библиотеки, файлы скриптов и т.д.) будут располагаться в каталоге */opt/kmi*. В данном каталоге будет создана структура папок, соответствующая названиям компонентов системы (kmi_dal, console и т.д.), библиотеки будут находиться непосредственно в каталоге *kmi_files*.
- Домашние каталоги пользователей системы KGS будут находиться на шифрованном разделе в оперативной памяти сервера. В данных директориях будут создаваться временные файлы с ключами в процессе работы Workflow, также через данные каталоги будет происходить обмен файлами с FTP-сервером. Подробности о структуре папок пользователей на Processing Server и на FTP Server – см. в разделе Структура папок пользователей.
- На FTP-сервере корневой каталог FTP будет располагаться по пути */opt/kmi/kmi_ftp*, в нем будут созданы папки пользователей системы (см. структуру папок в разделе Структура папок пользователей).
- Файлы СУБД с данными базы данных KMI_DB – в каталоге */opt/kmi_tablespace*
- Файлы с бекапами базы данных будут создаваться на DB Server в папке */var/backups*. Файлы бекапа будут перемещаться из данной папки на FTP-сервер, при проблемах передачи на FTP файлы будут оставаться в этой папке.
- Временные файлы базы данных, файлы WAL режима архивации БД – в каталоге */tmp* в соответствующих подкаталогах. Подробности описаны в Настройка режима бекапирования СУБД.
- Файлы с логами компонентов – в каталоге */var/log/kmi*.

6.3.2. Структура папок пользователей

Для экспорта / импорта данных на Processing server и FTP-сервере должны быть созданы папки и настроены права пользователей.

Общая схема необходимых директорий приведена на рисунке ниже.



Настройка папок производится администратором KGS и обновляется при каждом создании/удалении пользователя в консоли KGS.

> ⚠ Как правило, используется иерархическая структура папок БЕЗ элемента workflow_folder, т.е. вместо */user_folder/workflow**N**_folder/in(out)* используется */ user_ folder/ in(out)*.
>
> Структура меняется администратором KGS путем изменения настроечных параметров для соответствующих операций (workflows).

6.3.3. Исходные пользователи

На момент установки системы планируется создать следующих пользователей:

- *kmiadmin*: администратор KGS, обладает правами суперпользователя на все директории и операции в KGS, под этим пользователем настраиваются все компоненты KGS (в ОС на Processing Server и DB Server). Пользователь kmiadmin также создается в СУБД, этот пользователь является владельцем базы и администратором базы KMI_DB, под этим пользователям подключается DAL к базе KMI_DB. Пользователь создается автоматически при развертывании KMI_DB.
- *backup*: пользователь с данным именем создается в системе KGS, не имеет доступа в интерфейс, а используется только для создания/передачи бекапов БД на FTP-сервер.

6.3.4. Общий алгоритм создания пользователей и папок на FTP-Server

В общем случае последовательность действий следующая:

1. Создать пользователя с именем *username*:

```
sudo useradd <username>
```

2. Задать пароль пользователя с именем *username*:

```
sudo passwd <username>
```

3. Повторно ввести пароль для подтверждения.
4. Создать иерархию папок пользователя:

```
sudo mkdir -p /opt/kmi_ftp/<username>/{in,out}
```

5. Выдать пользователю права доступа на созданные папки:

```
sudo chmod 776 /opt/kmi_ftp/<username> -R
sudo chown <username>:<username> /opt/kmi_ftp/<username> -R
```

После создания пользователя и выдачи ему прав доступа рекомендуется загрузить public PGP-ключ пользователя в KMI_DB. Ключи пользователя *username* загружаются из папки */in* текущего пользователя (т.е. пользователя, под которым выполняется операция в KMI_CONSOLE), а не USERNAME. Каждый пользователь, который будет экспортировать данные на FTP-сервер, должен иметь один или несколько PGP-ключей.

6.3.5. Создание директорий на FTP-Server

Директории создавать от sudo:

```
sudo mkdir /opt/kmi_ftp
sudo mkdir /opt/kmi_ftp/backup
sudo mkdir /opt/kmi_ftp/user1
sudo mkdir /opt/kmi_ftp/user2
```

У каждого пользователя должна быть своя папка: user1 – */opt/kmi_ftp/user1*, user2 – */opt/kmi_ftp/user2* и т.д.

Пользователь kmiadmin должен иметь home-директорию "--home */opt/kmi_ftp*", т.е. корневой каталог (для остальных пользователей).

6.3.6. Создание пользователей на FTP-Server

Необходимо установить proftp (см. Установка и настройка proftp).

Действия, описанные ниже, необходимо выполнить для КАЖДОГО пользователя. В приведенном примере используется имя пользователя (*user 2*), его необходимо заменить на выбранное значение.

1. Выполнить команды:

```
sudo touch /etc/proftpd/ftpd.passwd
sudo ftpasswd -passwd --file=/etc/proftpd/ftpd.passwd --name=user2 --shell=/bin/false --home /opt/kmi_ftp
/user2/ --uid=119 --gid=65500
```

2. Система запросит пароль для *user2*, ввести пароль и подтвердить его.

Перейти к Содержанию...

7. Настройка DB Server

7.1. Настройка операционной системы

Аналогично описанному выше.

7.1.1. Расширение репозиториев

Аналогично описанному выше.

7.1.2. Установка дополнительных утилит

⚠ Процедура выполняется на всех серверах.

Данные программные пакеты устанавливаются для удобства установщика. Их перечень может быть изменен.

Утилиты, которые должны быть установлены на DB Server:

- Аналогично FTP server:
  - sudo curl iptables ssh
  - openssh-client=1:8.4p1-5 zlib1g=1:1.2.11.dfsg-2 libc6=2.31-13+deb11u2
  - openssh-server libarchive13 libpython3.9
  - nano wget mc ntpdate
- Дополнительно:
  - lshw libjsoncpp24
  - libtool unixodbc
  - libpgm-5.3-0 libsodium23 libzmq5
  - libboost1.74-all-dev
  - pigz

ⓘ Утилита lshw необходима для генерации Binding key, используемого в лестнице ключей, – без неё KGS работать не будет.

Пакеты libtool и unixodbc необходимы для работы KMI_FW_DAL; пакеты libpgm-5.3-0, libsodium23, libzmq5 устанавливаются на оба сервера; пакет libjsoncpp24 - для работы KMI_FW_DAL (см. Установка и настройка файлов KGS Framework (KMI_FW)).

pigz предназначен для ускорения процесса архивации на многоядерных системах.

Последовательность действий:

1. Установка тех же утилит и пакетов, что и на FTP server. См. здесь.
2. Установить lshw и libjsoncpp24:

```
sudo apt-get install lshw libjsoncpp24
```

3. Установить пакеты libtool и unixodbc (на запросы системы нажимать y):

```
sudo apt-get install libtool unixodbc
```

4. Установить пакеты libpgm-5.3-0, libsodium23, libzmq5:

```
sudo apt-get install libpgm-5.3-0 libsodium23 libzmq5
```

5. Установить пакеты libboost1.74-all-dev:

```
sudo apt-get install libboost1.74-all-dev
```

6. Установить pigz:

```
sudo apt update
sudo apt-get install -y pigz
```

7.1.3. Проверка наличия локали en_US.utf8 и ru_RU.UTF-8

> На серверах с ОС Debian кодировка UTF-8 должна быть установлена ПО УМОЛЧАНИЮ. Тем не менее, необходимо УДОСТОВЕРИТЬСЯ в том, что нужная локаль (en_US.utf8 и ru_RU.UTF-8) установлена на серверах.
>
> Использование русской кодировки на серверах KGS НЕ ПРИВЕТСТВУЕТСЯ, тем не менее, команда создания БД требует установленной русской локали.

> ⚠ Процедура выполняется на Processing Server и DB Server.

Подробное описание приведено здесь:

http://webhamster.ru/mytetrashare/index/mtb0/1355746267lougdkzfg3

Последовательность действий:

1. Проверить, какая локаль сейчас установлена на сервере:

```
locale -a
```

2. Результат команды будет выведен на экран. Если в списке нет ru_RU.UTF-8, то эту локаль надо добавить, выполнив дальнейшие действия.
Пример результата (в списке нет ru_RU.UTF-8, нужны дальнейшие действия):

```
C
C.UTF-8
en_US.utf8
POSIX
```

3. Выполните команду:

```
sudo dpkg-reconfigure -plow locales
```

4. Убедитесь, что в списке локализаций отмечены **en_US.UTF-8** и **ru_RU.UTF-8**. Если это не так, выберите её в добавок к уже имеющимся и нажмите *Ok*.
5. Проверьте, что вывод имеет вид:

```
Generating locales (this might take a while)...
   en_US.UTF-8... done
   ru_RU.UTF-8... done
Generation complete.

*** update-locale: Warning: LANGUAGE ("en_US:en") is not compatible with LANG (ru_RU.UTF-8). Disabling
it.
```

6. Вновь выполните команду *locale -a* (результат должен содержать en_US.utf8 и ru_RU.UTF-8):

```
locale -a

C
C.UTF-8
en_US.utf8
POSIX
ru_RU.utf8
```

7.1.4. Настройка времени/часовых поясов на серверах

Аналогично описанному выше.

7.1.5. Настройка NTPDATE

Аналогично описанному выше.

7.1.6. Настройка фаервола iptables

⚠ Процедура выполняется как на Processing Server, так и на DB Server.

На Database Server необходимо разрешить подключения к СУБД PostgreSQL и NFS. Так как DB Server будет доступен только для подключения со стороны Processing Server, можно разрешить доступ только с ip-адреса Processing Server без указания портов используемых сервисов, при необходимости можно ограничить список портов.

Настройки следует выполнять под правами суперпользователя (под *sudo*).

Последовательность действий:

1. Настроить все необходимые правила фаервола (например, закрыть все входящие соединения по умолчанию: *iptables -P INPUT DROP*). Конфигурация фаеврола остается на усмотрение администратора. Об использовании iptables в Debian можно почитать здесь: https://wiki.debian.org/iptables
2. На Processing Server необходимо открыть входящие с Database Server, **и наоборот** - на DB Server открыть входящие с Processing Server, например:

```
sudo iptables -A INPUT -s 192.168.14.160 -j ACCEPT
```

3. Для того чтобы изменения вступили в силу после перезагрузки, нужно установить пакет iptables-persistent:

```
sudo apt-get install iptables-persistent
```

4. Убедиться, что конфиги в */etc/iptables/* верны (по умолчанию будут записаны текущие правила).

7.1.7. Задание имени сервера

Рекомендуется задать серверу понятное имя hostname, например: kgs-db.

Аналогично описанному выше.

Перейти к Содержанию...

7.2. Настройка NFS и монтирование папки бекапов с DB Server на Processing Server

Требования:

- наличие NFS-server на DB Server;
- наличие NFS-client на Processing Server;
- Папка *'some_path_to_files_with_backups'* смонтирована на Processing Server в папку */var/backups/out* (фиксированный путь).

> ⚠ В приведенном ниже подразделе использован следующий IP-адрес для Processing Server – 192.168.14.160.

Подробное писание приведено здесь:

http://www.tecmint.com/how-to-setup-nfs-server-in-linux/

7.2.1. Настройка NFS-server на DB Server

1. Установить компоненты NFS:

```
sudo apt-get install nfs-kernel-server nfs-common portmap
```

2. Запустить startup-скрипты:

```
sudo /etc/init.d/rpcbind start
sudo /etc/init.d/nfs-kernel-server start
```

3. Создать папку /home/kmiadmin/backup_files на DB Server:

```
sudo mkdir /home/kmiadmin/backup_files
```

4. Задайте каталог экспорта в */ etc/exports*:

```
sudo nano /etc/exports
```

5. Добавить строку с IP-адресом Processing Server:

```
/home/kmiadmin/backup_files           192.168.14.160(rw,sync,no_root_squash,no_subtree_check)
```

> ⚠ Папка, в которую система KGS сохраняет бекапы БД, захардкожена.

6. Выполнить команду:

```
sudo exportfs -a
```

Перейти к Содержанию...

7.3. Установка и настройка БД

> ⚠ БД устанавливается на DB Server. Процедура должна быть выполнена ДО запуска DAL.

7.3.1. Установка PostgreSQL

Все операции выполнять под *sudo*.

Перед началом установки KMI_DB на Database Server необходимо установить СУБД PostgreSQL 13.7. Описание процедуры установки и настройки PostgreSQL выходит за рамки данного документа. Процедуру установки можно посмотреть, например, здесь: https://computingforgeeks.com/how-to-install-postgresql-14-on-debian/

В общем случае, последовательность действий следующая:

1. Подключить репозиторий производителя Системы, содержащий необходимые системные пакеты (было выполнено ранее, см. Расширение репозиториев).
2. Обновить список пакетов системы:

```
sudo apt-get update
```

3. Установить пакеты postgresql:

```
sudo apt-get install postgresql-13 postgresql-client-13 postgresql-contrib postgresql postgresql-common
postgresql-client-common
```

7.3.2. Настройка PostgreSQL

Все операции выполнять под *sudo*.

В PostgreSQL **ОБЯЗАТЕЛЬНО** надо внести следующие настройки:

1. Открыть для редактирования файл **postgresql.conf**:

```
sudo nano /etc/postgresql/13/main/postgresql.conf
```

2. Установить часовой пояс (timezone) такой же, как и в Debian (т.е. MSK):
   - изменить параметры timezone и log_timezone в */etc/postgresql/13/main/postgresql.conf*:

```
timezone = 'Europe/Moscow'
log_timezone = 'Europe/Moscow'
```

   - для просмотра времени и временной зоны в psql используются команды:

```
show timezone;
select now();
```

3. Внести параметры подключения сетевых машин (Processing Server) к локальной ЭВМ, на которой установлена СУБД (Database Server). Для этого необходимо:
   - найти строки:

```
# - Connection Settings –
[...]
#listen_addresses = 'localhost'
[...]
#port = 5432
[...]
```

   - раскомментировать строки, внести IP-адрес и порт PostgreSQL-сервера либо указать '*', чтобы слушать всех клиентов на данном порте:

```
# - Connection Settings -
[...]
listen_addresses = '*'
[...]
port = 5432
[...]
```

4. Открыть для редактирования файл **pg_hba.conf**:

```
sudo nano /etc/postgresql/13/main/pg_hba.conf
```

5. Внести параметры ЭВМ (Processing Server), для которых в дальнейшем будет разрешено подключение к БД (все остальные ЭВМ будут игнорироваться), настроить подключение к серверу. Необходимо, чтобы:
   К **postgres** могли подсоединиться любые процессы с локальной машины.
   Подключение к серверу без пароля (trust-режим) должно быть только для **postgres** и **kmiadmin**, для любых других подключений - ***md5*** либо ***peer***.
   В поле *host* (*IPv4 local connections*) внести БД (в данном случае - **kmi**) и пользователя (в данном случае - **postgres** и **kmiadmin**), которому будет разрешено подключение. Вместо **x.x.x.x/x** ввести IP-адрес/порт Processing Server либо только IP-адрес.
   Ipv6 local connections в KGS не используются (должны быть закомментированы).

   Таким образом, файл может выглядеть следующим образом:

```
[...]
# TYPE  DATABASE        USER            ADDRESS                 METHOD

# "local" is for Unix domain socket connections only
local   all             postgres                                trust
local   kmi             kmiadmin                                trust
# IPv4 local connections:
host    kmi             postgres        127.0.0.1/8             md5
host    kmi             kmiadmin        127.0.0.1/8             md5
host    kmi             kmiadmin        x.x.x.x/x               md5
[...]
```

Опытным путем установлено, что:

- для установки/обновления KMI_DB нужны строки: *host kmi postgres 127.0.0.1/8 md5*; *host kmi kmiadmin 127.0.0.1/8 md5* (см. выше).

- если не указать строку *local kmi kmiadmin trust* (см. выше), то не работают некоторые workflows (например, выгрузка pgp ключей). При этом должен быть trust-доступ (для md5 будет ошибка).

В приведенном выше примере вместо **x.x.x.x/x** необходимо ввести (без пробелов) IP-адрес /маску Processing Server, от которого будет устанавливаться соединение.

Маска задается в формате CIDR (количество бит в маске подсети, 255=8бит). Например, чтобы разрешить подключение localhost по ipv4: 127.0.0.1/32 (32=255.255.255.255); чтобы разрешить локальной сети: 10.126.9.0/24 (24=255.255.255.0) или 172.16.0.0/16 (если маска 255.255.0.0).

В случае нахождения за маршрутизатором необходимо указывать адрес шлюза.

Настоятельно рекомендуется "минимизировать" маску для разрешенного хоста. Для того, чтобы разрешить доступ с одного хоста, необходимо использовать маску 32.

Для повышения безопасности системы настоятельно рекомендуется вместо *trust*-доступа (method) выбрать метод выше чем *trust*, например, *md5*.

6. Сохранить изменения, перезапустив PostgreSQL:

```
sudo /etc/init.d/postgresql restart
```

При отсутствии указанных настроек в PostgreSQL база данных KGS не будет корректно работать.

7.3.3. Создание директории для Tablespace

Tablespace должно быть создано до установки KMI_DB_SCH и KMI_DB_API.

Создать, если это не было сделано ранее (см. Общая структура папок, используемая в KGS, п.4), на DB-Server хранилище – папку */opt/kmi_tablespace:*

```
sudo mkdir /opt/kmi_tablespace
sudo chown postgres:postgres /opt/kmi_tablespace -R
```

7.3.4. Редактирование файла по начальному наполнению БД

Перед установкой KMI_DB следует проверить конфигурацию файла, отвечающего за начальное наполнение БД.

Последовательность действий:

1. Открыть для редактирования файл *INIT_DATA**init_data.sql*** (x.x.x - версия *KMI_DB_API*):

   ```
   sudo nano /tmp/kmi_db_api_x.x.x/INIT_DATA/init_data.sql
   ```

2. Внести в файл параметры FTP-dropbox и имена пользователей:

   ```
   dmz_ftp_url   varchar(80)    = 'DMZ_FTP_URL';
   dmz_ftp_login varchar(80)    = 'DMZ_FTP_LOGIN';
   dmz_ftp_password varchar(80) = 'DMZ_FTP_PASSWORD';

   user1_name varchar(80)      = 'USER1';
   user1_sysname varchar(80)   = 'user1';
   user2_name varchar(80)      = 'USER2';
   user2_sysname varchar(80)   = 'user2';
   ```

   Параметры имеют следующие значения:

   a. dmz_ftp_url = 'URL DMZ-ресурса'
   b. dmz_ftp_login = 'имя пользователя, используемого для загрузки и выгрузки данных на FTP-сервер'
   c. dmz_ftp_password = 'пароль доступа'
   d. user1_name = 'имя 1-го пользователя Linux, используемого в workflow'
   e. user1_sysname = 'имя 1-го пользователя, который подключается к Linux'
   f. user2_name = 'имя 2-го пользователя Linux, используемого в workflow'
   g. user2_sysname = 'имя 2-го пользователя, который подключается к Linux'

   ⓘ Указанные параметры будут занесены в БД KGS. После выполнения скрипта **init_data** их можно будет изменить, но только подключившись к базе.

7.3.5. Установка KMI_DB_SCH

ⓘ В данном разделе описана установка KMI_DB_SCH "с нуля".

⚠ ВНИМАНИЕ! В случае загрузки с dump, устанавливать db_sch и db_api не нужно.

Особенности:

- В рамках дальнейших алгоритмов подразумевается, что на сервер уже установлен и настроен Postgres, созданы роли и табличное пространство.
- Скрипты запускаются от sudo (не от postgres, как раньше).
- Необходимо выдать права на папку с файлами, например сделать папку открытой для всех пользователей (наиболее простой вариант):

```
sudo chmod -R 755 /home/kmi_db_sch
sudo chmod -R 755 /home/kmi_db_api
```

- Для установки sch и api вне зависимости от того, устанавливаются они на уже существующие компоненты (обновляются) или при установке на чистый сервер, необходимо запускать один и тот же единственный скрипт (внутри скрипта происходит запуск нужных скриптов (либо для обновления, либо для изначальной установки)).
- Пароли пользователей KMI_DB (kmiadmin) и postgres могут быть как переданы напрямую (указаны в открытом виде в bash строке), так и переданы в файлах (в которых уже внутри будут пароли для соответствующего пользователя).

Последовательность действий:

1. Перейти в папку с распакованной сборкой kmi_db_sch (в нашем примере это /home/kmi_db_sch):

```
cd /home/kmi_db_sch
```

2. **Если в процессе установки пароли будут указаны в открытом виде**, то убедиться, что внутри директории со скриптами check_install_***.sh **отсутствуют** файлы, совпадающие по имени с паролем.
3. **Если в процессе установки нежелательно указывать пароли в открытом виде**, то:
    a. Создать файлы (один для хранения пароля пользователя kmiadmin (user password), один - для хранения postgres password), например:

```
echo kmiadmin > kmiadminpass
echo postgres > postgrespass
```

    b. Выдать пользователю, который запускает скрипты установки check_install_***.sh, разрешение на доступ к файлам (права на чтение файла).
4. Запустить скрипт check_install_sch.sh (скрипт лежит в подпапке common_db) со следующими параметрами:

```
sudo bash check_install_sch.sh $1 $2 $3 $4 $5 $6 $7 $8 $9
```

    где:

    a. $1 - host name
    b. $2 - port
    c. $3 - db name

d. $4 - user name
e. $5 - user password / file with user password
f. $6 - postgres user
g. $7 - postgres password / file with postgres password
h. $8 - DB scheme
i. $9 - extra parameters

5. Пример (пароли передаются в открытом виде):

```
sudo bash check_install_sch.sh 127.0.0.1 5432 kmi kmiadmin kmiadmin postgres postgres kmi
'TBS_TBL=kmi_tablespace'
```

6. Пример (пароли передаются через файлы):

```
echo kmiadmin > kmiadminpass
echo postgres > postgrespass
sudo bash check_install_sch.sh 127.0.0.1 5432 kmi kmiadmin kmiadminpass postgres postgrespass kmi
'TBS_TBL=kmi_tablespace'
```

7. Проверить файл install.log (в подпапке common_db) на успешность выполнения (не должно быть error сообщений).

7.3.6. Установка KMI_DB_API

ⓘ Процедуру следует выполнять только после установки KMI_DB_SCH.

Последовательность действий:

1. Перейти в папку с распакованной сборкой kmi_db_api (в нашем примере это /home/kmi_db_api):

```
cd /home/kmi_db_api
```

⚠ Далее предполагается, что пользователем уже принято решение о передаче паролей (в открытом виде / через файлы), а соответствующие действия / проверки уже были выполнены ранее (см. раздел Установка KMI_DB_SCH, шаги 2, 3).

2. Запустить скрипт check_install_api.sh (скрипт лежит в подпапке common_db) со следующими параметрами:

```
sudo bash check_install_api.sh $1 $2 $3 $4 $5 $6 $7 $8
```

где:

a. $1 - host name
b. $2 - port
c. $3 - db name
d. $4 - user name
e. $5 - user password / file with user password
f. $6 - postgres user
g. $7 - postgres password / file with postgres password

h. $8 - DB scheme

3. Пример (пароли передаются в открытом виде):

```
sudo bash check_install_api.sh 127.0.0.1 5432 kmi kmiadmin kmiadmin postgres postgres kmi
```

4. Пример (пароли передаются через файлы, см. раздел Установка KMI_DB_SCH, шаги 2, 3):

```
sudo bash check_install_api.sh 127.0.0.1 5432 kmi kmiadmin kmiadminpass postgres postgrespass kmi
```

5. Проверить файл install.log (в подпапке common_db) на успешность выполнения (не должно быть error сообщений).

Перейти к Содержанию...

7.4. Установка и настройка компонентов KGS на DB Server

7.4.1. Установка и настройка файлов KGS Framework (KMI_FW)

⚠ Установка компонентов KGS Framework, за исключением KMI_FW_BACKUP, осуществляется на Processing Server. Модуль KMI_FW_BACKUP устанавливается на DB Server.

Установка компонентов KGS Framework осуществляется с помощью DEB-пакетов ОС (Debian). Файлы на обоих серверах будут установлены в папку */opt/kmi/*.

ⓘ DEB-пакеты KMI_FW и KMI_CONSOLE используют библиотеки python, поэтому Python3 должен быть установлен до установки этих компонентов.

Поскольку Python3 входит в эталонный образ Debian11 ("устанавливается из коробки"), то дополнительные действия не требуются.

7.4.1.1. Установка компонентов на DB Server

1. Скопировать из репозитория на Database Server следующие DEB-пакеты (путь к папке значения не имеет):
   - kmi_fw_backup-**X.X.X**-linux-x86_64.deb
2. Перейти в папку, содержащую DEB-пакеты (в этом случае не нужно прописывать полный путь к пакетам, см. ниже).
3. Выполнить команды, заменив **X.X.X** на релизные версии компонентов (**выполнять только в указанной последовательности**):

```
sudo dpkg -i kmi_fw_backup-X.X.X-linux-x86_64.deb
```

⚠ Если при установке не обнаружены какие-либо библиотеки или пакеты установлены не в том порядке, то выполните команду: *sudo ldconfig*

4. Дождаться окончания выполнения операции. Загрузка каждого компонента должна составлять 100%.

⚠ При возникновении ошибок рекомендуется удалить компоненты и установить их заново.

Перейти к Содержанию...

7.4.1.2. Настройка конфигурационного файла KGS

⚠ Процедура выполняется после установки компонентов KGS Framework, но до запуска служб KGS Framework, на серверах, на которых установлены эти компоненты (т.е. на Processing Server и DB Server).

Последовательность действий:

1. Перейти в папку */opt/kmi/etc*.
2. Открыть для редактирования файл kmi_cfg.xml:

```
sudo nano kmi_cfg.xml
```

ⓘ **При установке KGS "с нуля"**: в папке */opt/kmi/etc* нет файла kmi_cfg.xml, поэтому kmi_cfg.xml.default автоматически копируется и переименовывается в kmi_cfg.xml.

При обновлении KGS: если в папке */opt/kmi/etc* перед обновлением присутствовал файл kmi_cfg.xml (без .default), то после обновления он не будет удален. Т.е. в нем будут сохранены и использоваться прежние настройки.

3. Внести в конфигурационный файл IP-адрес и номер порта, по которым будет осуществляться взаимодействие с компонентами PGP_server_route, KMI_FW_DAL, KMI_FW_BACKUP, KMI_FW_TRANSFER. В параметре *<PGP_server_route>* указываются IP-адрес и порт, по которым обращается KMI_FW_TRANSFER для получения PGP-ключей. **В случае KGS в теге *<PGP_server_route>* указать те же значения, что и в теге *<KMI_Dal_Route>***. Компоненты DAL и Transfer установлены на Processing Server, Backup - на DB Server. Строка *<IP-адрес:порт>* должна быть **уникальной** (не повторяться) для каждого параметра в одном файле. В теге <KMI_vendor_code> указать полный путь к файлу с vendor_code.

⚠ Файл с vendor_code требуется при работе KGS с установленным HASP-ключом (token). Если KGS эксплуатируется без HASP (no_hasp) или на данном сервере HASP не используется (отдельный DB Server), тег *<KMI_vendor_code>*.... игнорируется.

Если тег <KMI_vendor_code> не будет найден, путь к файлу vendor_code определяется "по-старому", т.е. *текущая рабочая директория + /etc/vendor_code*.

   a. Пример. Пусть Processing Server имеет адрес - 192.168.1.1, а DB Server - 192.168.2.2, путь к vendor_code - */opt/kmi/etc/vendor_code*, тогда настройки на Processing Server и на DB Server будут следующими:

```
<?xml version="1.0"?>
<Config>
    <PGP_server_route>tcp://192.168.1.1:4321</PGP_server_route>
    <KMI_Dal_Route>tcp://192.168.1.1:4321</KMI_Dal_Route>
    <KMI_Backup_Route>tcp://192.168.2.2:4322</KMI_Backup_Route>
    <KMI_Transfer_Route>tcp://192.168.1.1:4323</KMI_Transfer_Route>
    <KMI_vendor_code>/opt/kmi/etc/vendor_code</KMI_vendor_code>
</Config>
```

4. Выполнить настройку на втором сервере.

Перейти к Содержанию...

7.4.1.3. Настройка PATH

⚠ Процедура выполняется на Processing Server и DB Server.

ⓘ Все исполняемые файлы KGS хранятся в папке */opt/kmi/bin*. Для того чтобы не вводить этот каталог в процессе установки, настройки и использования KGS, необходимо настроить систему таким образом, чтобы она по умолчанию искала исполняемые файлы в этом каталоге при каждой сессии.

С этой целью в переменную PATH добавляется каталог */opt/kmi/bin*.

Выполнение этой операции необязательно, применяется исключительно для удобства дальнейшей установки системы.

Последовательность действий:

1. Открыть для редактирования .profile (редактируется тот файл, который лежит в каталоге пользователя, под которым осуществляется установка и настройка системы KGS):

```
nano ~/.profile
```

2. Внести в него следующие параметры:

```
export PATH="$PATH:/opt/kmi/bin"
```

3. Применить настройки без перелогинивания на сервере, выполнив команду:

```
source ~/.profile
```

4. Убедиться, что настройки заданы верно:
   a. Выполнить команду:

```
echo $PATH
```

   b. В результате должна быть выведена строка, содержащая **/opt/kmi/bin**, например:

```
/usr/local/bin:/usr/bin:/bin:/usr/local/games:/usr/games:/opt/kmi/bin
```

Перейти к Содержанию...

7.5. Проверка автоматического запуска компонентов KGS Framework (KMI_FW)

⚠ Настройка автоматического запуска компонентов выполняется автоматически при установке компонентов с помощью DEB-пакетов (см. "Установка и настройка файлов KGS Framework (KMI_FW)"). Ниже приведены дополнительные действия, которые нужно выполнить для проверки / обеспечения запуска.

7.5.1. Проверка автоматического запуска KMI_FW_BACKUP

⚠ Автоматический запуск KMI_FW_BACKUP настраивается автоматически при установке компонента с помощью DEB-пакетов (см. "Установка и настройка файлов KGS Framework (KMI_FW)").

Чтобы проверить автоматический запуск сервиса kmi_fw_backup, выполните команду:

```
systemctl is-enabled kmi_fw_backup
```

В случае успеха ответ должен быть следующим:

```
enabled
```

7.6. Настройка режима бекапирования СУБД

⚠ Процедура выполняется до запуска служб KGS, но ПОСЛЕ установки DEB-пакетов.

Необходимо настроить режим архивации БД для резервного копирования (ВСЕ операции выполняются под пользователем postgres). Последовательность действий:

1. Открыть для редактирования файл **postgresql.conf**:

   ```
   vi /etc/postgresql/13/main/postgresql.conf
   ```

2. В открытом файле (см. предыдущий шаг) */etc/postgresql/13/main/postgresql.conf* найти строки:

```
#------------------------------------------------------------------------
# WRITE AHEAD LOG
#------------------------------------------------------------------------

# - Settings -

#wal_level = minimal


# - Archiving -

#archive_mode = on
#archive_command = 'test ! -f /wal_archive/%f && cp %p /wal_archive/%f'
```

3. Заменить значения в строках на следующие:

```
#------------------------------------------------------------------------
# WRITE AHEAD LOG

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

# - Settings -

wal_level = archive


# - Archiving -

archive_mode = on
archive_command = 'test ! -f /var/lib/postgresql/start_backup || (test ! -f /var/lib/postgresql
/wal_archive/%f && cp %p /var/lib/postgresql/wal_archive/%f)'
```

4. Создать каталог и файл:

```
su postgres
mkdir -p /var/lib/postgresql/wal_archive
touch /var/lib/postgresql/start_backup
```

5. Сохранить изменения, перезапустив PostgreSQL:

```
sudo /etc/init.d/postgresql restart
```

6. Проверить файл **kmi_backup.sh.default** (*/opt/kmi/scripts/kmi_backup.sh.default*) – файл должен содержать внесенные изменения (директории для хранения архивов и бекапов):

```
PGDATA_ADD1=/opt/kmi_tablespace (relevant path to tablespace must be added)
...
WAL_ARCHIVE=/var/lib/postgresql/wal_archive
...
PGBACKUP=/home/kmiadmin/backup_files
...
```

> В параметре PGDATA_ADD1 указывается реальный путь к tablespace (вместо */home /kmi_tablespace* должно быть */opt/kmi_tablespace*). Путь к хранилищу определяется на этапе создания Tablespace (см. выше). Если путь в параметре PGDATA_ADD1 указан неверно, то бекапирование данных не производится, ВОССТАНОВЛЕНИЕ БД НЕВОЗМОЖНО.

> Файл появляется только после установки DEB-пакета с KMI_FW_BACKUP на DB Server.

7. Проверить файл **kmi_restore.sh.default** (*/opt/kmi/scripts/kmi_restore.sh.default*) – файл должен содержать внесенные изменения (директории для хранения архивов и бекапов). Все замечания, указанные на предыдущем шаге для kmi_backup.sh, справедливы и для скрипта восстановления.
8. Проверить файл **kmi_backup_scheme.sh.default** (*/opt/kmi/scripts/kmi_backup_scheme.sh.default*) – файл должен содержать директорию для хранения бекапов:

```
WORK_DIR=/home/kmiadmin/backup_files
```

> В файле, в команде *pg_dump -U* ... также указываются имя пользователя PostgreSQL (*postgres*) и БД KGS (*kmi*). При использовании других имен их эту строку также необходимо настроить.

9. Проверить файл **kmi_restore_scheme.sh.default** (*/opt/kmi/scripts/kmi_restore_scheme.sh.default*) – файл должен содержать директорию для хранения бекапов:

```
WORK_DIR=/home/kmiadmin/backup_files
```

> В файле, в команде *gunzip -c $WORK_DIR/$1 | psql -U* ... также указываются имя пользователя PostgreSQL (*postgres*) и БД KGS (*kmi*). При использовании других имен их эту строку также необходимо настроить.

10. **Установить** пакет параллельного архиватора **pbzip2**. При наличии доступа в сеть Интернет это можно сделать с помощью команды:

```
sudo apt-get install pbzip2
```

Перейти к Содержанию...

8. Настройка Processing Server

8.1. Настройка операционной системы

Аналогично описанному выше.

8.1.1. Расширение репозиториев

Аналогично описанному выше.

8.1.2. Установка дополнительных утилит

⚠ Процедура выполняется на всех серверах.

Данные программные пакеты устанавливаются для удобства установщика. Их перечень может быть изменен.

Утилиты, которые должны быть установлены на Processing Server:

- Аналогично FTP server:
  - sudo curl iptables ssh
  - openssh-client=1:8.4p1-5 zlib1g=1:1.2.11.dfsg-2 libc6=2.31-13+deb11u2
  - openssh-server libarchive13 libpython3.9
  - nano wget mc ntpdate
- Аналогично DB Server:
  - lshw libjsoncpp24
  - libtool unixodbc
  - libpgm-5.3-0 libsodium23 libzmq5
  - libboost1.74-all-dev
- Дополнительно:
  - logrotate, компилятор *C* (-y gcc autoconf automake)
  - haveged (libhavege1_1.9.1-7_amd64.deb)
  - libcurl4 libqt5opengl5
  - libgpgme11
  - libarchive for zip64 (libarchive-dev-3.3.2-amd64.deb)

ⓘ Утилита lshw необходима для генерации Binding key, используемого в лестнице ключей, – без неё KGS работать не будет.

Демон haveged необходим для генерации PGP-ключей заданного размера - при его отсутствии KMI_CONSOLE зависнет на этапе генерации PGP-ключа.

Пакеты libtool и unixodbc необходимы для работы KMI_FW_DAL; пакеты libpgm-5.3-0, libsodium23, libzmq5 устанавливаются на оба сервера; пакеты libcurl4 и libqt5opengl5 - для работы KMI_FW_TRANSFER; пакет libjsoncpp24 - для работы KMI_FW_DAL (см. Установка и настройка файлов KGS Framework (KMI_FW)).

Последовательность действий:

1. Установка тех же утилит и пакетов, что и на FTP server. См. здесь.
2. Установка тех же утилит и пакетов, что и на DB server. См. здесь.
3. Установить на Processing Server утилиту logrotate, компилятор C:

```
sudo apt-get install logrotate
sudo apt-get install -y gcc autoconf automake
```

4. Установка и запуск демона haveged:

> В репозиториях эталонного образа пакеты с такими версиями отсутствуют. В связи с этим их установка осуществляется с официального репозитория Debian.

> В зависимости от архитектуры сервера, нужно скачивать и устанавливать пакеты с amd64 либо i386.
>
> Ниже приведен пример для amd64.

   a. Скачать deb-пакет libhavege1_1.9.1-7, перейдя по ссылке http://ftp.debian.org/debian/pool/main/h/haveged/
   b. Перейти в папку, содержащую DEB-пакет (в этом случае не нужно прописывать полный путь к пакету).
   c. Установить пакет libhavege1, необходимый для работы haveged:

```
sudo -E dpkg -i libhavege1_1.9.1-7_amd64.deb
```

   d. Скачать deb-пакет с демоном, перейдя по ссылке http://ftp.debian.org/debian/pool/main/h/haveged/
   e. Перейти в папку, содержащую DEB-пакет (в этом случае не нужно прописывать полный путь к пакету).
   f. Установить haveged из deb-пакета:

```
sudo -E dpkg -i haveged_1.9.1-7_amd64.deb
```

   g. Запустить службу:

```
sudo service haveged start
```

> Запуск haveged выполняется на Processing Server ДО запуска служб KGS Framework.

5. Установить пакеты libcurl4 и libqt5opengl5:

```
sudo apt-get install libcurl4 libqt5opengl5
```

6. Установить libgpgme11 (требуется при запуске KMI_FW_DAL):

```
sudo apt-get install libgpgme11
```

7. Установить пакет libarchive for zip64:

   a. Скопировать пакет ***libarchive-dev-3.3.2-amd64.deb*** на Processing Server.
   b. Установить пакет:

```
sudo dpkg -i libarchive-dev-3.3.2-amd64.deb
```

8.1.3. Проверка наличия локали en_US.utf8 и ru_RU.UTF-8

Аналогично описанному выше.

8.1.4. Настройка времени/часовых поясов на серверах

Аналогично описанному выше.

8.1.5. Настройка NTPDATE

Аналогично описанному выше.

8.1.6. Настройка фаервола iptables

Аналогично описанному выше.

8.1.7. Задание имени сервера

Рекомендуется задать серверу понятное имя hostname, например: kgs-processing.

Аналогично описанному выше.

Перейти к Содержанию...

8.2. Создание пользователей и каталогов

8.2.1. Общие сведения

Аналогично описанному выше:

- Структура папок пользователей - см. здесь.
- Исходные пользователи - см. здесь.

8.2.2. Создание пользователей

⚠ Пользователи и папки на Processing Server создаются с помощью скрипта (см. Создание пользователей KGS Console), доступного после установки файлов KGS Framework.

Перейти к Содержанию...

8.3. Настройка NFS и монтирование папки бекапов с DB Server на Processing Server

Требования:

- наличие NFS-server на DB Server;
- наличие NFS-client на Processing Server;
- Папка *'some_path_to_files_with_backups'* смонтирована на Processing Server в папку */var/backups/out* (фиксированный путь).

  ⚠ В приведенном ниже подразделе использован следующий IP-адрес для DB Server – 192.168.14.162.

Подробное писание приведено здесь:

http://www.tecmint.com/how-to-setup-nfs-server-in-linux/

8.3.1. Настройка NFS-client на Processing Server

1. Установить компоненты NFS:

   ```
   sudo apt-get install nfs-common portmap
   ```

2. Убедиться, что существует каталог */var/backups/out*, в который помещаются файлы с DB Server. Если каталог отсутствует, то его необходимо создать:

   ```
   sudo mkdir /var/backups
   sudo mkdir /var/backups/out
   ```

3. Настроить автоматическое монтирование в fstabs:
   - открыть для редактирования файл:

     ```
     sudo nano /etc/fstab
     ```

   - добавить в него строку (указывается IP-адрес DB Server):

     ```
     192.168.14.162:/home/kmiadmin/backup_files /var/backups/out nfs defaults 0 0
     ```

4. Смонтировать указанные в файле /etc/fstab каталоги командой:

```
sudo mount -a
```

5. Проверить монтирование:

```
df -h
```

Перейти к Содержанию...

8.4. Установка и настройка компонентов KGS на Processing Server

8.4.1. Установка HASP

⚠ Процедура выполняется только на Processing Server.

USB-HASP устанавливается в Processing Server, используется в лестнице ключей.

HASP-ключ используется при шифровании лестницей ключей, экспортированной на сервер из системы KGS. Тем не менее, при описании процедуры установки библиотек KGS (см. Установка библиотек KGS) можно задать параметр **no_hasp**, позволяющий не пользоваться HASP.

Во всех остальных случаях наличие HASP-USB и установленных HASP drivers обязательно для установки и корректной работы библиотек KGS и, как следствие, всей системы.

Последовательность действий:

1. Убедитесь, что на сервере установлен пакет usbutils (данный пакет должен быть установлен по умолчанию). Если пакет отсутствует, его необходимо установить:

```
sudo apt-get install usbutils
```

2. Поместить прилагаемый HASP-ключ в USB-порт сервера.
3. Скопировать пакеты с драйверами HASP-ключа и необходимыми библиотеками на сервер, если это не было сделано ранее:
    a. ***aksusbd_8.43-1_amd64.deb***

Драйверы отсутствуют в репозитории производителя Системы. Предполагается, что их поиск остается на совести администратора, устанавливающего систему.

Подразумевается, что если у администратора есть HASP ключ, то он знает, где взять для него HASP drivers.

ВАЖНО! Наличие драйверов HASP обязательно в случае KMI_HASP_VERSION <> 'no_hasp'. В противном случае невозможна установка и использование библиотеки KMI_FW_TDE (см. Установка библиотек KGS).

4. Выполнить установку пакета с драйверами:

```
sudo dpkg -i aksusbd_8.43-1_amd64.deb
```

> На момент установки пакетов HASP-ключ должен быть установлен в USB-порт сервера.

5. Проверить статус установленных драйверов:

```
sudo service aksusbd status
```

6. **Перезагрузить сервер** (обязательно), с тем чтобы для того чтобы HASP-USB ключ был корректно обнаружен системой.
7. Скопировать файл с vendor_code на сервер, переименовать этот файл в vendor_code (файл без расширения) и положить в папку *./etc* , лежащую в рабочей директории (папка, откуда запускается исполняемый файл данного продукта). Файл с vendor_code предоставляется по специальному запросу.

> Рабочие директории у разных продуктов могут отличаться. Уточняйте детали у разработчиков конкретного продукта.
>
> Во избежание путаницы путь к файлу vendor_code, которым будет оперировать система KGS, указывается в настройках kmi_cfg.xml (см. ниже).

> Файл vendor_code поставляется и закрепляется за разработчиком при первоначальной покупке HASP-ключей.
>
> HASP-ключи разных серий разработчика (vendor code) обладают различным криптоповедением, поэтому ключи от одной серии не подходят для работы с приложением, защищённым HASP-ключами другой серии разработчика.

8.4.2. Настройка ODBC драйверов на Processing Server

Для корректной работы БД на ЭВМ (**Processing Server**), с которой будет осуществляться подключение к БД, должны быть установлены и настроены драйверы ODBC.

> Процедура выполняется только на Processing Server.

Для корректной работы БД на Processing Server необходимо установить следующее ПО:

- unixodbc-driver;
- PostgreSQL – odbc-driver;

Драйверы и библиотека могут быть установлены в любой момент времени, но до начала эксплуатации KMI_FW_DAL.

Все операции выполнять под *sudo*.

Последовательность действий:

1. Убедиться, что установлен драйвер unixodbc (процедура была выполнена ранее, см. Установка дополнительных утилит).
2. Установить драйвер ODBC-PostgreSQL:

```
apt-get install odbc-postgresql
```

3. Настроить ODBC (http://www.asteriskdocs.org/en/3rd_Edition/asterisk-book-html-chunk/installing_configuring_odbc.html):
   - отредактировать файл */etc/odbcinst.ini* (содержит информацию о драйверах) – убедиться, что в файле содержится следующая информация, и файлы по указанным путям существуют:

```
[PostgreSQL ANSI]
Description=PostgreSQL ODBC driver (ANSI version)
Driver=psqlodbca.so
Setup=libodbcpsqlS.so
Debug=0
CommLog=1
UsageCount=1
[PostgreSQL Unicode]
Description=PostgreSQL ODBC driver (Unicode version)
Driver=psqlodbcw.so
Setup=libodbcpsqlS.so
Debug=0
CommLog=1
UsageCount=1
```

   - отредактировать файл */etc/odbc.ini* (содержит настройки драйверов) – внести в файл следующую информацию (указывается IP-адрес DB Server):

```
[KMI_DB]
Description = PostgreSQL ANSI
Driver = PostgreSQL ANSI
Trace = No
TraceFile =
Database = kmi
Servername = 192.168.14.162
Username = kmiadmin
Password = kmiadmin
Port = 5432
#Protocol = 6.4
ReadOnly = No
RowVersioning = No
ShowSystemTables = No
ShowOidColumn = No
FakeOidIndex = No
ConnSettings =
```

   - Проверить доступность БД по ODBC командой:

```
sudo isql -v KMI_DB
```

8.4.3. Установка ограничений

Необходима проверка ограничений на память (hard/soft), в том числе на размер памяти.

На Processing Server необходимо установить ограничения на область памяти, защищенную от кэширования. По умолчанию, этот объем памяти слишком мал и требует прав *sudo* для разграничения программой доступной памяти: лимит = 1024000 (1 Гб памяти).

Ограничения задаются в файле */etc/security/limits.conf* внесением следующих данных:

```
* hard memlock 1024000
* soft memlock 1024000
```

Ограничения нужно задавать заново при каждом перезапуске сервера.

8.4.4. Установка и настройка файлов KGS Framework (KMI_FW)

Установка компонентов KGS Framework, за исключением KMI_FW_BACKUP, осуществляется на Processing Server. Модуль KMI_FW_BACKUP устанавливается на DB Server.

Установка компонентов KGS Framework осуществляется с помощью DEB-пакетов ОС (Debian). Файлы на обоих серверах будут установлены в папку */opt/kmi/*.

DEB-пакеты KMI_FW и KMI_CONSOLE используют библиотеки python, поэтому Python3 должен быть установлен до установки этих компонентов.

Поскольку Python3 входит в эталонный образ Debian11 ("устанавливается из коробки"), то дополнительные действия не требуются.

DEB-пакеты нужно устанавливать только в указанной последовательности (для сохранения зависимостей между пакетами). Следующий DEB-пакет можно устанавливать, только если успешно установлен предыдущий. Если в процессе установки возникла ошибка, то пакет нужно удалить (команда sudo dpkg -r <название_DEB_пакета>), устранить причину ошибки и установить пакет заново.

Если по каким-либо причинам DEB-пакеты были установлены с ошибкой, то нужно удалить их в обратной последовательности (т.е. снизу вверх).

8.4.4.1. Установка компонентов на Processing Server

Версия библиотеки HASP указывается как значение переменной KMI_HASP_VERSION. Система KGS использует переменную KMI_HASP_VERSION для обработки того, какой вариант HASP должен быть установлен. Описание возможных значений KMI_HASP_VERSION приведено в отдельном документе (доступ **строго ограничен**).

Перед установкой библиотеки kmi_fw_tde необходимо знать версию внешней библиотеки HASP, которая будет использоваться. Если используется версия 'bb' (значение по умолчанию), то выполняется стандартная установка библиотеки, никаких дополнительных действий не требуется. Если используется значение, отличное от значения по умолчанию, то сначала требуется установить новое значение KMI_HASP_VERSION и лишь затем устанавливать kmi_fw_tde.

Если реальная версия библиотеки HASP и значение переменной KMI_HASP_VERSION не совпадают, то после установки компонентов KGS любой продукт, который использует TDE (т.е. лестницу ключей, генерируемую KGS), не может быть запущен из-за ошибки инициализации.

Последовательность действий:

1. Определить, какая версия HASP-USB-ключа используется на сервере (в KGS подразумевается использование KMI_HASP_VERSION = kmi). Если фактически используется версия, отличная от значения по умолчанию (например, при разработке и тестировании HASP может не использоваться), то изменить значение переменной KMI_HASP_VERSION (в данном примере - kmi):

```
export KMI_HASP_VERSION=kmi
```

   В KGS также существует переменная TDE_NO_METRICS.

   Объявление переменной TDE_NO_METRICS (*export TDE_NO_METRICS=YES*) означает установку KMI_FW_TDE без сбора метрик, в остальных случаях (переменная имеет другое значение или не объявлена) - установку со сбором метрик. Вариант установки влияет на содержимое Binding Key (ключ в лестнице ключей).

   В системе KGS используется установка KMI_FW_TDE **только со сбором метрик**, поэтому никаких дополнительных действий не требуется (нельзя объявлять TDE_NO_METRICS).

2. Скопировать из репозитория на Processing Server следующие DEB-пакеты (путь к папке значения не имеет), входящие в состав релиза:
   - kmi_fw_api-**X.X.X**-linux-x86_64.deb
   - kmi_fw_dal-**X.X.X**-linux-x86_64.deb
   - kmi_fw_hwrk-**X.X.X**-linux-x86_64.deb
   - kmi_fw_dbmk-**X.X.X**-linux-x86_64.deb
   - kmi_fw_tde-**X.X.X**-linux-x86_64.deb
   - kmi_fw_transfer-**X.X.X**-linux-x86_64.deb
3. Перейти в папку, содержащую DEB-пакеты (в этом случае не нужно прописывать полный путь к пакетам, см. ниже).
4. Выполнить команды, заменив **X.X.X** на релизные версии компонентов (**выполнять только в указанной последовательности**):

```
sudo -E dpkg -i kmi_fw_tde-X.X.X-linux-x86_64.deb
sudo dpkg -i kmi_fw_dal-X.X.X-linux-x86_64.deb
sudo dpkg -i kmi_fw_api-X.X.X-linux-x86_64.deb
sudo dpkg -i kmi_fw_transfer-X.X.X-linux-x86_64.deb
sudo dpkg -i kmi_fw_hwrk-X.X.X-linux-x86_64.deb
```

> ⚠ При установке компонента kmi_fw_tde **обязательно** указать параметр *-E*.

> ⚠ Последним пакетом (после kmi_fw_hwrk) может быть установлен kmi_console-**X.X.X**-linux-x86_64.deb
>
> Установка и настройка KMI_CONSOLE описана ниже.

5. Дождаться окончания выполнения операции. Загрузка каждого компонента должна составлять 100%.
6. **Обязательно** скопировать файл с vendor_code с HASP-USB на Processing Server, в папку */opt/kmi/etc* и (в папке) сменить имя файла на **vendor_code** (т.е. на Processing Server должен появиться файл */opt/kmi/etc/vendor_code*).

Перейти к Содержанию...

8.4.4.2. Настройка конфигурационного файла KGS

Аналогично описанному выше.

8.4.4.3. Настройка PATH

Аналогично описанному выше.

8.4.5. Установка KMI_FW_DBMK

> ⚠ У компонента kmi_fw_dbmk нет зависимостей от других компонентов KGS Framework, поэтому kmi_fw_dbmk по факту может быть установлен и использоваться на любом сервере, если выполнены Системные требования (описаны в документе "DBMKGenerator. Руководство пользователя" (доступ предоставляется по запросу)) и на вход был подан HWRK-ключ, сгенерированный на Processing Server.
>
> Предполагается, что установка kmi_fw_dbmk осуществляется на отдельный сервер. Тем не менее, все необходимые требования для работы kmi_fw_dbmk уже выполнены на Processing Server.

Установка KMI_FW_DBMK выполняется аналогично другим компонентам KMI_FW:

1. Установить библиотеки, необходимые для работы KMI_FW_DBMK (см. документ "DBMKGenerator. Руководство пользователя" (доступ предоставляется по запросу), раздел "Системные требования"), на сервер, где будет использоваться KMI_FW_DBMK.

   > ⚠ Необходимые пакеты и библиотеки устанавливаются при развертывании KGS, в рамках установки других библиотек и их зависимостей. Т.е. если установка осуществляется на Processing Server, то установка дополнительных пакетов уже выполнена, дополнительных действий не требуется.

2. Скопировать из репозитория на сервер DEB-пакет KMI_FW_DBMK (путь к папке значения не имеет), входящий в состав релиза:
   a. kmi_fw_dbmk-**X.X.X**-linux-x86_64.deb
3. Перейти в папку, содержащую deb-пакет (в этом случае не нужно прописывать полный путь к пакетам, см. ниже).

4. Выполнить команду установки, заменив **X.X.X** на релизную версию компонента:

```
sudo dpkg -i kmi_fw_dbmk-X.X.X-linux-x86_64.deb
```

5. Дождаться окончания выполнения операции. Загрузка компонента должна составлять 100%.
6. KMI_FW_DBMK запускается автоматически при установке компонента с помощью DEB-пакета.

8.4.6. Установка и настройка KGS Console (KMI_CONSOLE)

⚠ KMI_CONSOLE устанавливается на Processing Server.

Требования:

- Установленный Python3.
- Наличие пользователя с *systemName*, идентичным тому, что занесено в БД, в таблицу **kmi_user**.
- У данного пользователя есть права доступа хотя бы на Management workflow.

8.4.6.1. Установка файлов KGS Console

Последовательность действий:

1. Скопировать из репозитория на Processing Server следующие DEB-пакеты (путь к папке значения не имеет):
    - kmi_console-**X.X.X**-linux-x86_64.deb
2. Перейти в папку, содержащую DEB-пакеты (в этом случае не нужно прописывать полный путь к пакетам, см. ниже).
3. Установить пакет с KMI_CONSOLE, заменив **X.X.X** на релизную версию компонента:

```
sudo dpkg -i kmi_console-X.X.X-linux-x86_64.deb
```

4. Дождаться окончания выполнения операции. Загрузка каждого компонента должна составлять 100%.

8.4.7. Настройка keyring

⚠ Процедура выполняется на одном сервере с компонентом KMI_FW_DAL, т.е. на Processing Server.

Для дополнительной очистки keyring при перезагрузке ото ВСЕХ ключей (использованных в прошлом или не удалившихся из-за каких-то возможных ошибок) можно монтировать keyring с PGP-ключами в RAM, добавив в */etc/fstab* строку:

```
tmpfs /root/.gnupg tmpfs defaults 0 0
```

Перейти к Содержанию...

8.5. Проверка автоматического запуска компонентов KGS Framework (KMI_FW)

8.5.1. Проверка автоматического запуска KMI_FW_DAL, KMI_FW_TRANSFER

⚠ Автоматический запуск настраивается автоматически при установке компонентов с помощью DEB-пакетов (см. "Установка и настройка файлов KGS Framework (KMI_FW)").

Чтобы проверить автоматический запуск сервиса kmi_fw_dal, выполните команду:

```
systemctl is-enabled kmi_fw_dal
```

Чтобы проверить автоматический запуск сервиса kmi_fw_transfer, выполните команду:

```
systemctl is-enabled kmi_fw_transfer
```

В обоих случаях ответ должен быть следующим:

```
enabled
```

Перейти к Содержанию...

8.6. Генерация ключей HWRK и DBMK

⚠ Процедура выполняется на Processing Server, перед запуском системы.

Для использования KMI_FW_DAL необходимо последовательно сгенерировать ключи HWRK и DBMK. Ключи являются первыми в цепочке ключей, которыми шифруется секретная часть БД KGS. При отсутствии ключей система работать не будет.

Последовательность действий:

1. Перейти в папку, содержащую kmi_fw_hwrk (*/opt/kmi/bin*).
2. Запустить HWRKGenerator (kmi_fw_hwrk) на Processing Server:
    a. Если команда выполняется на Processing Server напрямую (не удаленно):
    ```
    sudo ./kmi_fw_hwrk
    ```
    b. Если запуск осуществляется в терминале, который подключен к серверу по ssh:
        i. Установить tmux:
        ```
        sudo apt-get install tmux
        ```

ii. Выполнить переход в локальный терминал с помощью tmux:

```
sudo tmux
```

iii. Запустить HWRKGenerator (kmi_fw_hwrk) через этот терминал:

```
sudo ./kmi_fw_hwrk
```

> Если выполнить команду *sudo ./kmi_fw_hwrk* в терминале, который подключен к серверу по ssh, без tmux, то она завершится ошибкой: *[BindingKeyException]: Unable to login to HASP. Error code: 27*
>
> Необходимые ключи при этом не сгенерируются.
>
> Одним из способов обхода этой проблемы является установка tmux (sudo apt-get install tmux), переход в локальный терминал с помощью tmux (sudo tmux) и запуск kmi_fw_hwrk через этот терминал (sudo ./kmi_fw_hwrk).

3. При выполнении необходимых условий (установка lshw и HASP на Processing Server, наличие устройства HASP-USB в физическом сервере) утилита kmi_fw_hwrk генерирует файлы с открытым (по умолчанию **kmi_file12.dat**) и секретным (по умолчанию **kmi_file11.dat**) ключами. Оба ключа хранятся в той же папке, что и утилита.
4. Переслать открытый (public) HWRK-ключ (по умолчанию **kmi_file12.dat**) одному из уполномоченных хранителей, который с помощью ключа и утилиты KMI_FW_DBMK осуществляет генерацию и экспорт DBMK-ключей. Подробное описание приведено в документе "DBMKGenerator. Руководство пользователя" (доступ предоставляется по запросу).
   В общем случае необходимо:
   a. Перейти в папку, содержащую kmi_fw_dbmk (*/opt/kmi/bin*).
   b. Получить public HWRK-ключ (kmi_file12.dat), положить его в ту же папку.
   c. Выполнить команду вида:

   ```
   ./kmi_fw_dbmk –-mode 1 --hwrk-key kmi_file12.dat
   ```

   d. Дождаться успешного окончания операции.
5. Уполномоченный хранитель отправляет администратору KGS секретный (по умолчанию **kmi_file21.dat**) и открытый (по умолчанию **kmi_file22.dat**) DBMK-ключи, зашифрованные ключом HWRK.
6. Переместить ключи DBMK (по умолчанию, **kmi_file21.dat** и **kmi_file22.dat**) на Processing Server, в каталог */opt/kmi*.

> Если ключи были сгенерированы и зашифрованы в соответствии с приведенным алгоритмом, то дальнейшие действия не требуются.

Перейти к Содержанию...

8.7. Создание пользователей KGS Console

> Процедура выполняется на Processing Server.

Приведенные ниже процедуры создания пользователей и назначения паролей добавляются после установки DEB-пакетов KGS на сервер.

Последовательность действий:

1. Создать пользователя, выполнив команду:

```
sudo /opt/kmi/bin/create_kmi_user <user_system_name>
```

   Для нового пользователя на Processing Server будут автоматически созданы нужные каталоги (в том числе */in* и */out*), а также настроен автоматический запуск KMI_CONSOLE.
2. Назначить пользователю пароль, выполнив команду:

```
sudo passwd <user_system_name>
```

3. Ввести пароль пользователя и нажать Enter.
4. Подтвердить пароль и нажать Enter.

   Для нового созданного пользователя автоматически будут доступны следующие workflows:

   a. Работа с отчетами
      i. Создание отчета о состоянии
      ii. Информация о производстве STB
      iii. Получение списка заблокированных устройств для типа партии
      iv. Экспорт текущих статусов устройств для типа партии
      v. Экспорт истории статусов для выбранных устройств
      vi. Создание отчета о состоянии в2
   b. PGP Ключи Пользователя
      i. Получение списка пользовательских ключей
      ii. Импорт открытых PGP ключей
      iii. Создание пары PGP ключей
      iv. Экспорт открытых PGP ключей
      v. Удаление пользовательского ключа
   c. Ключи для тестовых устройств
      i. Экспорт тестовых корневых ключей
      ii. Экспорт тестовых ключей прошивки

Перейти к Содержанию...

9. Окончательная настройка и запуск развернутой системы KGS

9.1. Запуск служб KGS Framework (KMI_FW)

⚠ Процедура выполняется на Processing Server и DB Server.

ПРОЦЕДУРА ДОЛЖНА БЫТЬ ВЫПОЛНЕНА ПЕРЕД СДАЧЕЙ СИСТЕМЫ В ЭКСПЛУАТАЦИЮ, КОГДА УСТАНОВЛЕНЫ И НАСТРОЕНЫ ВСЕ КОМПОНЕНТЫ.

Запуск служб KGS осуществляется перед пробным запуском системы, с помощью команды вида:

```
sudo service <_> start
```

Вместо <*название_службы*> подставляются:

- *kmi_fw_dal* - на Processing Server.
- *kmi_fw_transfer* - на Processing Server.
- *kmi_fw_backup* - на DB Server.

Примечание. Вообще название службы совпадает с названием устанавливаемого deb-пакета (до версии компонентов).

В случае успеха в конце выполнения команды должно появиться **OK**.

Перейти к Содержанию...

9.2. Пробный запуск

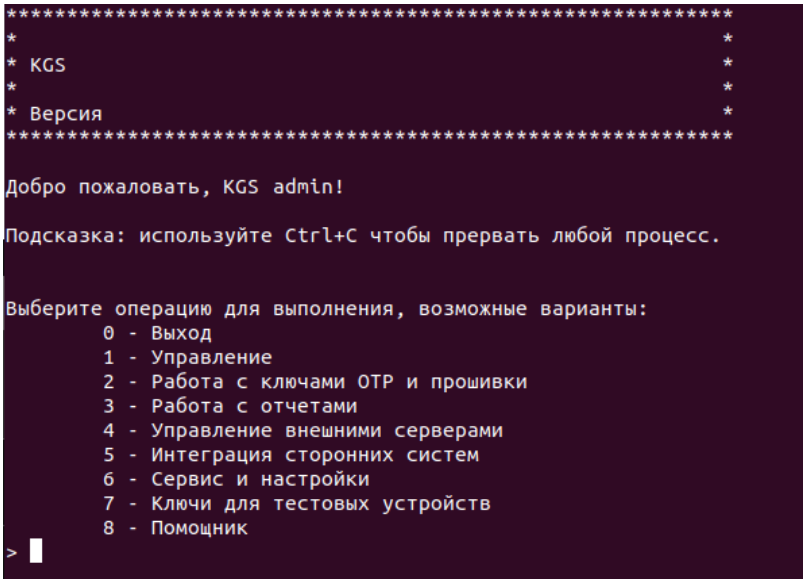
⚠ Пробный запуск осуществляется после установки и настройки всех компонентов KGS.

Управление системой KGS осуществляется с консоли (KVM), расположенной в закрытой комнате с ограниченным доступом.

Последовательность действий:

1. Открыть консоль (KVM).
2. Пройти авторизацию, введя имя пользователя и пароль, выданные администратором системы. После запуска KMI_CONSOLE откроется главное меню программы (см. ниже).

В случае если главное окно программы не появилось, необходимо проверить права доступа для пользователя.



3. Выполнить несколько операций, на которые у пользователя есть права доступа и не связанных с экспортом данных (например, добавить данные с помощью Management, Generate status report). Следует учесть, что многие операции требуют наличия public PGP-ключа в папке пользователя и (или) набор взаимосвязанных данных – при отсутствии этих данных выполнение некоторых операций приведет к ошибке. Подробное описание приведено в документе "Руководство пользователя".
4. Повторить шаги 2 и 3 для других пользователей, созданных в процессе установки KGS.

9.3. Добавление AMLOGIC_PATCH_PTTP в базу с помощью консоли

Для исправления ошибки, связанной с неверной перезаписью байт в ключе SEED и, как следствие, неверными значениями root-ключей после снятия обфускации в Trusted Application, в базу данных должен быть добавлен AMLOGIC_PATCH_PTTP. Добавление выполняется с помощью KMI_CONSOLE.

Последовательность действий:

1. Запустить консоль.
2. Перейти в **Сервис и настройки -> Конфигурация -> Добавление параметра**.
3. Выбрать *Тип ресурса* = Component.
4. Выбрать Id = 100 (KMI_External).
5. Выбрать *Тип параметра* = number.
6. Ввести имя AMLOGIC_PATCH_PTTP, нажать Enter.
7. Ввести значение = 129.
8. Нажать Enter, чтобы вернуться в главное меню.
9. Выйти из консоли.

Перейти к Содержанию...

9.4. Рекомендации по начальной настройке в KMI_CONSOLE

Для эксплуатации системы, установленной "с нуля", рекомендуется выполнить следующие минимальные действия в KMI_CONSOLE:

1. Создать пользователей KMI_CONSOLE, выдать им права доступа на workflows (**Сервис и настройки** -> **Пользователи и разрешения**).
2. Сгенерировать либо импортировать PGP ключи для этих пользователей (**Сервис и настройки** -> **PGP Ключи Пользователя**).
3. Задать основные сущности, с которыми будут оперировать пользователи (см. "Руководство пользователя").

Для получения более подробной информации по работе с KMI_CONSOLE рекомендуется обратиться к документам "Руководство пользователя" и "Руководство администратора" (доступ предоставляется по запросу).

9.5. Многопользовательский режим KGS

В KMI_CONSOLE используется многопользовательский режим: в консоли могут работать одновременно несколько пользователей (пользователи подключаются к консоли удаленно, например, по SSH). Если один пользователь работает в workflow, то указанное workflow будет заблокировано для других пользователей. По окончании работы пользователя с workflow блокировка снимается. Подробности, касающиеся блокировки workflows и действий пользователей, описаны в документе "Руководство пользователя".

В многопользовательском режиме служба KMI_FW_DAL работает с несколькими потоками. При установке системы KGS **количество потоков по умолчанию = 3 шт**. Это означает, что в KMI_CONSOLE могут **одновременно** выполняться три команды (пользователь вызывает команды, например, при отображении списка либо генерации данных, при этом ввод значений при выполнении workflow команду DAL не вызывает). При этом количество пользователей, которые **одновременно** подключены к KMI_CONSOLE, может быть любым (намного больше).

Таким образом, может возникнуть ситуация, когда одновременно будут заняты три потока (т.е. одновременно будут выполняться три команды DAL), при этом есть другие пользователи, подключенные к KMI_CONSOLE и желающие выполнить какую-либо команду DAL. В результате у пользователей, которые не успели "занять" потоки для работы с DAL, KMI_CONSOLE будет "висеть" до освобождения потока.

Если потребуется изменить количество потоков для одновременной работы с DAL, то необходимо выполнить действия, описанные в документе "Руководство администратора" (доступ предоставляется по запросу), в разделе "Запуск KMI_FW_DAL с альтернативными параметрами".

Перейти к Содержанию...