1 Общие сведения о программе

1.1 Назначение и функции ПУМ

ПУМ является распределённым программно-техническим комплексом, который включает: серверы, сетевые коммутаторы, устройства распределения электропитания, а также программные системы для управления всем программно-техническим комплексом.

1.1.1. Структура программы

ПУМ состоит из следующих подсистем:

а) Подсистема управления:

1) Модуль управления ПДУ и КРК
2) Модуль управления трафиком, топологией и технологическими параметрами .

б) Подсистема мониторинга:

1) Модуль мониторинга ПДУ и КРК
2) Модуль мониторинга оборудования в)

Подсистема технического учёта:

1) Модуль учета оборудования;
2) Модуль учета логических ресурсов;

г) Подсистема взаимодействия с внешними системами:

1) Модуль экспорта данных во внешние системы;
2) Модуль обработки запросов API в реальном времени;

д) Подсистема контроля доступа;

е) ПО АРМ администратора

ж) ЛСУ и агенты ЛСУ

Каждый из модулей ПУМ состоит из:

– агента (агентов), устанавливаемого в управляемый компонент (вПДУ функции агентов ПУМ могут выполнять модули ПДУ);
– агентов рабочих мест администраторов.

1.2 Назначение и функции подсистем

1.2.1 Подсистема управления

1.2.1.1 Модуль управления ПДУ и КРК

Модуль управления ПДУ и КРК обеспечивает:

а) Управление процессом работы на подсистем ПКРК, ПДУ,ключами оконечных клиентских узлов (с учетом действующей в РФ нормативной базы), включая:

1) Генерацию ключей;
2) Скорость генерации КЗК для отдельных клиентов (потребителей) квантовых каналов;
3) Хранение ключей;
4) Управление циклом жизни ключей;
5) Возможность удаленной перезагрузки оборудования ПКРК и ПДУ.

б) Управление безопасной передачей КЗК через цепочку доверенных опорных узлов средствами ПДУ;

в) Управление резервированием ПКРК;

г) Электронное журналирование действий персонала ПУМ;

д) Электронное журналирование событий ПУМ;

е) Синхронизацию времени на всех компонентах ПУМ. Аппаратное время ПУМ должно получаться из источника вне ПУМ, но не из Интернет.

Установка модуля управления ПДУ и КРК производится в ЦУМ.

1.2.1.2 Модуль управления трафиком, топологией и технологическими параметрами

Модуль управления трафиком, топологией и технологическими параметрами должен обеспечивать:

а) Формирование топологии сети в графическом виде, отображение всех сетевых узлов и связей между ними;

б) управление ресурсами ККП и :

1) автоматический расчет емкости сети
2) определение технической возможности включения новых потребителей (клиентов) ККП без ухудшения параметров и уровня обслуживания для подключенных ранее потребителей (клиентов)
3) приоритезация пользователей и трафика. Устанавливаются три уровня приоритета: низкий, обычный, высокий. Трафик ключевой информации должен всегда иметь высокий приоритет. Принципы управления трафиком должны строиться на использовании резервных каналов передачи данных и управлением их пропускной способностью.

Установка модуля управления трафиком, топологией и технологическими параметрами производится в ЦУМ.

в) Устанавливаются три уровня приоритета: низкий, обычный, высокий.

г) Трафик ключевых данных всегда имеет высокий приоритет.

д) Принципы управления трафиком строятся на использовании резервных каналов передачи данных (при их наличии) и управлением их пропускной способностью.

Модуль управления трафиком, топологией и технологическими параметрами ККП состоит из двух частей: для выполнения на ЦУМ и для выполнения на ЛСУ.

а) Функционал модуля в ЦУМ обеспечивает:

1) Инициализацию модуля;
2) Взаимодействие модуля в ЦУМ с подмножеством ЛСУ (либо сетевых ПДУ в случае выбора варианта реализации);

3) Взаимодействие с ОУ;
4) Взаимодействие с пользователем;
5) Взаимодействие с модулем мониторинга;

б) Функционал модуля в ЛСУ обеспечивает:
1) Инициализация модуля
2) Взаимодействие с ЦУМ;
3) Взаимодействие с коммутаторами транспортной сети передачи данных

1.2.2 Подсистема мониторинга

1.2.2.1 Модуль мониторинга ПДУ и КРК

Модуль мониторинга ПДУ и КРК должен осуществлять автоматическую диагностику, мониторинг и формирование предупреждений в реальном времени по следующим событиям в ККП:

- превышение порога QBER с подозрением на попытку несанкционированного доступа к квантовой сети;
- нарушение работоспособности оборудования ПКРК и ПДУ.

Модуль мониторинга ПДУ и КРК взаимодействует с внешней системой (ВС), которая предназначена для обеспечения единого непрерывного жизненного цикла управления технологическими сетями передачи данных.

Модуль мониторинга ПДУ и КРК обеспечивает передачу следующих данных в ВС:

- передачу событий на основе данных мониторинга ПДУ и КРК в привязке к конфигурационным единицам (событие – это любой выход значений параметров любого компонента за допустимые пределы и /или расхождение в составе оборудования в ПУМ и ВС).

Модуль мониторинга ПДУ и КРК обеспечивает прием следующих данных из ВС:

– прием данных о результатах обработки событий в привязке к инцидентам и работам для информирования администратора ПУМ. ВС направляет в модуль мониторинга данные о всех работах и/или инцидентах, связанных с событиями, которые модуль мониторинга оборудования ранее отправил в ВС.

Установка модуля мониторинга ПДУ и КРК производится в ЦУМ.

1.2.2.2 Модуль мониторинга оборудования ККП

Модуль мониторинга оборудования ККП, обеспечивает мониторинг, сбор, хранение и обработку следующей информации ККП:

– температуры внутри аппаратных модулей узлов ККП;
– результаты контроля работоспособности вентиляторов и источников питания;
– параметры утилизации ресурсов;
– результатов тестирования программных компонентов ПКРК,
– иной информации, определенной в эксплуатационной документации на КРК (покупное изделие), используемых в ПКРК ККП в соответствии с руководством по эксплуатации производителя (с учётом требований Регулятора).

Модуль мониторинга оборудования обеспечивает визуализацию (вывод через единый графический интерфейс администратора сети КРК) следующих параметров ККП:

– скорость генерации квантовых ключей между каждой парой ПОУ и ОУ и сравнение с заданным порогом SLA для каждого из клиентских сервисов;
– QBER между каждой парой ПОУ и ОУ;
– частота обновления КЗК между парой узлов;
– иную информацию.

Модуль мониторинга оборудования взаимодействует с внешней системой ВС, которая предназначена для обеспечения единого непрерывного жизненного цикла управления технологическими сетями передачи данных.

Модуль мониторинга оборудования обеспечивает передачу следующих данных в ВС:

- передачу событий на основе данных мониторинга оборудования в привязке к конфигурационным единицам (событие – это любой выход значений параметров любого компонента за допустимые пределы и /или расхождение в составе оборудования вПУМ и ВС).

Модуль мониторинга оборудования обеспечивает прием следующих данных из ВС:

- прием данных о результатах обработки событий в привязке к инцидентам и работам для информирования администратора ПУМ. ВС направляет в модуль мониторинга данные о всех работах и/или инцидентах, связанных с событиями, которые модуль мониторинга оборудования ранее отправил в ВС.

Установка модуля мониторинга оборудования производится в ЦУМ.

1.2.3 Подсистема технического учёта

1.2.3.1 Модуль учета оборудования

Модуль учета оборудования осуществляет:

- автоматическую загрузку состава оборудования ККП с возможностью заполнения места расположения, координат, номеров коммерческих заказов и прочей дополнительной информации.

– создания объектов, загруженных из внешних систем с возможностью внесения описания.

Модуль учета оборудования взаимодействует с внешней системой ВС, которая предназначена для обеспечения единого непрерывного жизненного цикла управления технологическими сетями передачи данных. Компоненты, оборудование, соединения компонентов описываются в ВС. Все описания из ВС реплицируются в модуле учета оборудования.

Модуль учета оборудования обеспечивает передачу следующих данных в ВС:

– регулярную передачу инвентаризационных данных, полученных с оборудования.

Модуль учета оборудования должен обеспечивать прием следующих данных в ВС:

– прием данных о типах, моделях и экземплярах конфигурационных единиц, соединениях между конфигурационными единицами.

Установка модуля учета оборудования производится в ЦУМ.

1.2.3.2 Модуль учета логических ресурсов

Модуль учета логических ресурсов осуществляет:

– автоматическую загрузку состава логических ресурсов ККП с возможностью заполнения дополнительной информации.
– создания объектов, загруженных из внешних систем с возможностью внесения описания.

Модуль учета логических ресурсов взаимодействовует с внешней системой ВС, которая предназначена для обеспечения единого непрерывного жизненного цикла управления технологическими сетями передачи данных.

Модуль учета логических ресурсов обеспечивает передачу следующих данных в ВС:

– регулярную передачу данных о логических ресурсах, полученных с оборудования.

Установка модуля учета логических ресурсов производится в ЦУМ.

1.2.4 Подсистема взаимодействия ПУМ с внешними системами

1.2.4.1 Модуль экспорта данных во внешние системы

Модуль экспорта данных во внешние по отношению к ПУМ системы осуществляет средствами сетевого доступа:

– Периодический экспорт определённой части данных во внешнюю систему (периодичность задается Администратором ПУМ) состава оборудования и логических ресурсов, аварийных сообщений, журналов логирования и пр.;

– Взаимодействие с внешними системами должно быть обеспечено в необходимом объеме функций и гарантировать отсутствие влияния на безопасность ПУМ.

1.2.4.2 Модуль обработки запросов API в реальном времени

Модуль обработки API в реальном времени работает по следующей схеме:

– Протокол информационного обмена с внешними системами должен быть реализован в форме обмена сообщениями в формате YAML;

– Прием запроса в формате REST (HTTP GET/POST, JSON) от внешней системы. Протокол информационного обмена определяется совместно со Стратегическим индустриальным партнером на этапе Технического проекта;

– Прием запроса от внешней системы с использованием системы очередей сообщений брокера AMQP RabbitMQ;

– Обработка запроса с использованием диагностической информации с соответствующих подсистем и компонентов ;

– Возврат полученных в ходе обработке данных инициатору запроса.

Запросы формируются в формате JSON (YAML). Содержательные поля запроса:

Поле запроса	Значение поля запроса
TIMESTAMP	Дата формирования запроса в секундах, начиная с 1.01.1970 (UNIX timestamp)
SYSNAME	Имя устройства PDU согласно установленной нотации
PARMNAME_1	Название параметра №1
…	…
PARAMNAME_n	Название параметра №n

Формат ответа:

Поле ответа	Значение поля ответа
TIMESTAMP	Дата формирования ответа в секундах, начиная с 1.01.1970 (UNIX timestamp)
SYSNAME	Имя устройства PDU согласно установленной нотации
PARMNAME_1	Значение параметра №1
…	…
PARAMNAME_n	Значение параметра №n

Список параметров PARAMNAME:

– QBER между каждой парой модулей КРК;

– Скорость генерации квантовых ключей между каждой парой модулей КРК;

– Значение параметров оптических модулей: температура оптического модуля, оптическая мощность на прием и передачу;

– Прочие оптические параметры, доступные по DDM оптического модуля;

– Температурные характеристики детектора одиночных фотонов (при наличии технической возможности);

– Прочие параметры.

Для случая получения запроса от внешней системы через систему очередей брокера AMQP RabbitMQ на стороне внешней системы должны быть выполнены определенные условия.

Посылка запроса и прием результата выполнения запроса могут быть разнесены по времени. То есть результат выполнения запроса может быть запрошен не сразу, а через какой-то промежуток времени. Результат выполнения запроса будет храниться в очереди ответа, пока не будет принят.

Для работы с брокером RabbitMQ использовать пакет подпрограмм pika.

Можно использовать Python 3 или другие языки программирования.

Общий формат запроса (YAML), посылаемый в очередь брокера RabbitMQ должен быть такой:

```
description: “…..”
rabbitmq-params:
  host: "…"
  login: "..."
  password: "..."
  exchange: ""
  request-queue: "..."
  result-queue: "..."
```

timeout: nn

body:

{command: …}

expected-result:

{ok: true/false, status: …}

Параметр “description” описывает назначение запроса.

Параметр “rabbitmq-params” задает параметры брокера AMQP RabbitMQ.

– Host – имя(адрес) брокера RabbitMQ;

– login – имя пользователя в брокере RabbitMQ;

– password – пароль пользователя;

– exchange – имя exchange, по умолчанию “”;

– request-queue – имя очереди, куда посылается запрос;

– result-queue – имя очереди, куда посылается результат выполнения запроса, для каждого запроса проще создавать отдельную очередь ответа. Для генерации уникального имени очереди ответа можно использовать команду uuidgen. Если для всех запросов использовать одну очередь ответа, то может возникнуть ситуация, когда ответ на запрос, который послан раньше, может быть помещен в очередь ответа позже;

– timeout – время ожидания (в секундах) получения результата из очереди result-queue.

Параметр “body” задает команду запроса вместе с параметрами команды

Параметр “expected-result” задает ожидаемый результат выполнения запроса

Посылка запроса в очередь брокера RabbitMQ

Чтобы послать запрос request в очередь request-queue брокера RabbitMQ нужно выполнить примерно такой код;

```
Credentials = pika.PlainCredentials(login, password)
Parameters = pika.ConnectionParameters(host=host, credentials=Credentials)
connection = pika.BlockingConnection(Parameters)
channel = connection.channel()
channel.queue_declare(queue=result-queue)
channel.queue_bind(exchange=exchange, queue=result-queue)
correlation_id = str(uuid.uuid4())
channel.basic_publish(exchange=exchange,
                      routing_key=request-queue,
                      properties=pika.BasicProperties(
                         reply_to=result-queue,
                         correlation_id=correlation_id
                      ),
                      body=yaml.dump(request, default_flow_style=True))
```

Получение ответа из очереди брокера RabbitMQ

Чтобы получить ответ на запрос request из очереди ответа result-queue брокера RabbitMQ нужно выполнить примерно такой код:

```
def callbackreply(ch=None, method=None, properties=None, body=None):
    if ch is None or method is None or properties is None or body is None:
        print ("Timeout {} expired. Exiting".format(timeout))
        deadline = True
        return
    if correlation_id != properties.correlation_id:
```

```
            print ("correlation_id mismatch: {} not equal
{}".format(properties.correlation_id,    correlation_id))
            reply = None
            print ("body: {}".format(body))
            return
        deadline = False
        reply = body
        ch.stop_consuming()
        return
channel.basic_consume(result-queue, callbackreply, auto_ack=True)
connection.call_later(timeout, callbackreply)
while reply is None and deadline is False:
        connection.process_data_events()
channel.stop_consuming()
channel.queue_delete(queue=result-queue)
if deadline is False:
        reply = yaml.safe_load(reply)
else:
      reply = None
channel.close()
connection.close()
print ("reply: {}".format(reply))
```

1.2.5 Подсистема контроля доступа

Подсистема контроля доступа ПУМ обеспечивает:

- аутентификацию, авторизацию и аудит учетных записей.

- доступ пользователей в систему через графический интерфейс автоматизированного рабочего места;
- регистрацию пользователя (администратора) в системе с использованием подтверждения со стороны администратора и назначения группы, в которую включён пользователь. Данные пользователя автоматически вносятся в OpenLDAP;
- контроль учетных данных пользователей;
- внесение изменений в учётные данные пользователей через графический интерфейс;
- определение принадлежности пользователей к группам;
- запись событий входа и взаимодействия с системой в системные логи;
- блокировка и разблокировка пользователей;
- автоматизация создания учетных данных в системах OpenLDAP и Kerberos для дальнейшего использования устройствами системы;
- интеграция в модули управления и мониторинга квантовой коммуникационной платформы;
- единую точку входа по протоколу CAS 2.0, в которой используются данные из OpenLDAP.
- обеспечение отказоустойчивой работы за счёт репликации данных.

Установка подсистемы контроля доступа производится в ЦУМ.

1.2.6 Сервер и агенты ЛСУ

На Локальном сервере управления (далее по тексту ЛСУ) устанавливаются программные агенты, которые выполняют функции, соответствующие наименованиям агентов:

- Агент модуля учёта логических ресурсов основанный на заимствованном ПО fusion-inventory
- Агент подсистемы мониторинга, основанный на заимствованном ПО Zabbix-agent
- Агент подсистемы управления, включая модуль взаимодействия с СПДУ

Все агенты, выполняющиеся на ЛСУ, взаимодействуют с ЦУМ напрямую и не производят сохранение каких-либо данных непосредственно на ЛСУ.

Установка агентов ЛСУ производится на ЛСУ.

1.2.7 ПО АРМ администратора

Автоматизированное рабочее место администратора (АРМ администратора) реализует функции взаимодействия ККП с администраторами с помощью графического интерфейса. Графический интерфейс реализуется в браузере администратора и может быть выполнен на рабочем месте – ПК, имеющем доступ к сети управления. Серверная часть ПО АРМ администратора выполняется на ЦУМ и позволяет производить авторизацию пользователей, отображать окна графического интерфейса с учётом доступных ролей в системе за счёт использования подсистемы контроля доступа.

ПО АРМ администратора обеспечивает:

- генерацию окон графического интерфейса
- отображение окон в браузере администратора по запросу
- отправку действий пользователя на исполнение подсистеме управления ККП
- отображение результата выполнения действий подсистемой управления ККП в графическом виде
- интеграцию с подсистемой мониторинга и технического учёта для отображения их данных

Установка ПО АРМ администратора производится в ЦУМ.

1.3 Сведения о технических и программных средствах, обеспечивающих выполнение данной программы

Серверная часть ПО АРМ администратора выполняется на сервере ЦУМ в форме виртуальной машины (название ВМ). Клиентская часть ПО АРМ администратора выполняется на станции администратора ПУМ.

2 Настройка программы

2.1 Подсистема управления

2.1.1 Модуль управления ПДУ и КРК

Модуль управления ПДУ и КРК состоит из двух частей. Часть выполняемая на серверах ПУМ настраивается как компонент серверной части АРМ администратора. В частности задаются очереди сервера AMQP по которым происходит взаимодействие модуля с другими компонентами системы, а так же параметры доступа до сервера AMQP. Часть выполняемая на сервере ЛСУ получает свои настройки при загрузке системы на ЛСУ автоматически, и какой-либо дополнительной настройки не требует.

2.1.2 Модуль управления трафиком, топологией и технологическими параметрами

2.2 Подсистема мониторинга

2.2.1 Модуль мониторинга ПДУ и КРК

Настройка модуля мониторинга ПДУ и КРК выполняется непосредственно из веб-интерфейса после установки заимствованной программной системы Zabbix.

В панели Настройка -> Узлы сети добавляются и настраиваются все ПДУ и КРК, за которыми осуществляется мониторинг, к ним добавляются необходимые шаблоны мониторинга.

В панели Настройка -> Шаблоны создаются и редактируются шаблоны, которые содержат сведения о данных (Элементы данных), которые надо мониторировать, условия и пороги срабатывания аварийных событий (Триггеры), графики.

В панели Настройка -> Действия -> Действия триггеров настраиваются действия, выполняемые при создании аварийного события.

В панели Администрирование -> Способы оповещений настраиваются способы оповещений администраторов.

В панели Администрирование -> Аутентификация настраивается способ и параметры аутентификации через подсистему контроля доступа.

В панели Администрирование -> Группы пользователей настраиваются права на просмотр и редактирование сведений о мониторинге.

В панели Администрирование -> Пользователи настраивается принадлежность пользователя к определенным группам пользователей.

Настройка конфигурации сервера выполняется в файле /etc/zabbix/zabbix_server.conf:

```
DBHost= АДРЕС _БД_ZABBIX
DBName=zabbix
DBUser=zabbix
DBPassword=ПАРОЛЬ_БД_ZABBIX
```

Настройка конфигурации веб-интерфейса выполняется в файле /etc/zabbix/web/zabbix.conf.php:

```
<?php
// Zabbix GUI configuration file.
$DB['TYPE']                     = 'POSTGRESQL';
$DB['SERVER']                   = 'АДРЕС _БД_ZABBIX';
$DB['PORT']                     = '0';
$DB['DATABASE']                 = 'zabbix';
$DB['USER']                     = 'zabbix';
$DB['PASSWORD']                 = 'ПАРОЛЬ_БД_ZABBIX';
// Schema name. Used for PostgreSQL.
$DB['SCHEMA']                   = '';
// Used for TLS connection.
$DB['ENCRYPTION']               = false;
$DB['KEY_FILE']                 = '';
$DB['CERT_FILE']                = '';
$DB['CA_FILE']                  = '';
```

```
$DB['VERIFY_HOST']          = false;
$DB['CIPHER_LIST']          = '';
// Use IEEE754 compatible value range for 64-bit Numeric (float) history values.
// This option is enabled by default for new Zabbix installations.
// For upgraded installations, please read database upgrade notes before enabling this option.
$DB['DOUBLE_IEEE754']   = true;
$ZBX_SERVER                     = 'АДРЕС_ZABBIX';
$ZBX_SERVER_PORT            = '10051';
$ZBX_SERVER_NAME            = 'АРМ МОНИТОРИНГ ЦУМ-1';
$IMAGE_FORMAT_DEFAULT   = IMAGE_FORMAT_PNG;
```

Настройка агента выполняется в файле /etc/zabbix/zabbix_agentd.conf:

ServerActive=АДРЕС_ZABBIX

Hostname=ИМЯ_ХОСТА

2.2.2 Модуль мониторинга оборудования ККП

Настройка модуля мониторинга ПДУ и КРК выполняется непосредственно из веб-интерфейса после установки заимствованной программной системы Zabbix.

В панели Настройка -> Узлы сети добавляются и настраиваются все оборудование , за которыми осуществляется мониторинг. Задается IP адрес, тип сбора данных (от агента или по SNMP), добавляются необходимые шаблоны мониторинга.

В панели Настройка -> Шаблоны создаются и редактируются шаблоны, которые содержат сведения о данных (Элементы данных), которые надо мониторировать, условия и пороги срабатывания аварийных событий (Триггеры), графики.

В панели Настройка -> Действия -> Действия триггеров настраиваются действия, выполняемые при создании аварийного события.

В панели Администрирование -> Способы оповещений настраиваются способы оповещений администраторов.

В панели Администрирование -> Аутентификация настраивается способ и параметры аутентификации через подсистему контроля доступа.

В панели Администрирование -> Группы пользователей настраиваются права на просмотр и редактирование сведений о мониторинге.

В панели Администрирование -> Пользователи настраивается принадлежность пользователя к определенным группам пользователей.

Настройка конфигурации сервера выполняется в файле /etc/zabbix/zabbix_server.conf:

```
DBHost= АДРЕС _БД_ZABBIX
DBName=zabbix
DBUser=zabbix
DBPassword=ПАРОЛЬ_БД_ZABBIX
```

Настройка конфигурации веб-интерфейса выполняется в файле /etc/zabbix/web/zabbix.conf.php:

```
<?php
// Zabbix GUI configuration file.
$DB['TYPE']                     = 'POSTGRESQL';
$DB['SERVER']                   = 'АДРЕС _БД_ZABBIX';
$DB['PORT']                     = '0';
$DB['DATABASE']                 = 'zabbix';
$DB['USER']                     = 'zabbix';
$DB['PASSWORD']                 = 'ПАРОЛЬ_БД_ZABBIX';
// Schema name. Used for PostgreSQL.
$DB['SCHEMA']                   = '';
// Used for TLS connection.
```

```
$DB['ENCRYPTION']            = false;
$DB['KEY_FILE']              = '';
$DB['CERT_FILE']             = '';
$DB['CA_FILE']               = '';
$DB['VERIFY_HOST']           = false;
$DB['CIPHER_LIST']           = '';
// Use IEEE754 compatible value range for 64-bit Numeric (float) history values.
// This option is enabled by default for new Zabbix installations.
// For upgraded installations, please read database upgrade notes before enabling this option.
$DB['DOUBLE_IEEE754']   = true;
$ZBX_SERVER                  = 'АДРЕС_ZABBIX';
$ZBX_SERVER_PORT             = '10051';
$ZBX_SERVER_NAME             = 'АРМ МОНИТОРИНГ ЦУМ-1';
$IMAGE_FORMAT_DEFAULT   = IMAGE_FORMAT_PNG;
```

Настройка агента выполняется в файле /etc/zabbix/zabbix_agentd.conf:

ServerActive=АДРЕС_ZABBIX

Hostname=ИМЯ_ХОСТА

2.3 Подсистема технического учёта

2.3.1 Модуль учета оборудования

Настройка модуля учета оборудования выполняется непосредственно из веб-интерфейса после установки заимствованной программной системы GLPI.

В панели Активы -> Компьютеры добавляются и настраиваются все не сетевое оборудование .

В панели Активы -> Сетевое устройство добавляются и настраиваются все сетевое оборудование .

В панели Настройка -> Общий -> Общие настройки настраивается URL приложения http://АДРЕС_АРМ/glpi.

В панели Настройка -> Общий -> API настраивается URL API http://АДРЕС_АРМ/glpi/apirest.php и клиент API для ПО АРМ.

В панели Настройка -> Аутентификация -> LDAP каталоги настраивается подключение к LDAP:

Наименование: LDAP STRELA

Сервер: ldap://АДРЕС_LDAP,ldap://АДРЕС_LDAP

Порт: 389

База поиска: dc=strela

Поле имени пользователя: uid

Для взаимодействия с подсистемой контроля доступа в панели Настройка -> Аутентификация -> Другие способы аутентификации настраивается CAS аутентификация:

Сервер CAS: АДРЕС_АРМ

CAS версия: Версия 2

Порт: 80

Корневая директория: cas

URL перенаправления при выходе: http://АДРЕС_АРМ/cas/logout?service=http://АДРЕС_АРМ/glpi/

В панели Администрирование -> Профили настраиваются права на просмотр и редактирование сведений об оборудовании.

В панели Администрирование -> Пользователи настраивается принадлежность пользователя к определенным профилям.

Настройка конфигурации подключения к БД выполняется в файле /etc/glpi/config_db.php:

```
<?php
class DB extends DBmysql {
```

```
    public $dbhost     = 'АДРЕС _БД_GLPI';
    public $dbuser     = 'glpi';
    public $dbpassword = 'ПАРОЛЬ_БД_GLPI';
    public $dbdefault  = 'glpi';
}
```

Настройка агента выполняется в файле /etc/fusioninventory/agent.cfg:

server=http://АДРЕС_GLPI/plugins/fusioninventory/

logger = file

logfile = /var/log/fusioninventory.log

2.3.2 Модуль учета логических ресурсов

Настройка модуля учета оборудования выполняется непосредственно из веб-интерфейса после установки заимствованной программной системы GLPI.

В панели Активы -> Компьютеры добавляются сведения о логических ресурсах всего не сетевого оборудования .

В панели Активы -> Сетевое устройство добавляются сведения о логических ресурсах всего сетевого оборудования .

В панели Настройка -> Общий -> Общие настройки настраивается URL приложения http://АДРЕС_АРМ/glpi.

В панели Настройка -> Общий -> API настраивается URL API http://АДРЕС_АРМ/glpi/apirest.php и клиент API для ПО АРМ.

В панели Настройка -> Аутентификация -> LDAP каталоги настраивается подключение к LDAP:

Наименование: LDAP STRELA

Сервер: ldap://АДРЕС_LDAP,ldap://АДРЕС_LDAP

Порт: 389

База поиска: dc=strela

Поле имени пользователя: uid

Для взаимодействия с подсистемой контроля доступа в панели Настройка -> Аутентификация -> Другие способы аутентификации настраивается CAS аутентификация:

Сервер CAS: АДРЕС_АРМ

CAS версия: Версия 2

Порт: 80

Корневая директория: cas

URL перенаправления при выходе: http://АДРЕС_АРМ/cas/logout?service=http://АДРЕС_АРМ/glpi/

В панели Администрирование -> Профили настраиваются права на просмотр и редактирование сведений об оборудовании.

В панели Администрирование -> Пользователи настраивается принадлежность пользователя к определенным профилям.

Настройка конфигурации подключения к БД выполняется в файле /etc/glpi/config_db.php:

```
<?php
class DB extends DBmysql {
  public $dbhost     = 'АДРЕС _БД_GLPI';
  public $dbuser     = 'glpi';
  public $dbpassword = 'ПАРОЛЬ_БД_GLPI';
  public $dbdefault  = 'glpi';
}
```

Настройка агента выполняется в файле /etc/fusioninventory/agent.cfg:

server=http://АДРЕС_GLPI/plugins/fusioninventory/

logger = file

logfile = /var/log/fusioninventory.log

2.4 Подсистема взаимодействия с внешними системами

2.4.1 Модуль экспорта данных во внешние системы

Скрипт glpi_get_inv_data.py получает техническую конфигурацию с названиями устройств, географических локаций, соединений в формате JSON с некоторыми комментариями. Для его работы необходимо указать действующий URL, токен API и токен пользователя для доступа к подсистеме технического учета (переменные glpi_url, glpi_app_token, glpi_auth_token).

В файле zbx_input_data.conf описаны сведения о данных мониторинга, которые необходимо получить внешней системой. Для добавления или правки необходимых данных мониторинга править этот файл. Скрипт zbx_get_mon_data.py получает описанные данные мониторинга. Для его работы необходимо указать действующий URL, логин и пароль для доступа к подсистеме мониторинга (переменные zbx_url, zbx_user, zbx_pass).

2.4.2 Модуль обработки запросов API в реальном времени

В файле strela_docker_model/pum/example/products/responder.py описаны команды, которые могут быть выполнены. Для добавления команд править этот файл.

2.5 Подсистема контроля доступа

Настройка подсистемы контроля доступа выполняется в файлах конфигурирования сервиса httpd:

/etc/httpd/conf.d/auth_cas.conf:

LoadModule auth_cas_module modules/mod_auth_cas.so

CASCookiePath /var/cache/httpd/mod_auth_cas/

CASLoginURL https://ВНЕШНИЙ_АДРЕС/cas/login

CASValidateURL https://ВНЕШНИЙ_АДРЕС/cas/serviceValidate

CASVersion 2

```
cat /etc/httpd/conf.d/host.conf
<Proxy balancer://zabbix>
    BalancerMember http://АДРЕС_ ZABBIX/zabbix
    AuthType CAS
    CASAuthNHeader REMOTE_USER
    CasScope /
    AuthName "Authentication required"
    require valid-user
</Proxy>
<VirtualHost *:80>
  RewriteEngine on
  RewriteRule ^/zabbix$ /zabbix/ [R]
  RewriteRule ^/glpi$ /glpi/ [R]
  ProxyPass        "/zabbix" balancer://zabbix
  ProxyPassReverse "/zabbix" "http://АДРЕС_ZABBIX/zabbix"
  ProxyPass        "/glpi" "http://АДРЕС_GLPI/glpi"
  ProxyPassReverse "/glpi" "http:// АДРЕС_GLPI/glpi"
  ProxyPass        "/ws" "ws://127.0.0.1:8000/ws"
  ProxyPassReverse "/ws" "ws://127.0.0.1:8000/ws"
  ProxyPass        "/" "http://127.0.0.1:8000/"
  ProxyPassReverse "/" "http://127.0.0.1:8000/"
```

```
  ProxyPreserveHost On

  LimitRequestBody 50000000

</VirtualHost>
```

2.6 ПО АРМ администратора

Настройка программы ПО АРМ администратора выполняется путём изменения переменных окружения. При этом считывание переменных окружения выполняется в файле core/variables.py:

```
import os


# Администратор ПУМ

network_admin_group = 'netadm'

# Администратор трафика ККП

traffic_admin_group = 'trafadm'

# Администратор устройств ККП

device_admin_group = 'devadm'

user_group = 'user'

# Адрес LDAP1

AUTH_LDAP_SERVER_URI1 = "ldap://" + os.getenv('LDAP_HOST1', 'ldap')

# Адрес LDAP2

AUTH_LDAP_SERVER_URI2 = "ldap://" + os.getenv('LDAP_HOST2', 'ldap')

AUTH_LDAP_BIND_DN = os.getenv('AUTH_LDAP_BIND_DN', '')
```

```
AUTH_LDAP_BIND_PASSWORD =
os.getenv('AUTH_LDAP_BIND_PASSWORD', '')

RABBITMQ_HOST = os.getenv('RABBITMQ_HOST', '')

RABBITMQ_USER = os.getenv('RABBITMQ_USER', '')

RABBITMQ_PASSWORD = os.getenv('RABBITMQ_PASSWORD', '')

KERBEROS_USER = os.getenv('KERBEROS_USER', '')

ZABBIX_HOST = os.getenv('ZABBIX_HOST', '')

ZABBIX_USER = os.getenv('ZABBIX_USER', '')

ZABBIX_PASSWORD = os.getenv('ZABBIX_PASSWORD', '')

ZABBIX_LINK = os.getenv('ZABBIX_LINK', '')

GLPI_LINK = os.getenv('GLPI_LINK', '')

GLPI_API = os.getenv('GLPI_API', '')

GLPI_APP_TOKEN = os.getenv('GLPI_APP_TOKEN', '')

GLPI_USERNAME = os.getenv('GLPI_USERNAME', '')

GLPI_PASSWORD = os.getenv('GLPI_PASSWORD', '')

BACKEND = os.getenv('BACKEND', 'HTTP')


EXPORTER_FTP_ADDRESS = os.getenv('EXPORTER_FTP_ADDRESS', "ftp")

EXPORTER_FTP_USER = os.getenv('EXPORTER_FTP_USER', "user")

EXPORTER_FTP_PASSWORD = os.getenv('EXPORTER_FTP_PASSWORD',
"password")


SWITCH_TABLE = os.getenv('SWITCH_TABLE', '')
```

Установка переменных окружения выполняется в файле docker-compose.yaml в разделе pum.environment.

2.7 Сервер и агенты ЛСУ

Программные агенты, установленные на ЛСУ, настраиваются автоматически при загрузке системы на ЛСУ. Конфигурация ОС на сервере ЛСУ получается в автоматическом режиме с сервера ПУМ при загрузке системы.

3 Проверка программы

3.1 Подсистема управления

3.1.1 Модуль управления ПДУ и КРК

Модуль управления ПДУ и КРК можно проверить послав запрос из АРМ администратора со следущим содержанием в очередь AMQP которую слушает модуль управления ПДУ и КРК:

{command: ping}

В случае если модуль работает, то придёт ответ вида

{ ok: True, {result: “pong”}}

3.1.2 Модуль управления трафиком, топологией и технологическими параметрами

3.2 Подсистема мониторинга

3.2.1 Модуль мониторинга ПДУ и КРК

Проверка программы выполняется путём определения статуса демона zabbix-server. После установки и запуска программы должен отображаться статус "active (running)". Для проверки выполнить команду:

$ sudo systemctl status zabbix-server

Проверка веб-интерфейса программы выполняется путём перехода на страницу мониторинга АРМ в браузере и попытке входа в систему. При этом выполняется проверка взаимодействия АРМ с базой данных мониторинга и LDAP.

В панели Мониторинг -> Последние данные нужно увидеть полученные данные мониторинга с актуальной датой и временем.

Проверка агента выполняется путём определения статуса демона zabbix-agent. После установки и запуска агента должен отображаться статус "active (running)". Для проверки выполнить команду:

$ sudo systemctl status zabbix-agent

3.2.2 Модуль мониторинга оборудования ККП

Проверка программы выполняется путём определения статуса демона zabbix-server. После установки и запуска программы должен отображаться статус "active (running)". Для проверки выполнить команду:

$ sudo systemctl status zabbix-server

Проверка веб-интерфейса программы выполняется путём перехода на страницу мониторинга АРМ в браузере и попытке входа в систему. При этом выполняется проверка взаимодействия АРМ с базой данных мониторинга и LDAP.

В панели Мониторинг -> Последние данные нужно увидеть полученные данные мониторинга с актуальной датой и временем.

Проверка агента выполняется путём определения статуса демона zabbix-agent. После установки и запуска агента должен отображаться статус "active (running)". Для проверки выполнить команду:

$ sudo systemctl status zabbix-agent

3.3 Подсистема технического учёта

3.3.1 Модуль учета оборудования

Проверка веб-интерфейса программы выполняется путём перехода на страницу технического учёта АРМ в браузере и попытке входа в систему. При этом выполняется проверка взаимодействия АРМ с базой данных технического учёта и LDAP.

В панели Настройки -> Общий -> Система под пунктом "Server" не должно быть ошибок.

Проверка агента выполняется путём определения статуса демона fusioninventory-agent. После установки и запуска агента должен отображаться статус "active (running)". Для проверки выполнить команду:

$ sudo systemctl status fusioninventory-agent

3.3.2 Модуль учета логических ресурсов

Проверка веб-интерфейса программы выполняется путём перехода на страницу технического учёта АРМ в браузере и попытке входа в систему. При этом выполняется проверка взаимодействия АРМ с базой данных технического учёта и LDAP.

В панели Настройки -> Общий -> Система под пунктом "Server" не должно быть ошибок.

Проверка агента выполняется путём определения статуса демона fusioninventory-agent. После установки и запуска агента должен отображаться статус "active (running)". Для проверки выполнить команду:

$ sudo systemctl status fusioninventory-agent

3.4 Подсистема взаимодействия с внешними системами

3.4.1 Модуль экспорта данных во внешние системы

Для проверки необходимо выполнить скрипт glpi_get_inv_data.py и получить техническую конфигурацию с названиями устройств, географических локаций, соединений в формате JSON с некоторыми комментариями.

Затем выполнить скрипт zbx_get_mon_data.py и получить данные мониторинга.

3.4.2 Модуль обработки запросов API в реальном времени

Для проверки необходимо выполнить скрипт pum-tester.py с параметрами "./pum-tester.py -d examples -t 0000-test-ping.yaml" и получить ответ вида "result is

matching expected result". В результате выполнения скрипта pum-tester.py в очередь сообщений брокера RabbitMQ посылается команда:

{command: ping}

В случае успешного завершения из ответной очереди RabbitMQ принимается ответ:

{ok: True, status: pong}

и выдается сообщение «result is matching expected result”.

3.5 Подсистема контроля доступа

Для проверки необходимо выполнить команду $ ldapsearch -b "dn=strela" и получить сведения о наличии учетных записей администраторов в каталоге ldap.

3.6 ПО АРМ администратора

Проверка программы выполняется путём перехода на страницу АРМ в браузере и выполнении входа в систему. При этом выполняется проверка взаимодействия АРМ с базой данных и LDAP.

3.7 Сервер ЛСУ и агенты ЛСУ

Каждый агент ЛСУ на сервере можно проверить путём его опроса из соответствующей подсистемы ПУМ. Состояние сервера ЛСУ можно проверить путем обращения к подсистеме мониторинга. Описание работы с системой мониторинга приведено в п. 4.2 Руководства пользователя ПУМ.

4 Руководство по инсталляции ПУМ

4.1 Мероприятия по подготовке ко вводу в действие ПУМ

4.1.1 Общий план мероприятий

а) Убедиться в наличии Приказа по организации-заказчику о создании комиссии по приемке ПУМ ККП к эксплуатации.
б) Провести инструктаж пользователей.
в) Подготовка помещения и других условий для тестирования.
г) Доставка компонентов оборудования до места тестирования оборудования и установка ПО ПУМ.
д) Распаковка, монтаж, проверка функционирования оборудования в подготовленных помещениях.
е) Установка ПО ПУМ.
ж) Тестирование оборудования и ПО ПУМ.
з) Упаковка оборудования, подготовка надписей на таре, куда должна быть выполнена доставка (предназначенные места) оборудования.
и) Доставка оборудования в предназначенные места.
к) Тестирование оборудования, установленное в предназначенных местах.
л) Проведение испытаний ПУМ в комплексе с вновь установленным ПО и оборудованием.

4.1.2 Ввод в действие ПУМ

Для ввода в действие ПУМ необходимо выполнить следующие шаги:

а) Убедиться, что поставленное компьютерное и коммуникационной оборудование функционирует в соответствии с техническими документами на поставленное оборудование.
б) Выполнить необходимое конфигурирование компьютерного и коммуникационного оборудования (ЦУМ).
в) На серверах Центра Управления и Мониторинга (ЦУМ) развернуть назначенные операционные системы и прикладное программное обеспечение.
г) На сервере в ЦУМ произвести регистрацию требуемого числа пользователей (администраторов ПУМ).
д) Удостовериться в готовности помещений, где будет размещаться оборудование ЦУМ.
е) Переместить сервер(ы) ЦУМ в назначенные помещения.
ж) Сконфигурировать

1) сетевые коммутаторы для транспортной сети передачи данных;
2) сетевые коммутаторы для сети управления;
3) модемы для сотовой связи
4) ЛСУ для работы в режиме загрузке по сети
5) Проверить работоспособность новой конфигурации на коммутаторах и модемах.

з) Проверить функционирование ЛСУ совместно с коммутаторами и модемами.

и) Подготовить к транспортировке коммутаторы, модемы, ЛСУ в назначенные географические местоположения.

к) Транспортировать и установить коммутаторы, модемы и ЛСУ в назначенном географическом расположении для каждого ЛСУ.
1) Установка и соединение кабелей должны производиться по разработанной инструкции.

л) Последовательно проверить функционирование установленных по месту компонентов:
1) сеть управления (все узлы);
2) Сеть передачи данных (все узлы);
3) Убедиться, что ЦУМ в Санкт–Петербурге обеспечивает управление функционированием сети управления и сети передачи данных в полном объёме:
– Доступ к каждому ЛСУ как по сети управления, так и по резервному каналу с использованием сотовой связи.
– Доступ к коммутатору сети передачи данных по сети управления и по резервному каналу.
– Удалённое включение и выключение электропитания на всех устройствах.
– Убедиться, что все действия записаны в логи времена в логах соответствуют выбранной временной зоне в ПУМ.
4) Аналогично провести работы по проверке ЦУМ в г. Москва.

м) На всех шагах следует заполнить протоколы результатов проверки.

н) Провести обучение сотрудников эксплуатационного подразделения.

о) После успешного выполнения перечисленных выше шагов можно переходить к тестовой эксплуатации ПУМ.

4.1.3 Примерный расчет рабочего времени на выполнение конфигурирования и тестирования

а) **ЛСУ**

1) Оценка работоспособности устройства 1 час на устройство.
2) Конфигурирование устройства – 2 часа
3) Соединение кабелей для проверки – 1 час.
4) Прогон теста – 2 час.
5) Итого примерно 6 часов на ЛСУ

б) **Коммутатор транспортной сети**

1) Оценка работоспособности – 1 час;
2) Конфигурирование – 1 час;
3) Подключение – 0.5 часа.
4) Тестирование – 2 часа
5) Итого 4.5 часа.

в) **Коммутатор сети управления**

1) Оценка работоспособности – 0.5 часа;
2) Конфигурирование 1 час;
3) Подключение 1 час часа
4) Тестирование 2 часа.
5) Итого 4.5 часа.

г) **Модем**

1) Оценка работоспособности и сборка 2 часа;
2) Конфигурирование 1 час;
3) Подключение 0.5 часа;
4) Тестирование 2 часа;
5) Итого 5.5 часа

д) **Комплексный тест всех ЛСУ, коммутаторов, модемов**

1) Проведение тестовых испытаний, снятие характеристик всей системы в комплексе – примерно 672 часа рабочего времени.

На проверку всех компонентов потребуется примерно 22-24 рабочих дня, т.е. полный календарный месяц. Далее комплексная проверка потребует от 1 до 2 календарных месяцев.

Итого на подготовку помещения потребуется примерно на 3 месяца. Отдельная подготовка программного обеспечения не требуется.

4.1.4 Требования к помещению для проведения испытаний ПУМ

- Закрывающееся помещение с кондиционером для поддержки 18-25 градусов с доступом только узкого круга лиц.
- Электропитание 220 вольт с суммарной разрешаемой мощностью 20 КВт.

- Число установленных работающих розеток на 220 вольт для подключения оборудования не менее 40 штук.
- Не менее 80 патч-кордов (витая пара) длиной не менее 10 метров категории 6.
- Как минимум одна стандартная стойка 48U.
- Число рабочих мест (столов) не менее 4.
- Общая площадь помещения (помещений) не менее 60 кв. метров.
- Не менее двух комплектов инструментов для работы с компьютерным оборудованием, например, типа «SK-LAN-2 - набор инструментов для обслуживания кабельных линий связи и Ethernet в рюкзаке (https://skomplekt.com/tovar/7/1/9999957410/).

4.2 Подсистема управления

4.2.1 Модуль управления ПДУ и КРК

Модуль управления ПДУ и КРК на сервере ЦУМ инсталлируется в составе АРМ администратора.

Часть выполняемая на ЛСУ автоматически включается в образ ОС на ЛСУ при запуске системы.

4.2.2 Модуль управления трафиком, топологией и технологическими параметрами

Модуль управления трафиком, топологией и технологическими параметрами устанавливается совместно с образом виртуальной машины(ВМ) на сервере ЦУМ.

В состав сервисов этой ВМ входит сервис TFTP. Через него осуществляется конфигурирование сетевых маршрутизаторов для построения необходимых каналов передачи данных. Запрос на реконфигурацию осуществляется SNMP запросом, посылаемым соответствующей программой, отвечающей за приём данных, формирование файлов конфигурации и инициации загрузки этих файлов в целевой маршрутизатор. Всё необходимое ПО предустановлено на ВМ и требует только донастройки по используемым IP адресам и сетевой архитектуры.

4.3 Подсистема мониторинга

Подсистема мониторинга функционирует на основе использования заимствованной программной системы Zabbix. Установка Zabbix выполняется в соответствии с инструкцией, расположенной на официальном сайте для Centos 8, базы данных Postgresql и веб-сервера Apache (https://www.zabbix.com/ru/download?zabbix=5.0&os_distribution=centos&os_version=8&db=postgresql&ws=apache). Там же описан способ установки агента.

Для взаимодействия с подсистемой контроля доступа применить патч zabbix-server-front-cas.patch, расположенный по адресу https://gitlab-ct.itmo.ru/strela-project/zabbix-configuration/-/blob/master/zabbix-server-front-cas.patch.

$ sudo patch /usr/share/zabbix/include/classes/user/CWebUser.php < zabbix-server-front-cas.patch

Добавить следующие строки в файл /etc/httpd/conf.d/zabbix.conf:

SetEnvIfNoCase remote_user "(.*)" PHP_AUTH_USER=$1

SetEnvIfNoCase remote_user "(.*)" PHP_AUTH_PW=$1

4.4 Подсистема технического учёта

Подсистема технического учета функционирует на основе использования заимствованной программной системы GLPI. Установка GLPI выполняется в соответствии с инструкцией, расположенной на официальном сайте для Centos 8, базы данных MariaDB и веб-сервера Apache (https://glpi-install.readthedocs.io/en/latest/install/index.html).

Для взаимодействия с агентом на сервере устанавливается программный пакет Fusioninventory-for-glpi в соответствии с инструкцией, расположенной на официальном сайте для Centos 8 и веб-сервера Apache (https://fusioninventory.org/documentation/fi4g/installation.html).

Установка агента выполняется в соответствии с инструкцией, расположенной на официальном сайте для Linux (https://fusioninventory.org/documentation/agent/installation/linux/rhel.html).

4.5 Подсистема взаимодействия с внешними системами

4.5.1 Модуль экспорта данных во внешние системы

Модуль экспорта данных во внешние системы устанавливается вместе с подсистемами, из которых экспортируются данные. Затем выполняется конфигурирование согласно разделу 3.

4.5.2 Модуль обработки запросов API в реальном времени

Установка модуля обработки запросов API в реальном времени производится вместе с ПО АРМ администратора согласно подразделу 7.8.

4.6 Подсистема контроля доступа

Для инсталляции подсистемы контроля доступа должна быть выполнена установка пакетов nginx, mod_auth_cas и произведено конфигурирование согласно разделу 3.

4.7 ПО АРМ администратора

Для инсталляции ПО АРМ администратора необходимо:

– обеспечить доступ до сети управления, выполнить скачивание проекта ПО АРМ на сервер,

– выполнить установку пакетов podman, podman-compose,

– перейти в директорию arm, выполнить конфигурирование согласно разделу 3,

– выполнить podman-compose up –build -d.

4.8 Сервер и агенты ЛСУ

Образ ОС для сервера ЛСУ собирается с помощью автоматизированного инструментария по сборке образов ОС. Описание сборки производится в виде текстовых файлов в формате yaml и/или bash. При этом в образ включаются программные агенты ЛСУ и настраивается их автоматический запуск. Полученный образ упаковывается, и добавляется в систему конфигурации загрузки ЛСУ. Загрузка образа на сервер ЛСУ происходит в автоматизированном режиме в оперативную память сервера ЛСУ.

5 Дополнительные возможности

Дополнительные возможности не предусмотрены.

6 Сообщения системному программисту

Таблица 1. Сообщения системному программисту

Текст сообщения	Причина возникновения сообщения	Требуемое действие
Невозможно выполнить вход в АРМ	Вход не удаётся выполнить при вводе корректного пароля	Необходимо удостовериться в работоспособности системы LDAP
Невозможно перейти по ссылке Поддержка	Отсутствует возможность входа в подсистему технического учёта	Необходимо удостовериться в работоспособности подсистемы технического учёта
Невозможно перейти по ссылке Мониторинг	Отсутствует возможность входа в подсистему мониторинга	Необходимо удостовериться в работоспособности подсистемы мониторинга
PostgreSQL database "zabbix" on <Хост БД>:<Порт БД> is not available: <error message depending on the type of DBMS (database)>	База данных Zabbix недоступна сервером Zabbix.	Устранить причину недоступности базы данных Zabbix с хоста Zabbix сервера.

Database error: Error connecting to database.	База данных Zabbix недоступна веб-панелью Zabbix.	Устранить причину недоступности базы данных Zabbix с хоста веб-панели Zabbix.
Zabbix сервер не запущен: отображаемая информация может быть не актуальной.	Zabbix сервер недоступен веб-панелью Zabbix.	Устранить причину недоступности Zabbix сервера с хоста веб-панели Zabbix. Или включить Zabbix сервер.
You are not logged in: - Session terminated, re-login, please. - Login name or password is incorrect.	Пользователь не авторизован в Zabbix.	Заново ввести логин и пароль в Zabbix.
Access denied: - You are logged in as "ИМЯ_ПОЛЬЗОВАТЕЛЯ". You have no permissions to access this page. - If you think this message is wrong, please consult your administrators about getting the necessary permissions.	Недостаточно прав пользователя для совершения действия в Zabbix.	Получить необходимые права у администратора Zabbix.
A link to the SQL server could not be established. Please check your configuration.	База данных GLPI недоступна веб-панелью GLPI.	Устранить причину недоступности базы данных GLPI с хоста веб-панели GLPI.

У Вас нет прав для выполнения этой операции.	Недостаточно прав пользователя для совершения действия в GLPI.	Получить необходимые права у администратора GLPI.
'ACCESS_REFUSED - Login was refused using authentication mechanism PLAIN. For details see the broker logfile.	В доступе к API было отказано.	Ввести корректные данные входа из внешней системы.
result and expected result don't match	результат и ожидаемый результат выполнения команды не совпадают	Выполнять только известные команды

Перечень принятых сокращений

API	Application programming interface
CWDM	Coarse Wavelength Division Multiplexing
EVM	Extended Verification Module
IMA	Integrity Measurement Architecture
IMA/EVM	Integrity Measurement Architecture and Extended Verification Module
QBER	Quantum Bit Error Rate
QKD	Quantum key distribution
SLA	Service Level Agreement
SNMP	Simple Network Management Protocol
АРМ	Автоматизированное рабочее место
БД	База данных
ВМ	Виртуальная машина
КЗК	Квантово-защищенные ключи
ККП	Квантовая коммуникационная платформа цифровой экономики
КРК	Квантовое распределение ключей
ЛСУ	Локальный сервер управления
ОУ	Опорный узел
ПДУ	Подсистема организации доверенных опорных узлов квантовой связи
ПКРК	Подсистема квантового распределения ключей
ПО	Программное обеспечение
ПОУ	Промежуточный опорный узел

ПУМ	Подсистема управления и мониторинга квантовой сетью
СКЗИ	Система защиты информации
ЦУМ	Центр Управления и Мониторинга

Лист регистрации изменений

Изм	новых				Всего листов (страниц) в докум.	№ документа	Входящий номер сопроводи-тельного	Подпись	Дат а
	изменен-ных	заменен-ных	новых	изъятых					

Подп. и дата | Взам. инв. № | Инв. № дубл. | Подп. и дата | Инв. № подл

Ли	Изм.	№ докум.	Подп.	Дат		Лист