iTman Discovery

Установка системы

Версия: 5.5

Москва

06.12.2024

Оглавление

1. Технические требования

1.1. Системные требования

- Windows
  - 64x разрядные версии систем:
  - Windows Server 2008 R2 x64, 2012, 2012 R2, 2012 R2, 2016, 2019, 2022
  - Windows Server Core 2008, 2008 R2 x64, 2012, 2012 R2
  - Windows Server Standard (ранее известный как Windows Server Core) 2016, 2019
  - Windows 7, 8, 10, 11
- Linux
  - 64x разрядные версии систем:
  - Astra Linux 1.7 и выше
  - Astra Linux CE
  - Debian 9 и выше
  - Ubuntu 18.10 и выше
  - CentOS 7
  - Oracle Linux Server 8.3
  - Red Hat Enteprise 8
  - Red Hat Enterprise Linux 7.4
  - Suse Linux 15.5 и выше

1.2. Аппаратные требования

Требования к оборудованию зависят от количества инвентаризируемых клиентских компьютеров в среде и наличия дополнительных модулей в системе.

1.2.1. Минимальные требования

Управляющий сервер:

- Процессор: 2 CPU, AMD® A8 3870 3,6 Ghz или Intel® Core ™ i3 2100 3.1Ghz
- Память: 8 ГБ RAM + 8 ГБ RAM при работе распознавания правил в Библиотеке ПО (от 50000 правил)
- Хранилище (объем диска): 10 ГБ

АРМ:

- Должно быть выделено 1 ЦП/Ядра
- Память: 2 ГБ RAM
- Хранилище (объем диска): 100 МБ

1.2.2. 1 — 10 000 клиентских компьютеров / агентов

	Компоненты и модули системы	Память (ГБ)	ЦП/Ядра	Хранилище (объем диска)
1	Discovery Server	4	1/2	20ГБ
2	Discovery Interact	2	1/2	5ГБ
3	Discovery Agent (не получающий данные от других агентов)	0,5	1/2	2ГБ
4	Discovery Agent (выполняет роль шлюза, т.е. получает данные от других агентов)	Аналогично Discovery Interact		
5	Discovery Agent+ SCCM (без файлов)	2,5	1/2	5ГБ
6	Discovery Agent+ SCCM (со сбором файлов)	5	1/2	10ГБ
7	Модуль: Библиотека ПО (до 50000 правил)	8	1/2	5ГБ
8	Модуль: Библиотека ПО (от 500000 правил)	10	1/2	10ГБ

1.2.3. 10 000 — 20 000 клиентских компьютеров / агентов

	Компоненты и модули системы	Память (ГБ)	ЦП Ядра	Хранилище (объем диска)
1	Discovery Server	16	1/4	30ГБ
2	Discovery Interact	4	1/4	10ГБ
3	Discovery Agent (не получающий данные от других агентов)	0,5	1/2	2ГБ
4	Discovery Agent (выполняет роль шлюза, т.е.	Аналогично Discovery Interact		

	получает данные от других агентов)			
5	Discovery Agent+ SCCM (без файлов)	4,5	1/4	10ГБ
6	Discovery Agent+ SCCM (со сбором файлов)	12	1/4	15ГБ
7	Модуль: Библиотека ПО (до 50000 правил)	8	1/4	10ГБ
8	Модуль: Библиотека ПО (от 500000 правил)	10	1/4	15ГБ

1.2.4. 20 000 — 50 000 клиентских компьютеров / агентов

	Компоненты и модули системы	**Память (ГБ)**	**ЦП Ядра**	**Хранилище (объем диска)**
1	Discovery Server	32	1/4	40ГБ
2	Discovery Interact	8	1/4	15ГБ
3	Discovery Agent (не получающий данные от других агентов)	0,5	1/2	2ГБ
4	Discovery Agent (выполняет роль шлюза, т.е. получает данные от других агентов)	Аналогично Discovery Interact		
5	Discovery Agent+ SCCM (без файлов)	10,5	1/4	15ГБ
6	Discovery Agent+ SCCM (со сбором файлов)	14	1/4	20ГБ
7	Модуль: Библиотека ПО (до 50000 правил)	8	1/4	15ГБ
8	Модуль: Библиотека ПО (от 500000 правил)	12	1/4	15ГБ

1.2.5. 50 000 — 70 000 клиентских компьютеров / агентов

	Компоненты и модули системы	**Память (ГБ)**	**ЦП Ядра**	**Хранилище (объем диска)**
1	Discovery Server	32	1/8	50ГБ
2	Discovery Interact	16	1/8	20ГБ
3	Discovery Agent (не получающий данные от других агентов)	1	1/4	4ГБ

4	Discovery Agent (выполняет роль шлюза, т.е. получает данные от других агентов)	Аналогично Discovery Interact		
5	Discovery Agent+ SCCM (без файлов)	14,5	1/4	20ГБ
6	Discovery Agent+ SCCM (со сбором файлов)	18	1/8	30ГБ
7	Модуль: Библиотека ПО (до 50000 правил)	12	1/8	20ГБ
8	Модуль: Библиотека ПО (от 500000 правил)	16	1/8	20ГБ

1.2.6. 70 000 — 100 000 клиентских компьютеров / агентов

	Компоненты и модули системы	**Память (ГБ)**	**ЦП Ядра**	**Хранилище (объем диска)**
1	Discovery Server	64	1/16	128ГБ
2	Discovery Interact	32	1/8	30ГБ
3	Discovery Agent (не получающий данные от других агентов)	2	1/4	10ГБ
4	Discovery Agent (выполняет роль шлюза, т.е. получает данные от других агентов)	Аналогично Discovery Interact		
5	Discovery Agent+ SCCM (без файлов)	20,5	1/4	30ГБ
6	Discovery Agent+ SCCM (со сбором файлов)	32	1/16	35ГБ
7	Модуль: Библиотека ПО (до 50000 правил)	16	1/16	30ГБ
8	Модуль: Библиотека ПО (от 500000 правил)	32	1/16	30ГБ

1.2.7. Рекомендуемые требования для SCCM

При развертывании задач инвентаризации SCCM:

- Рекомендуется предоставить 16+ GB ОЗУ. Это позволит обработать 30 000+ устройств из базы SCCM.

При развертывании задач инвентаризации SCCM и файлов:

- Рекомендуется предоставить +12 GB ОЗУ.

Важно. Заполнение ОЗУ свыше 85% сильно замедляет выполнение задач. Сбор данных и логирование сохраняют все результаты.

При длительной эксплуатации или при развертывании задач инвентаризации SCCM и сетевой инвентаризации потребуется больше места (лучше 20+ GB) или очистка каталогов:

..\Discovery.Reader\{companyGuid}\TasksData

..\Discovery.Server\{companyGuid}\TasksData

..\Discovery.Agent\Data

1.3. Требования к SSL

Для использования SSL необходимо иметь действительный сертификат SSL. Сертификат выдается удостоверяющим центром (CA) и содержит информацию о владельце сертификата и открытый ключ для шифрования данных.

Поддержка браузеров: для обеспечения безопасного соединения с сервером, веб-браузеры должны поддерживать SSL и иметь доверительные центры сертификации в своих списке доверенных. Обратите внимание: SSL сертификат, полученный через Госуслуги, работает только в браузерах, которые поддерживают сертификаты российского удостоверяющего центра:

- Яндекс.Браузер
- Google chrome
- Atom

Тип сертификата: одноименный сертификат (Single Domain Certificate) для одного домена – минимальный. Для сервера и интеракта, установленных на одном устройстве.

Ключевая длина: рекомендуется использовать ключи длиной 2048 бит или более.

Формат сертификата: x.509.

Шифрование: SSL использует криптографические алгоритмы для защиты данных во время передачи. Используемые алгоритмы шифрования: RSA и AES.

1.4. Требования для SNMP

Для сбора данных с принтеров, сетевых коммутаторов, маршрутизаторов или ИБП используется протокол **SNMP версии 2**.

Для того чтобы реализовать сбор вам необходимо настроенное SNMP и также ввод **community name**. Сбор данных настраивается через задачу **Сетевой инвентаризации**.

1.5. Учетные записи и привилегии

Требования к учетным записям зависят от платформы:

Windows	**Linux**
Агент должен работать под системной учетной записью **local SYSTEM**	Агент должен работать от имени пользователя **root**

Так же при инвентаризации вам будет необходим PowerShell от версии 6.x и выше.

Для удалённого сбора данных Hyper-V / VMM должны быть доступны следующие порты:

- WMI - tcp/135,
- WinRM - tcp/5986,
- tcp - udp/49152-65535

Для сбора данных о виртуальных машинах на хостах VMware ESXi необходимо создать учетную запись пользователя с соответствующими правами доступа.

Чтобы получить доступ к данным о виртуальных машинах на хостах ESXi, учетной записи пользователя необходимо предоставить следующие права и роли:

- Права на чтение данных о виртуальных машинах, хостах, кластерах и датацентрах.
- Роль VirtualMachine.GuestOperations в соответствующих объектах. Это позволит выполнять операции на уровне гостевой операционной системы виртуальных машин.
- Роль Datastore.Browse для просмотра данных хранилища данных, используемых виртуальными машинами.

Кроме того, для удаленного доступа к данным на хостах ESXi необходимо настроить доступ по протоколу SSH.

Важно отметить, что доступ к данным на хостах ESXi по протоколу SSH может быть потенциально опасным, поэтому необходимо обеспечить соответствующую безопасность и защиту доступа к учетной записи пользователя.

2. Установка на Windows

2.1.Порядок действий при установке на ОС Windows

1. Определите компьютер, на которой будете устанавливать сервер, там будут храниться данные сервера и инвентаризации.
   Настройте к нему доступ по статическому IP адресу.
2. Организуйте хранение данных:
   - Установите БД **Postgre**
   - Создайте в БД пользователя с правами **суперадмин**, логин и пароль этого пользователя потребуются при установке системы.
3. Установите **Microsoft .NET 7.0 - Windows Server Hosting** на компьтеры, где будет установлен сервер и агенты.
4. Выпустите **SSL сертификат**, см. требования к сертификату.
5. Установите nginx на компьютер, где будет установлен сервер и выполните настройку.
6. Установите Discovery Server и Interact (совместная установка).
7. Настройте шифрование в Interact.
8. Установите **C++ Redistributable** на компьютеры, где будет установлены агенты.
9. Установите агентов.
10. Запустите Discovery Server.
11. Проверьте работу.

2.2.Дистрибутив

Для установки системы на Windows вам потребуется:

1. **Лицензионный ключ** (GUID компании)
2. Пакет приложения в zip, который содержит:
   - **DiscoveryServer_***.zip** или **.msi**
   - **DiscoveryWeb_***.zip** (если используется DiscoveryServer_***.zip).

- **DiscoveryAgentNative.msi** - агент с функцией сетевой инвентаризации. Это позволяет сканировать сеть, AD и SQL. Без нее агент будет сканировать только машину на которой он установлен.
- **DiscoveryAgentLightNative.msi** - агент без функции сетевой инвентаризации.

2.3. Установка и настройка nginx на ОС Windows

- Установите **nginx** на компьтер, где будет установлен сервер.
- Скопируйте директорую **html** из предоставленного дистрибутива **DiscoveryWeb_***.zip** в директорию установленного nginx **/www/html**.
- Внесите изменения в настройки **nginx** в файле **nginx.conf:**

```
server {
  listen 81 ssl
  location / {
    add_header Access-Control-Allow-Origin *;
    try_files $uri /index.html;
```

- Создайте директорию для хранения SSL сертификата и перенесите в нее сертификат и ключ.
- Добавьте в файле **nginx.conf** данные о SSL сертификате:

```
server {
  listen 81 ssl
  ssl_certificate      ./ssl/rootCA.pem;
  ssl_certificate_key  ./ssl/rootCA.key;
```

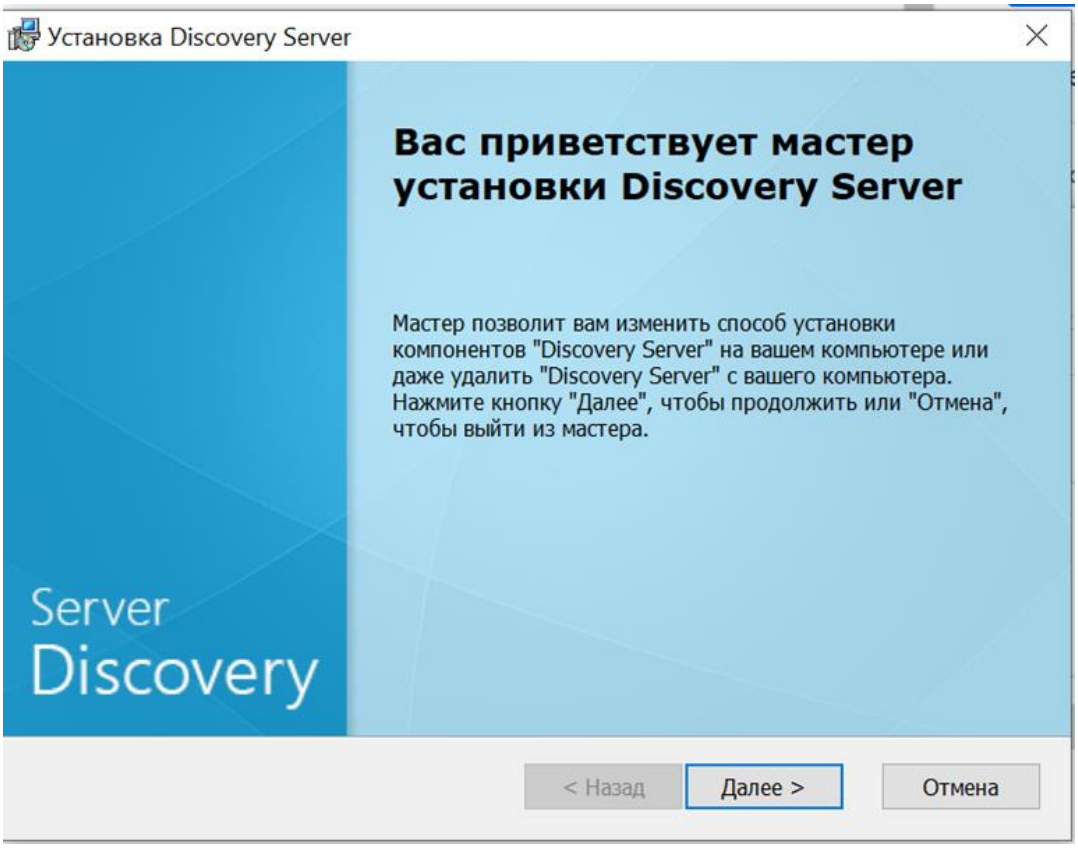
- Если у вас будут свои разрешенные порты, то вы их можете изменить их в этом файле **/www/html/config.json.**

```
{
  "BASE_API_PORT": "5047",
  "READER_API_PORT": "5087",
  "WRITER_API_PORT": "5097",
  "DEFAULT_INSTACE": "BaseInstance",
  "FLAG":""
}
```

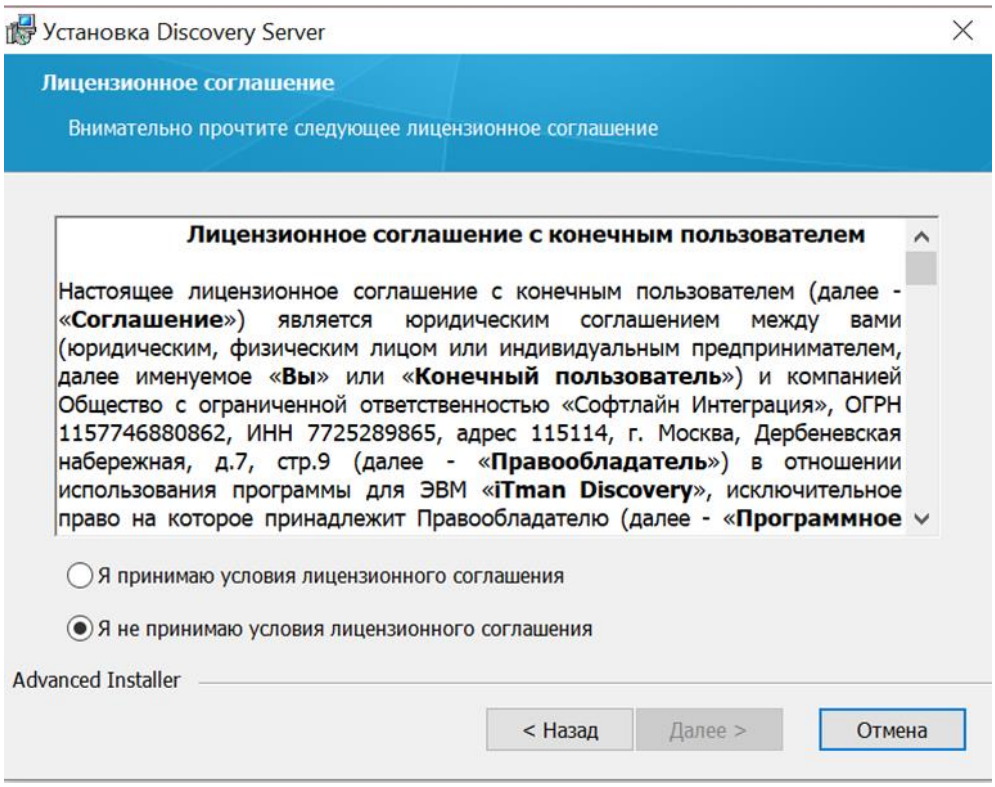
2.4. Установка Discovery Server на ОС Windows

- Запустите установочный файл **DiscoveryServer_***.msi** из-под учетной записи пользователя с правами администратора.

- Откроется окно Мастера установки.

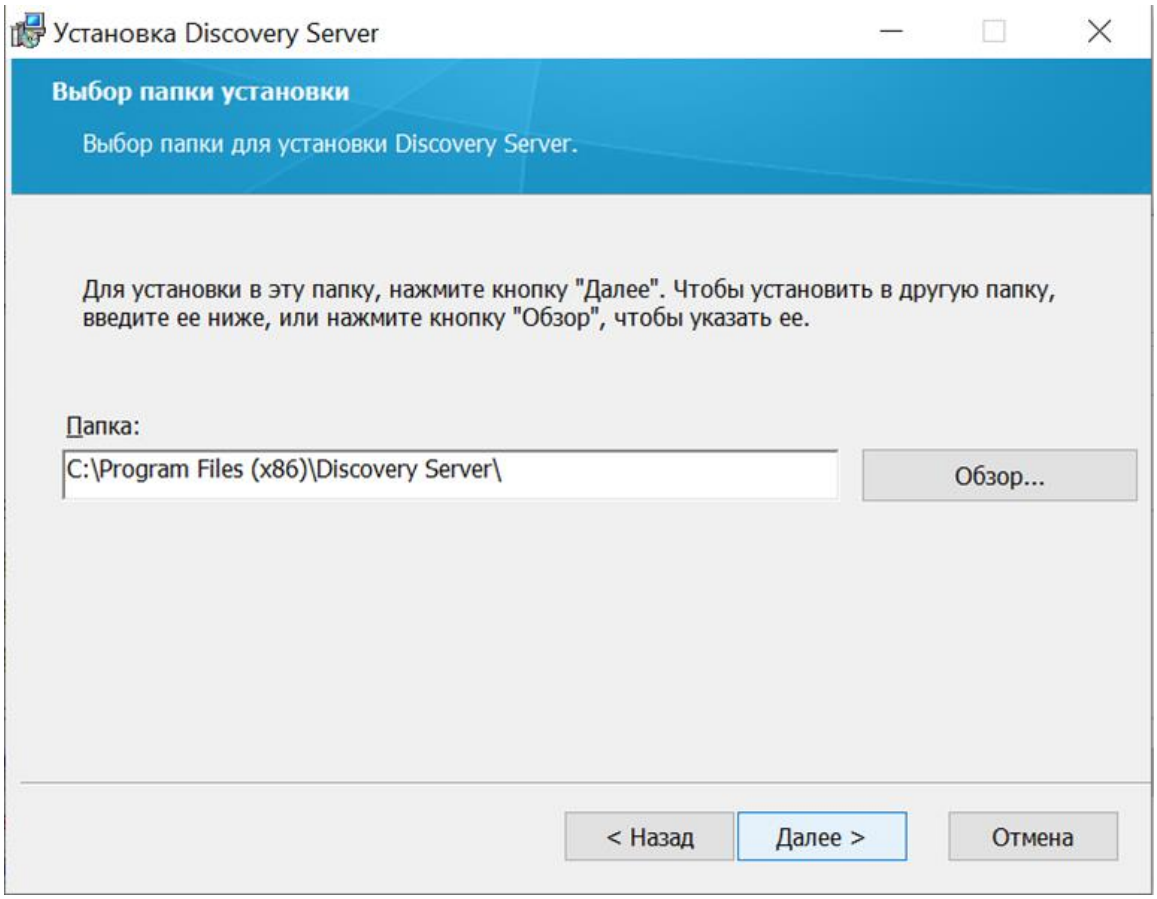


- Нажмите кнопку **Далее**.
- Откроется окно подтверждения лицензионного соглашения.

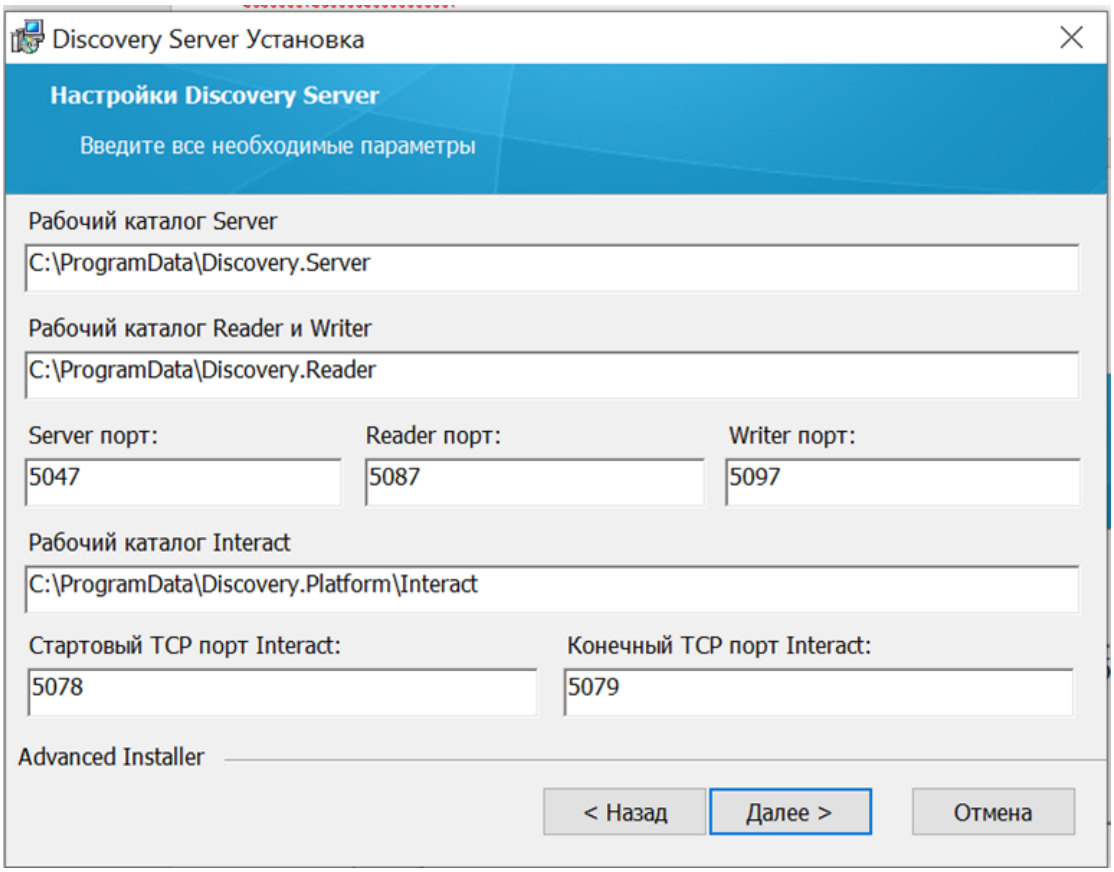


- Ознакомьтесь с условиями соглашения.
- В случае согласия с ними выберите вариант ответа "Я принимаю условия лицензионного соглашения".

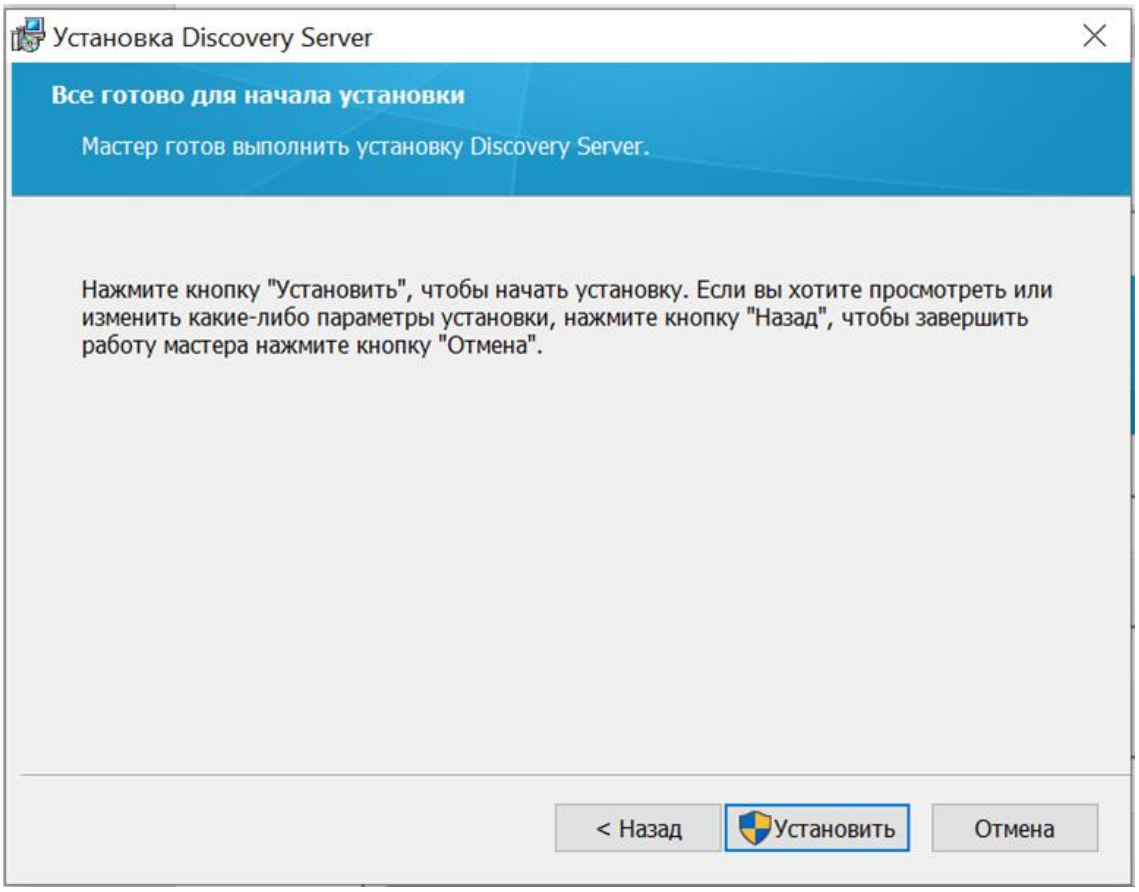
  Если вы не согласны с условиями лицензионного соглашения, то продолжить установку невозможно.
- Нажмите кнопку **Далее**.
- Откроется окно выбора пути установки сервера.



- Укажите путь установки на своем компьютере и нажмите **Далее**.
- Откроется окно настройки параметров установки.



- Заполните настройки:
    - Рабочий каталог сервера - это директория, в которой сохраняются данные сервера при работе с приложением
    - Рабочий каталог Reader и Writer
    - Server порт
    - Reader порт
    - Writer порт
    - Рабочий каталог Interact - это директория, в которой сохраняются данные о задачах распределенных между агентами и результаты, которые еще не попали в обработку Reader
    - Стартовый TCP порт Interact
    - Конечный TCP порт Interact
- Нажмите кнопку **Далее**.
- Откроется окно подтверждения установки.



- Нажмите **Установить**, чтобы запустить установку.

Если после установки вы захотите отредактировать данные, то вы можете это сделать в файлах настройки.

2.5. Установка Discovery Interact на ОС Windows

2.5.1. Порядок действий

Интеракт на Windows устанавливается автоматически при установке сервера.

Далее необходимо включить в нем шифрование, как описано далее.

Если вы хотите внести изменения в параметры установки после ее завершения, то это можно сделать в конфигурационном файле.

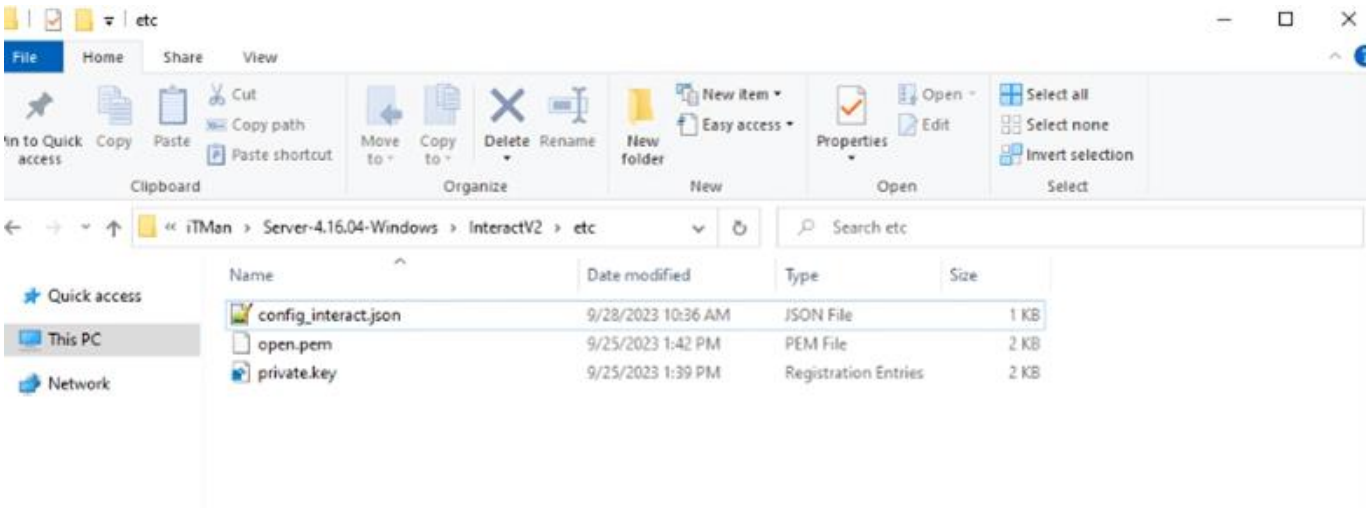
2.5.2. Включение шифрования

Для включения шифрования сделайте настройки SSL:

- Откройте директорию **Discovery Server** (путь по умолчанию: C:\Program Files\Discovery

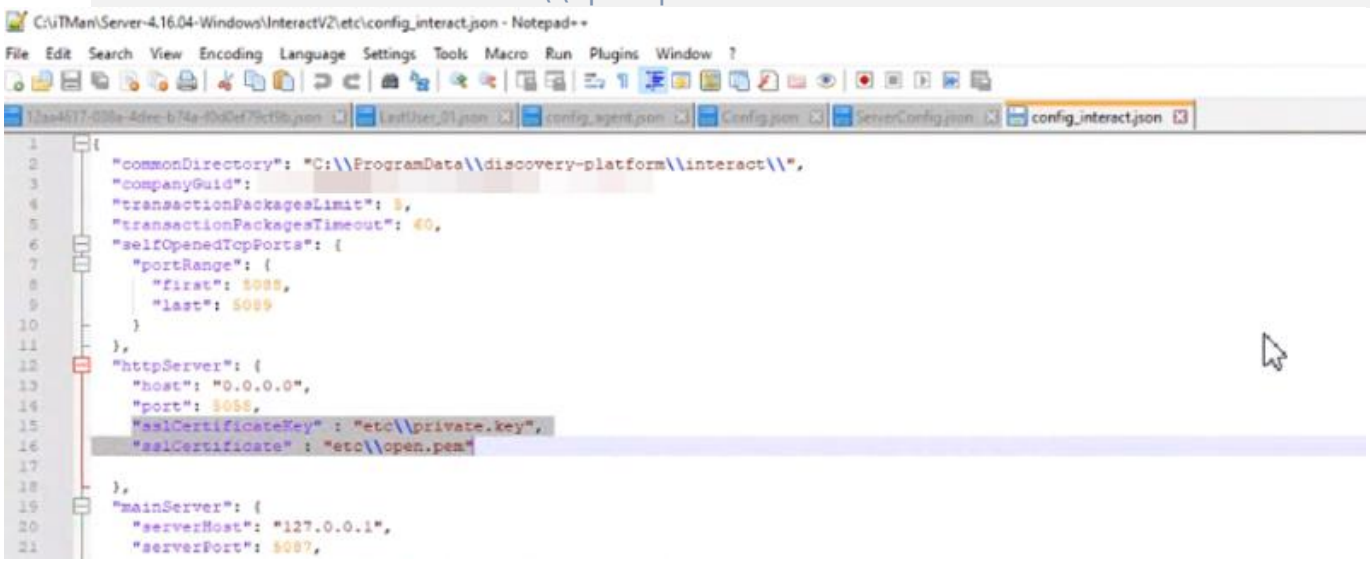
Server\Server).

- Перейдите в директорию **Interact/etc.**
- Сохраните ключ и сертификат для SSL.



- Откройте файл **config_interact.json** от имени администратора и добавьте информацию о сертификатах, как показано в примере (наименования файлов, расширения и путь к ключам у вас будут свои):

```
  "sslCertificateKey" : "etc\\private.key",
  "sslCertificate" : "etc\\open.pem"
```



2.6. Установка Discovery Agent на ОС Windows

В системе есть агенты двух типов:

- **DiscoveryAgentNative** – агент с функцией сетевой инвентаризации, позволяет сканировать сеть, AD и SQL.
- **DiscoveryAgentLiteNative** – агент без функции сетевой инвентаризации. Такой агент будет сканировать только компьютер, на котором он установлен.

При установке используйте файл в соответствии с типом агента.

Помимо этого, агент может быть:

- **Конечной точкой обработки данных** - в этом случае агент может только отправлять данные интеракту.
- **Работать в цепочке агентов** - в этом случае агент может получать данные от других агентов и передавать их интеракту.

2.6.1. Настройка учетных записей

Установка агента должна запускаться под **учетной записью с правами локального администратора** для установки программного обеспечения. При желании эта учетная запись может быть любой из:

- Уникальная учетная запись для каждого целевого устройства инвентаризации.
- Учетная запись, известная логической группе целевых устройств.
- Учетная запись администратора домена с правами установки на всех целевых устройствах.

В каждом случае учетные данные учетной записи должны быть сохранены в диспетчере паролей на принимающем маяке инвентаризации. Диспетчер паролей позволяет фильтровать учетные данные для группы устройств, например, путем сопоставления шаблонов с именами компьютеров.

При дальнейшей работе агенты инвентаризации работают, как **локальная системная учетная запись**. При работе агента под другой ролью, мы не гарантируем инвентаризацию данных, так как у системы может не хватить прав.

2.6.2. Установка C++ Redistributable

Установите **C++ Redistributable**, если не было установлено ранее.

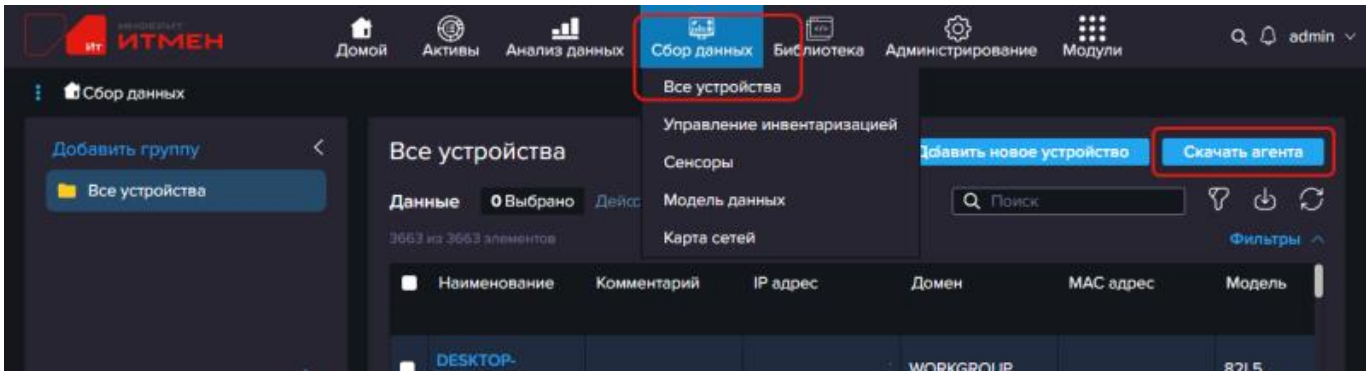
2.6.3. Получение файла установки

Перед установкой получите файл агента для установки.
Для этого:

- Возьмите файл из дистрибутива.
- Или скачайте через веб-интерфейс системы в разделе **Сбор данных / Все устройства**.

Обратите внимание, агенты бывают двух типов, вам нужно использовать только один из файлов, в зависимости от функций, которые будет выполнять агент:

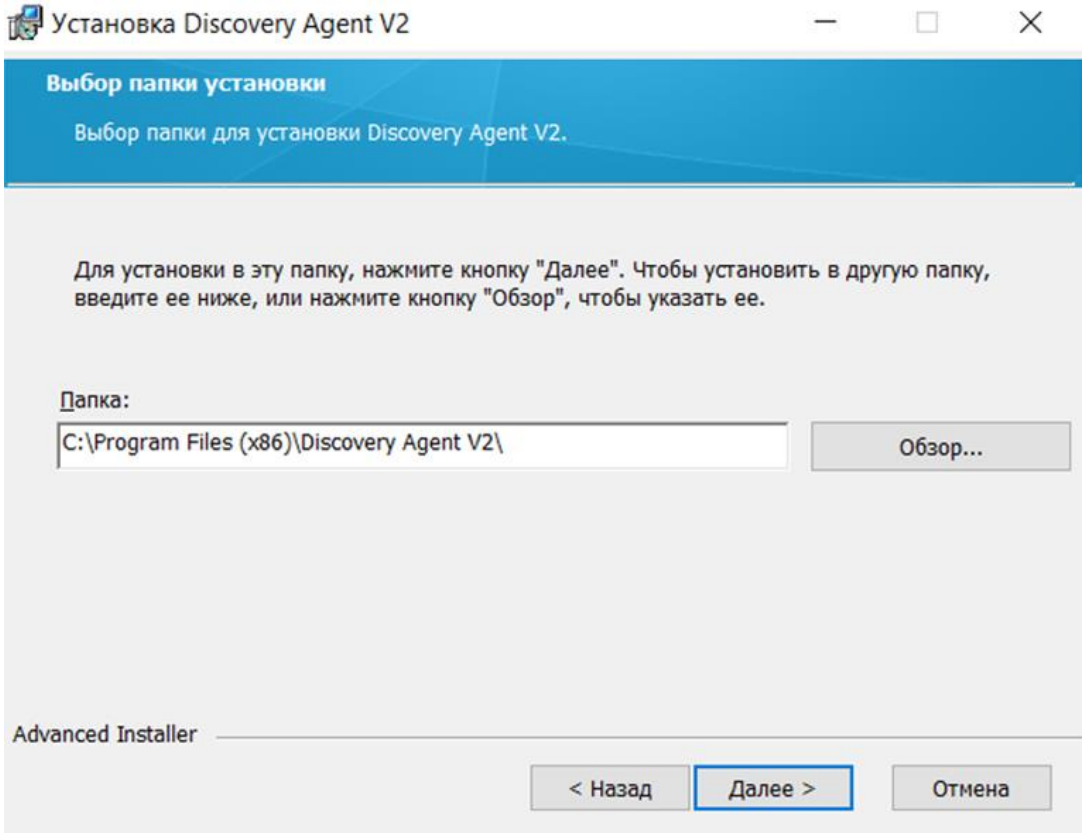
- **DiscoveryAgentNative.msi** – агент с функцией сетевой инвентаризации, позволяет сканировать сеть, AD и SQL. Используйте этот тип агента, если он должен получать данные от других агентов, т.е. работать в цепочке.
- **DiscoveryAgentLiteNative.msi** – агент без функции сетевой инвентаризации. Такой агент будет сканировать только компьютер, на котором он установлен.



2.6.4. Установка агента через интерфейс

Начало установки

- Запустите установочный файл, который вы получили, из-под учетной записи пользователя с правами администратора.
- Откроется окно Мастера установки.
- Нажмите кнопку **Далее**.
- Откроется окно выбора пути установки агента.

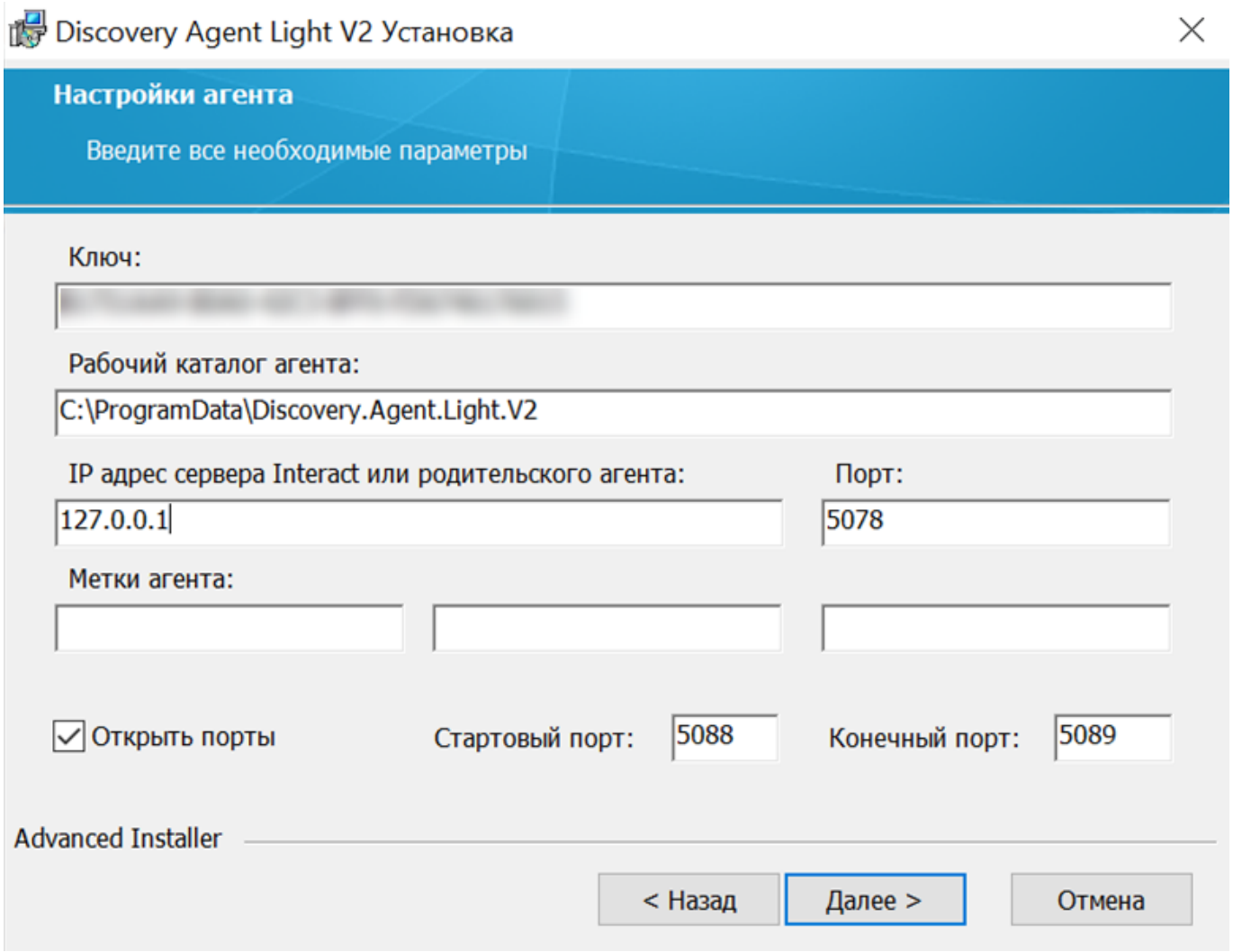


- Укажите путь установки на своем компьютере и нажмите **Далее**.

Ввод параметров агента

После выбора папки установки:

- Откроется окно настройки параметров агента.



- Заполните настройки:
    - **Ключ** – GUID ключ **вашей компании** (вида NNNNNNNN-NNNN-NNNN-NNNN-NNNNNNNNNNNN), который вы получили при покупке системы.
    - **Рабочий каталог агента** – путь сохранения временных данных при инвентаризации.
    - **IP адрес сервера Interact или родительского агента** - IP адрес родительского элемента (интеракта или агента-шлюза), который должен получать данные от

настраиваемого агента. Если интеракт и агент установлены на одном компьютере, то укажите 127.0.0.1. Если агент устанавливается на VDI, то укажите адрес устройства, на котором установлен VDI.

- **Порт** - порт родительского элемента (интеракта или агента-шлюза), который должен получать данные от настраиваемого агента (по умолчанию для интеракта: 5078, для агента-шлюза: 5088).
- **Метки агента (теги)** - специальные метки (теги) компьютера, на который устанавливается агент (если есть необходимость), например, VDI. В дальнейшем они используются при обогащении данных в процессе инвентаризации.
- **Открыть порты**:
  - Включите, если агент работает в цепочке из других агентов и может получать данные от других агентов, т.е. выполняет роль агента-шлюза.
  - Выключите, если агент является конечной точкой и никакой другой агент не будет ссылаться на текущий.
- **Стартовый порт, конечный порт** - начальный и конечный порты настраиваемого агента, открытые на прослушивание.
  Используется, если включено открытие портов агента, то есть агент участвует в цепочке и может получать данные от других агентов. Проверьте, что указанные порты не заняты. Например, если сервер и агент установлены на одном компьютере, и вы укажите порт, который занят сервером, то агент не будет работать корректно. Если открытие портов выключено, то настроенные порты игнорируются, даже если заполнены.

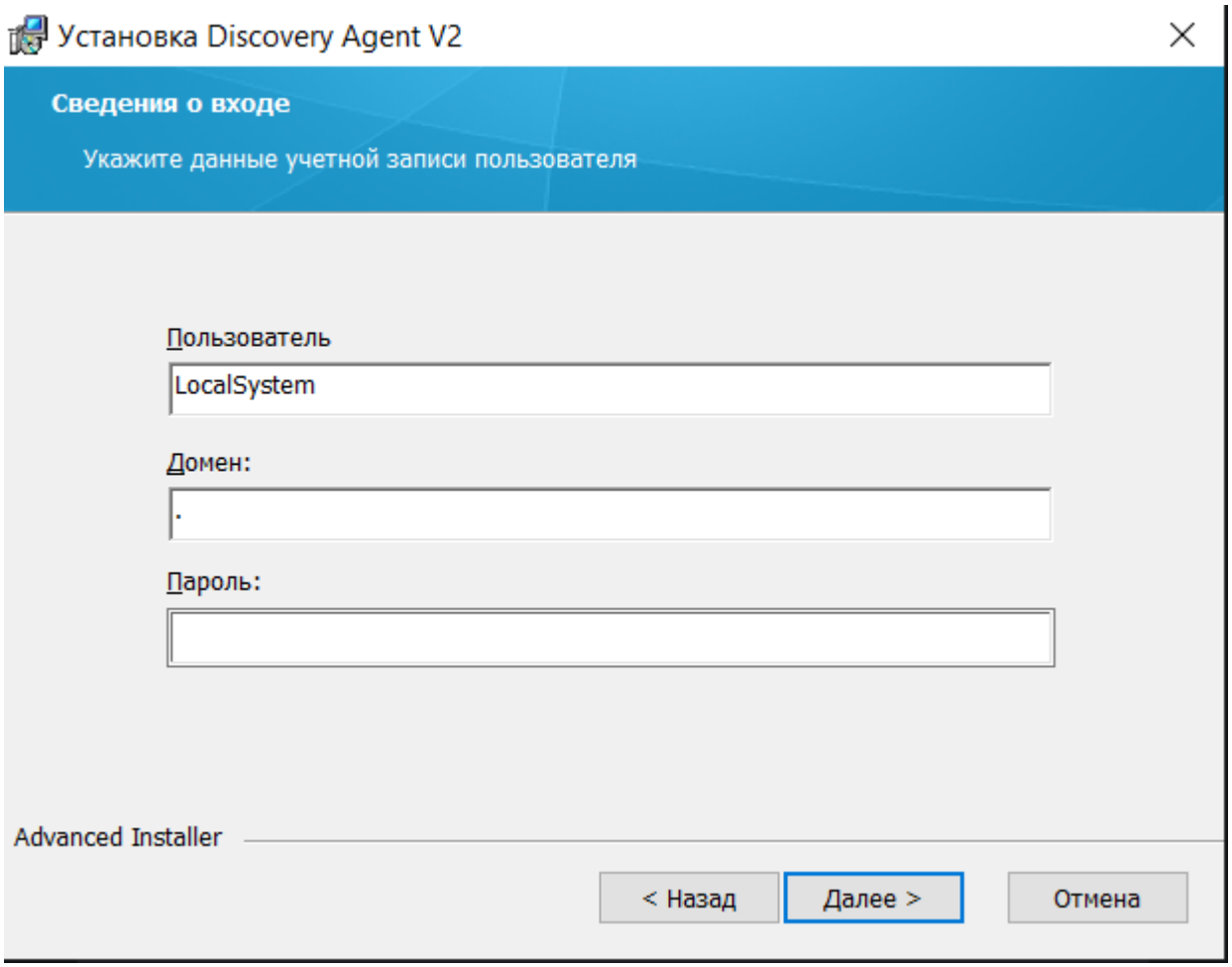
Если после установки агента, вы захотите изменить его параметры, то это можно сделать в файле конфигурации: config-agent.json.

- Нажмите кнопку **Далее**.

Ввод учетной записи пользователя

После заполнения параметров агента:

- Откроется окно настройки учетной записи пользователя.

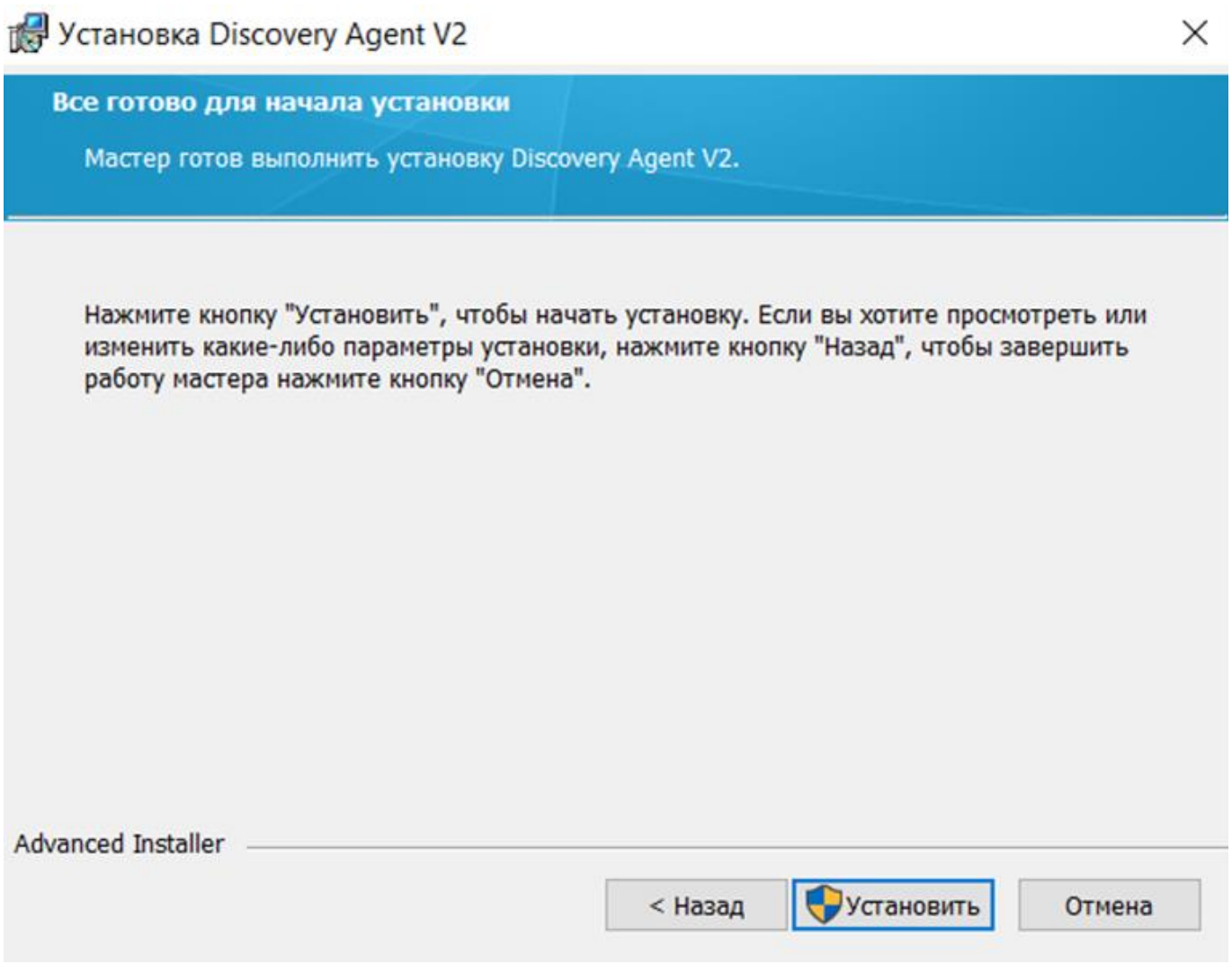


- Укажите учетные данные пользователя, под которым будет проходить инвентаризация.
  Рекомендуется оставить учетные данные по умолчанию (LocalSystem).
  Учетная запись LocalSystem — это предопределенная локальная учетная запись, используемая диспетчером управления службами.
  Обратите внимание, после завершения установки изменить учетную запись будет нельзя. Если требуется указать другую, то необходимо переустановить агент. В случае, если вы указываете другую учетную запись, то она должна отвечать требованиям.
- Нажмите кнопку **Далее**.

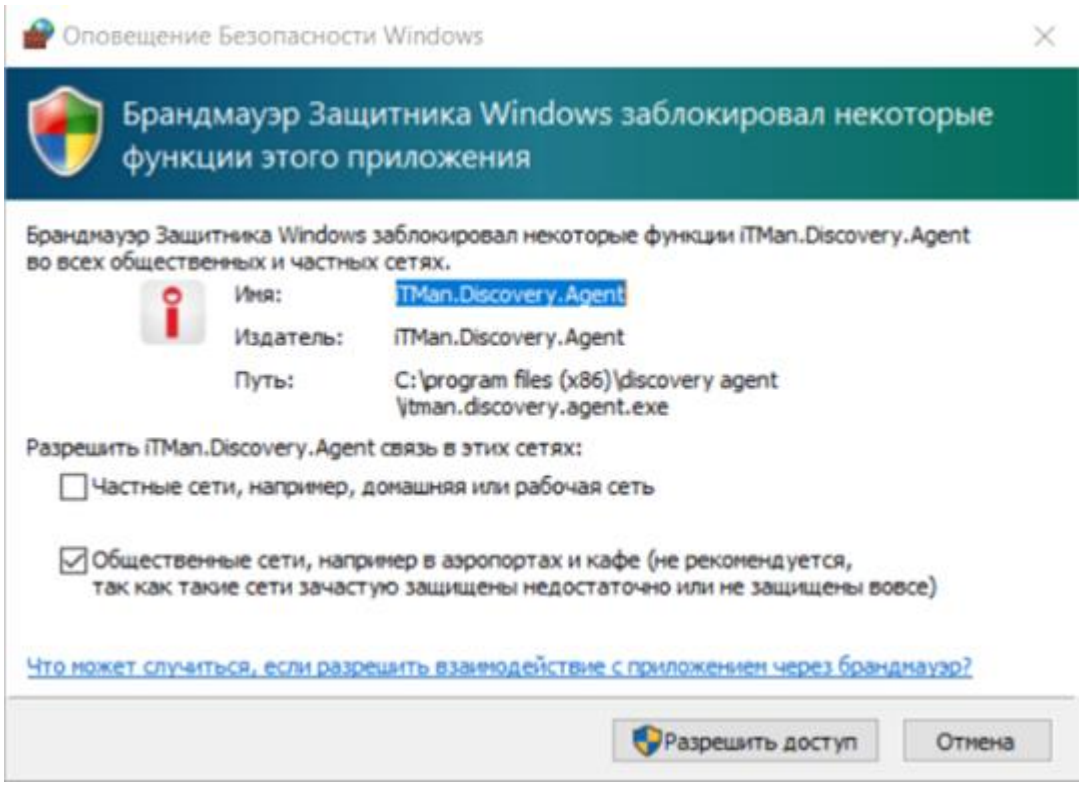
Завершение установки

После заполнения данных учетной записи:

- Откроется окно подтверждения установки.



- Нажмите **Установить**, чтобы запустить установку.
- Если после этого система запросит подтверждения предоставления доступа, то нажмите кнопку **Разрешить доступ**.



Важно

При сканировании Windows из-под windows агент в фоновом режиме добавит удаленный хост в доверенные хосты данного компьютера. Таким образом:

```
Set-Item wsman:\\localhost\\client\\TrustedHosts -Value
```

2.6.5. Тихая установка агента

Тихая установка - это установка в фоновом режиме без возможности интерактивно ввести параметры установки. Обычно это используется для пакетной установки, когда нужно поставить один и тот же комплект на несколько компьютеров.

Процедура установки в тихом режиме может быть выполнена несколькими способами:

- Из командной строки.
- Из диалогового окна Выполнить в Windows.
- С помощью CMD или BAT-файла.
- Через самораспаковывающийся архив с включенным в него конфигурационным файлом.

Формат команды установки:

```
msiexec /i "DiscoveryAgent_WindowsNative_N.NN.NN.msi" /qb!
```

```
COMMONDIR="C:\ProgramData\Discovery.Agent.V2" KEY="NNNNNNNN-NNNN-NNNN-
NNNN-NNNNNNNNNNNN" INTERACT_SERVER_ADDRESS="127.0.0.1"
INTERACT_PORT=5078 OPENSELFPORTSISENABLED=TRUE TCPSELFSTART=5088
TCPSELFEND=5089 TAG01="" TAG02="" TAG03="" USER_NAME="LocalSystem"
DOMAIN_NAME="." USER_PASSWORD=""
```

где:

- **/qb!** – режим тихой установки без интерфейса.
- **COMMONDIR** – рабочий каталог агента, укажите путь сохранения временных данных при инвентаризации.
- **KEY** – GUID ключ **вашей компании** (вида NNNNNNNN-NNNN-NNNN-NNNN-NNNNNNNNNNNN), который вы получили при покупке системы.
- **INTERACT_SERVER_ADDRESS** – IP адрес родительского элемента (агента или интеракта), который должен получать данные от настраиваемого агента. Если интеракт и агент установлены на одном компьютере, то укажите 127.0.0.1. Если агент устанавливается на VDI, то укажите адрес агента, на котором установлен VDI.
- **INTERACT_PORT** – порт родительского элемента (агента или интеракта), который должен получать данные от настраиваемого агента (по умолчанию для интеракта: 5078, для агента-родителя: 5088).
- **OPENSELFPORTSISENABLED** – включить/отключить открытие портов настраиваемого агента на прослушивание. Включите, если агент выполняет роль шлюза и должен получать данные от других агентов. Используется при настройке цепочки.
- **TCPSELFSTART, TCPSELFEND** – начальный и конечный порты настраиваемого агента, открытые на прослушивание.
  Используется, если включено открытие портов агента, то есть агент участвует в цепочке и может получать данные от других агентов. Проверьте, что указанные порты не заняты. Например, если сервер и агент установлены на одном компьютере, и вы укажите порт, который занят сервером, то агент не будет работать корректно.
  Если открытие портов выключено, то настроенные порты игнорируются, даже если заполнены.
- **TAG01, TAG02, TAG03** – специальные метки (теги) компьютера, на который устанавливается агент (если есть необходимость), например, VDI. В дальнейшем они используются при обогащении данных в процессе инвентаризации.

- **USER_NAME** – наименование учетной записи для запуска службы.
- **DOMAIN_NAME** – домен учетной записи для запуска службы.
- **USER_PASSWORD** – пароль учетной записи для запуска службы.

Если после установки агента, вы захотите изменить его параметры, то это можно сделать в файле конфигурации: config-agent.json

2.6.7. Настройка цепочки агентов

По умолчанию предполагается, что агенты передают собранные данные интеракту.

Вы можете настроить цепочку агентов, в которой агенты будут передавать данные по цепочке друг другу, а последний будет передавать данные интеракту.
Это используется, например, в случае если интеракт и агенты находятся в разных локальных сетях без доступа во внешнюю сеть.
Например, если в вашей компании несколько офисов, и в каждом из них только один компьютер имеет доступ во внешнюю сеть.

В таком случае:

- На компьютер, имеющий доступ к внешней сети, устанавливается агент, который выполняет роль шлюза.
- Все агенты внутри локальной сети отправляют данные на агент-шлюз.
- Далее агент-шлюз передает данные или интеракту или (как показано на схеме ниже), другому агенту-шлюзу, который в свою очередь уже передает их интеракту.

При установке агента, который выполняет роль шлюза, и получает данные от других агентов:

- Используйте тип агента **DiscoveryAgentNative.**
- Заполните данные родительского элемента - интеракта или другого агента-шлюза, который будет получать данные от настраиваемого агента:
    - IP адрес родительского элемента (по умолчанию для интеракта: 5078, для агента-родителя: 5088).
    - Порт родительского элемента.
- Включите **Открыть порты**.
- Укажите стартовый и конечный порт настраиваемого агента, открытые на прослушивание. На эти порты будут присылать данные другие агенты

При установке агента, который является конечным (собирает данные только с того устройства, на котором установлен, и не получает данные от других агентов):

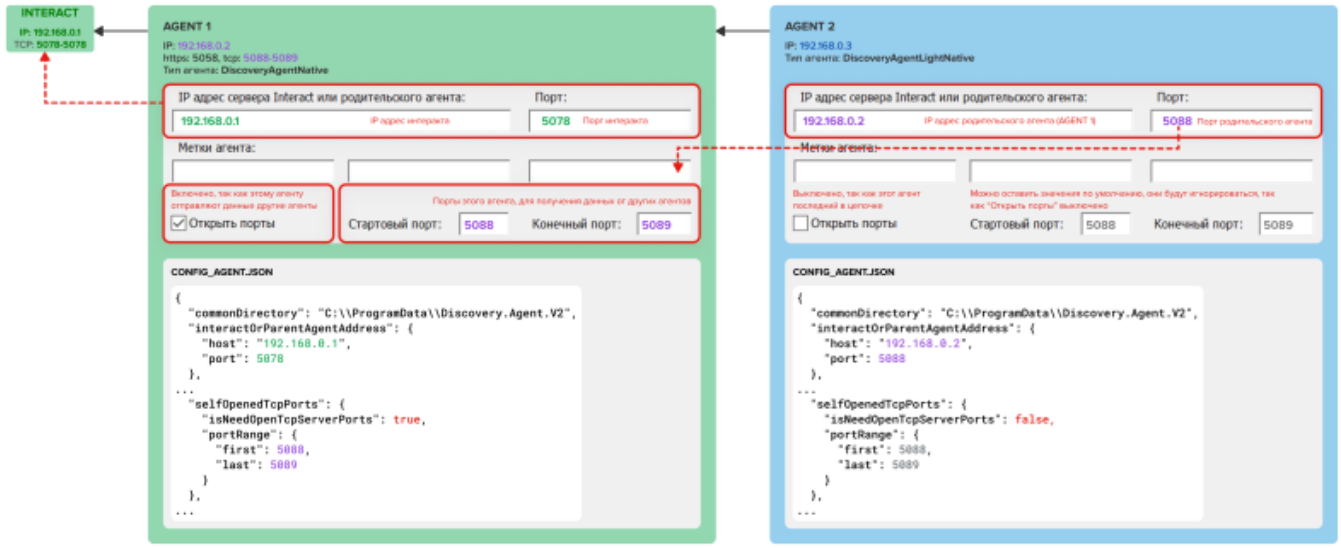
- Используйте тип агента **DiscoveryAgentLightNative.**
- Заполните данные родительского элемента - **агента-шлюза**, который будет получать

данные от настраиваемого агента:

- IP адрес родительского элемента.
- Порт родительского элемента (по умолчанию для агента-родителя: 5088).

- Выключите **Открыть порты**.
- Стартовый и конченый порт настраиваемого агента, на которые будут присылать данные другие агенты, можно оставить со значениями по умолчанию. Они будут игнорироваться.

Вы также можете отредактировать эти данные после установки в файлах конфигурации.



2.7. Настройка SSL сертификата для ОС Windows

Для обеспечения безопасности системы необходимо настроить работу компонентов системы по HTTPS протоколу. Для этого необходим SSL сертификат, выданный удостоверяющим центром.

Подключение SSL сертификата выполняется в процессе установки сервера и интеракта. Если вы выполнили соответствующие инструкции, то дополнительных действий не требуется.

В случае, если вы все сделали по инструкциям, но система не работает по HTTPS, - проверьте все этапы настройки, описанные далее.

2.7.1. Проверка основных настроек

Перед началом работы:

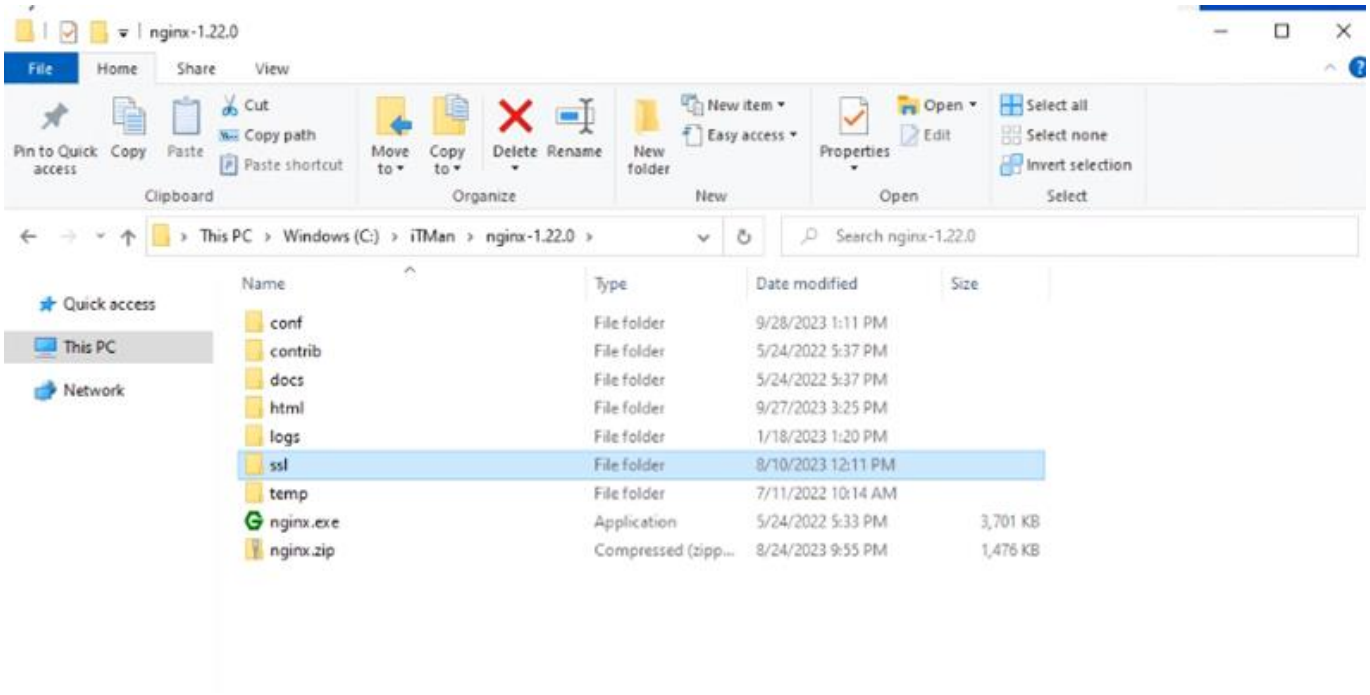
- Выпустите ключ и сертификат для SSL.

Обратите внимание на требования к SSL сертификату.

- Остановите **Discovery Server** и **Reader** в диспетчере задач, если они уже были установлены и запущены.

Далее:

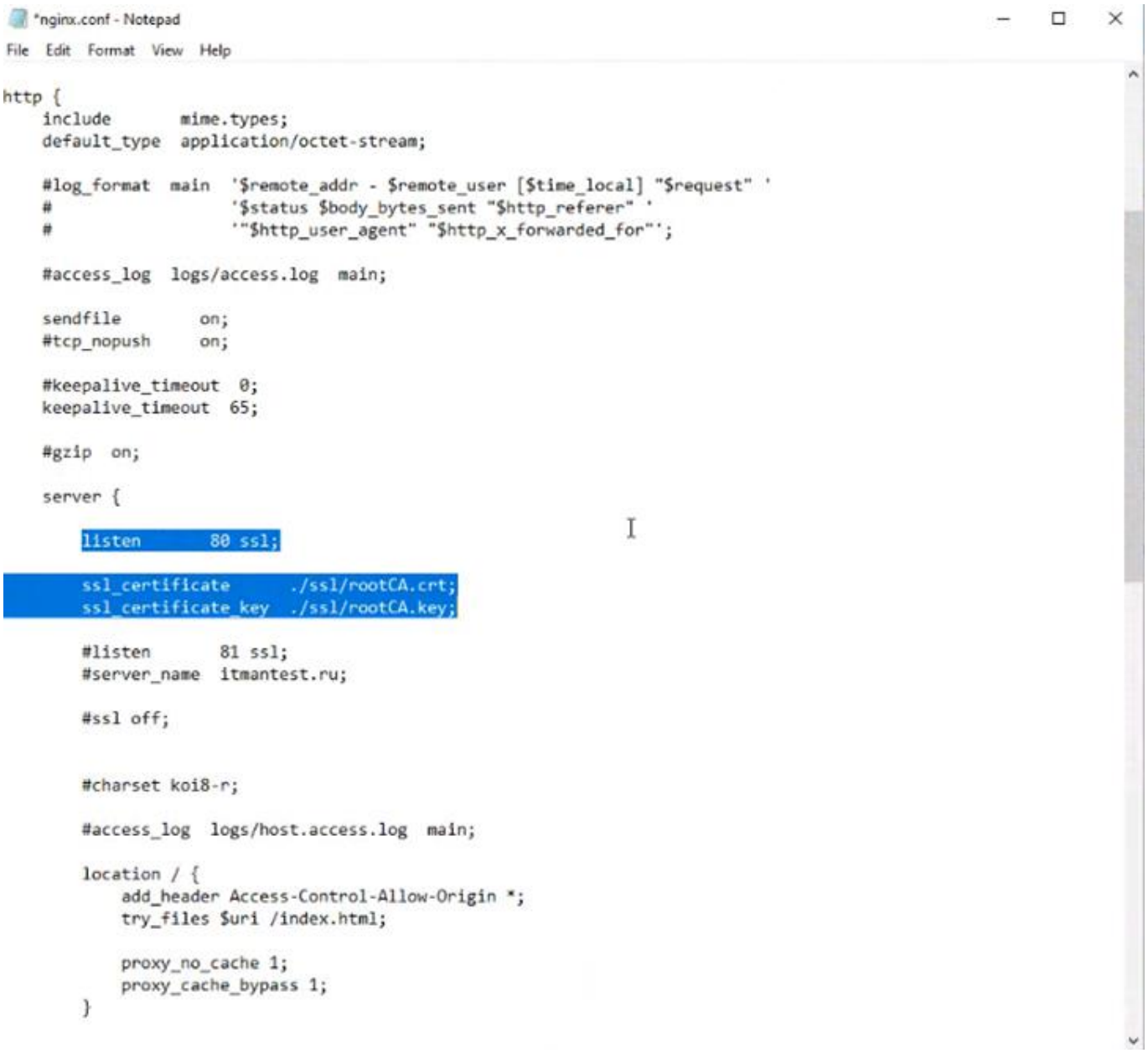
- Настройте отображение веб-сервиса.
  Для этого:
  - Откройте директорию, в которую установлен **nginx**.
  - Создайте в ней директорию с названием **SSL**.
  - Скопируйте в нее файлы приватного ключа сертификата.



- Перейдите в директорию **conf** в **nginx.**
- Откройте файл конфигурации **nginx.conf** на редактирование и добавьте в него данные:
  Обратите внимание, далее приведен пример - наименования файлов и расширения у вас будут свои.

```
server {
  listen 80ssl
  ssl_certificate ./ssl/rootCA.crt
  ssl_certificate_key ./ssl/rootCA.key
```

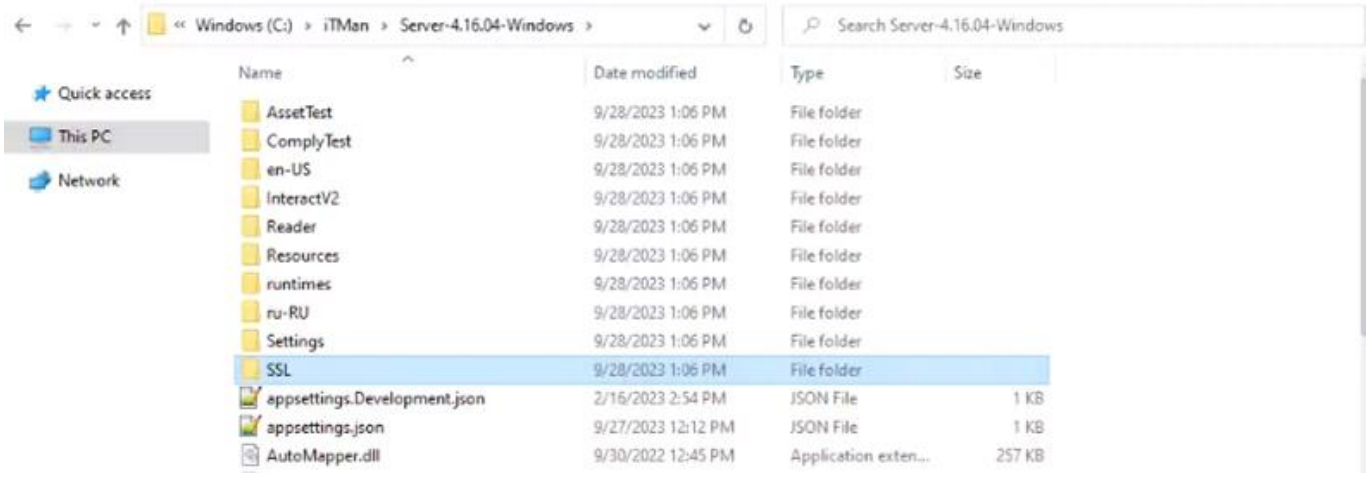
..



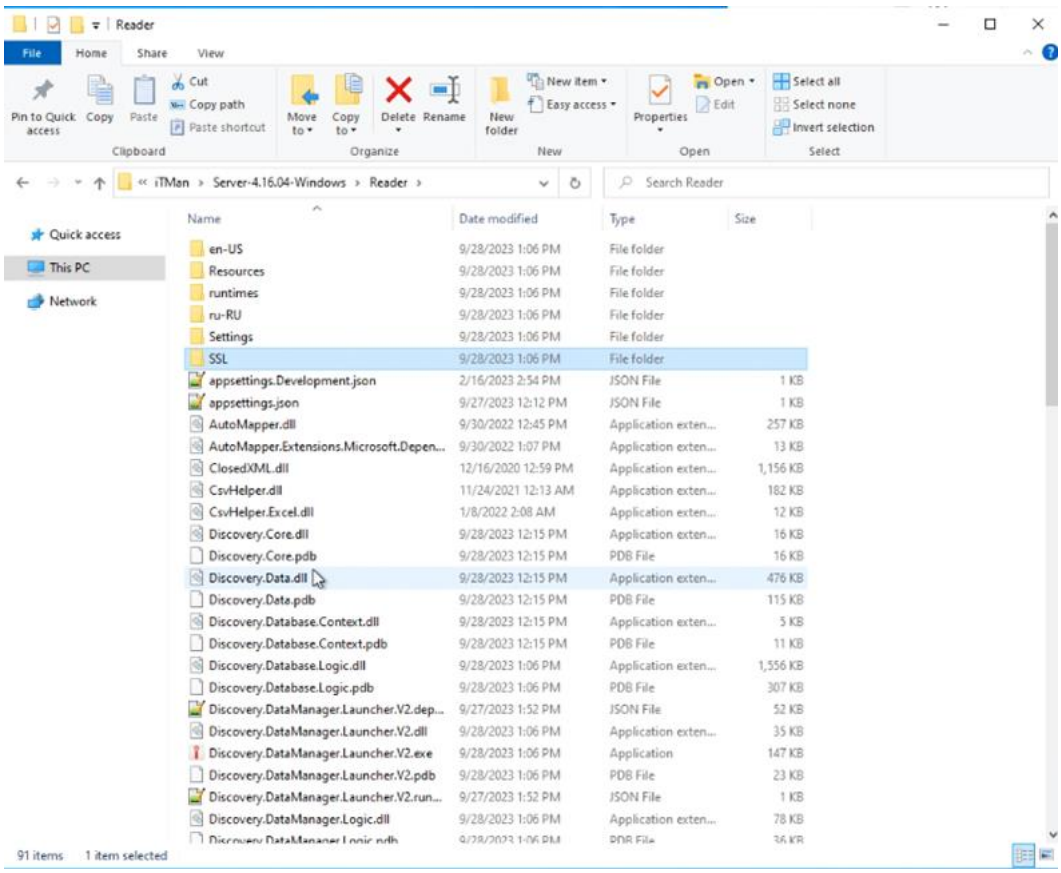
- Настройте передачу данных.

  Для этого:

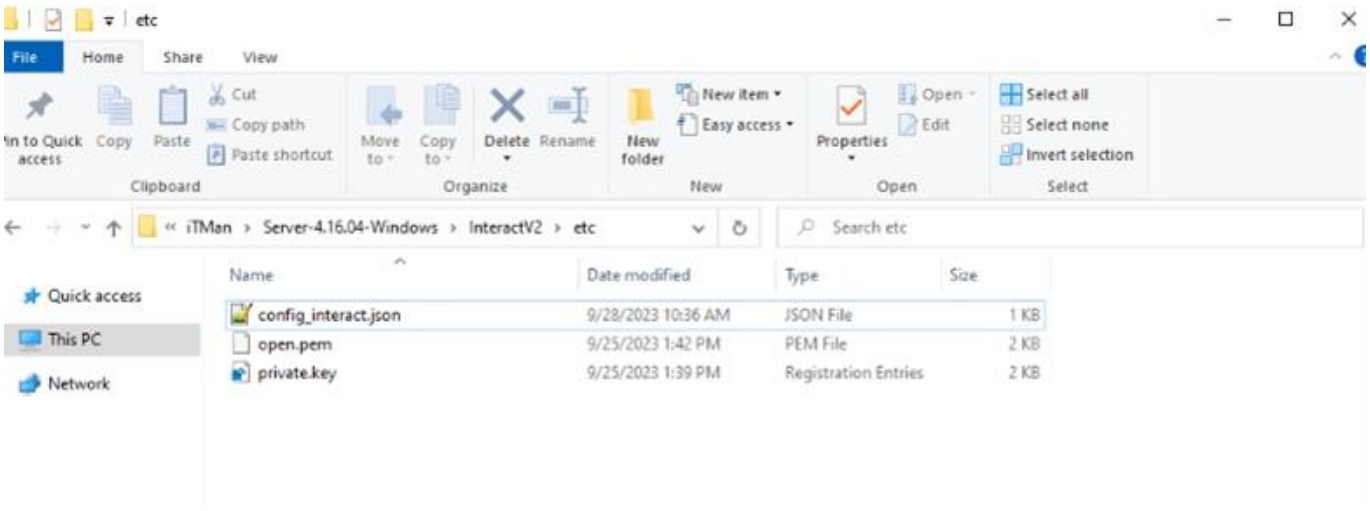
  - Откройте директорию сервера (та директория из которой вы будете запускать сервер).
  - Создайте в ней директорию с названием **SSL**, если это не было сделано ранее.
  - Скопируйте в нее файлы приватного ключа сертификата.



- Перейдите директорию **Reader** в общей директории сервера.
- Создайте в ней директорию с названием **SSL**, если это не было сделано ранее.
- Скопируйте в нее файлы приватного ключа сертификата.



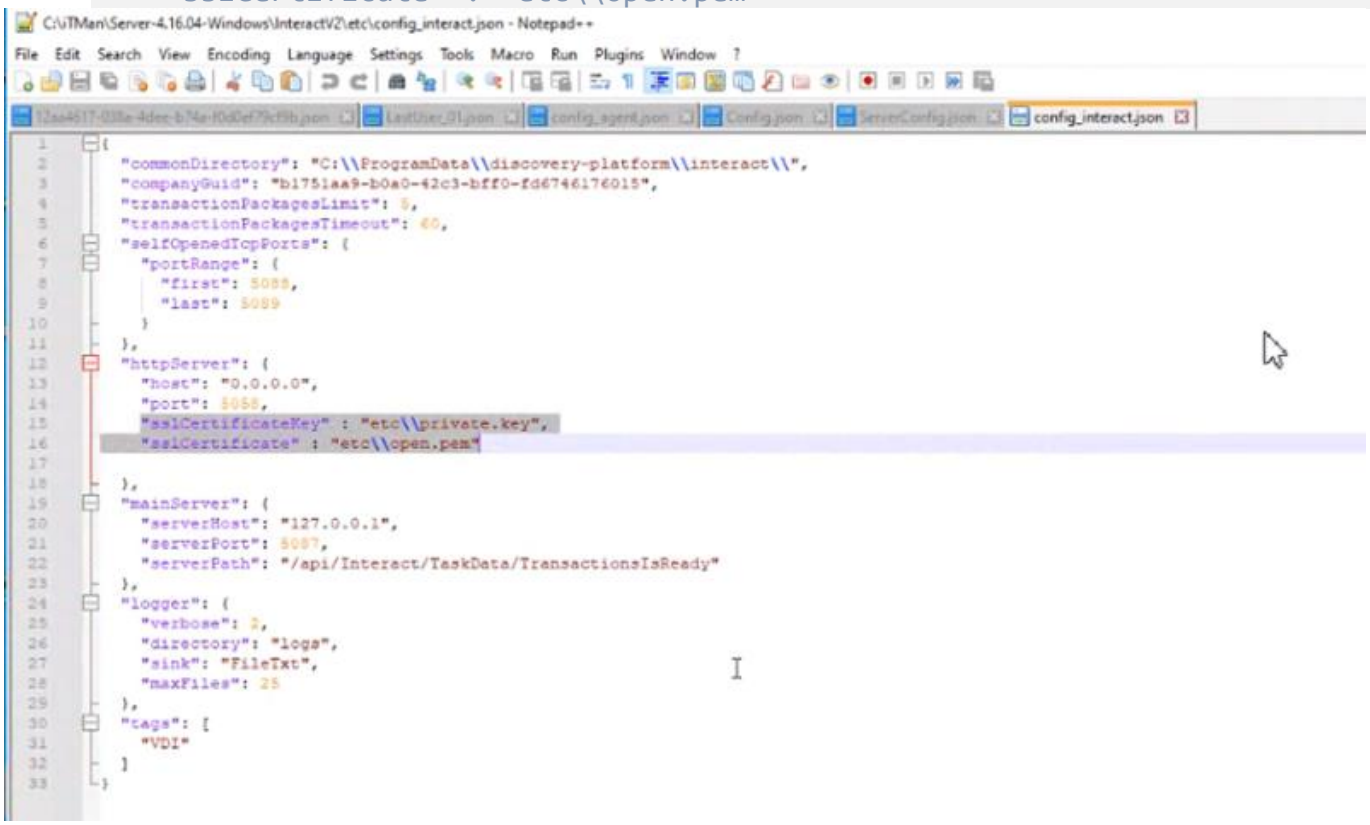
- Перейдите в директорию **Interact/etc** в общей директории сервера.
- Скопируйте в нее файлы приватного ключа сертификата.



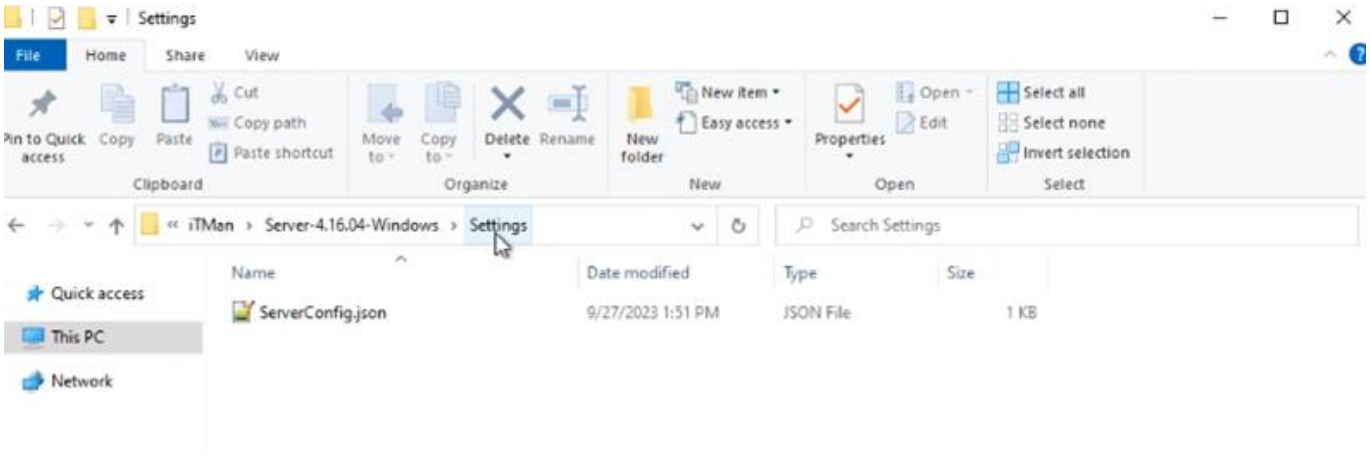
- Откройте файл **config_interact.json** от имени администратора.
- Добавьте в него информацию о сертификатах.

  Обратите внимание, далее приведен пример - наименования файлов и расширения у вас будут свои.

```
    "sslCertificateKey" : "etc\\private.key",
    "sslCertificate" : "etc\\open.pem"
```



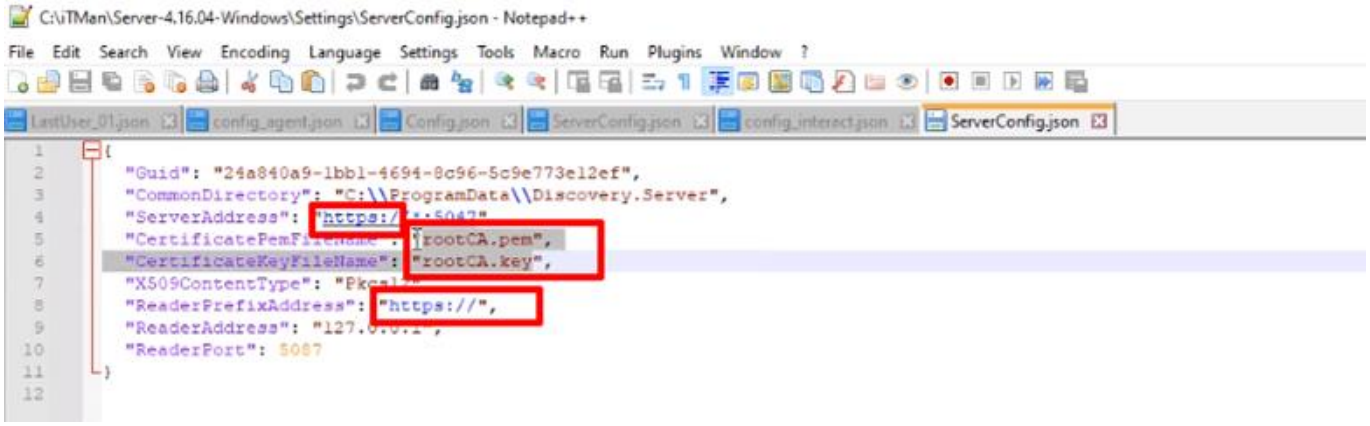
- Перейдите в директорию **Settings** в общей директории сервера.



- Откройте файл **ServerConfig.json** от имени администратора.
- Измените все значения **http** на **https** и добавьте следующие две строчки с названием ключа и сертификата.

  Обратите внимание, далее приведен пример - наименования файлов и расширения у вас будут свои.

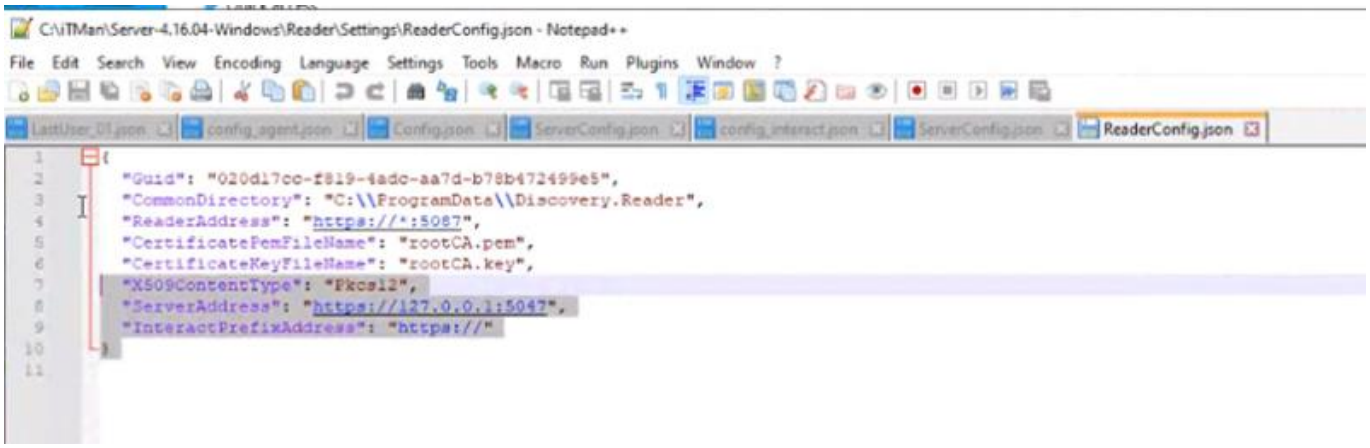
```
"CertificatePemFileName" : "rootCA.pem",
"CertificateKeyFileName " : "rootCA.key"
```



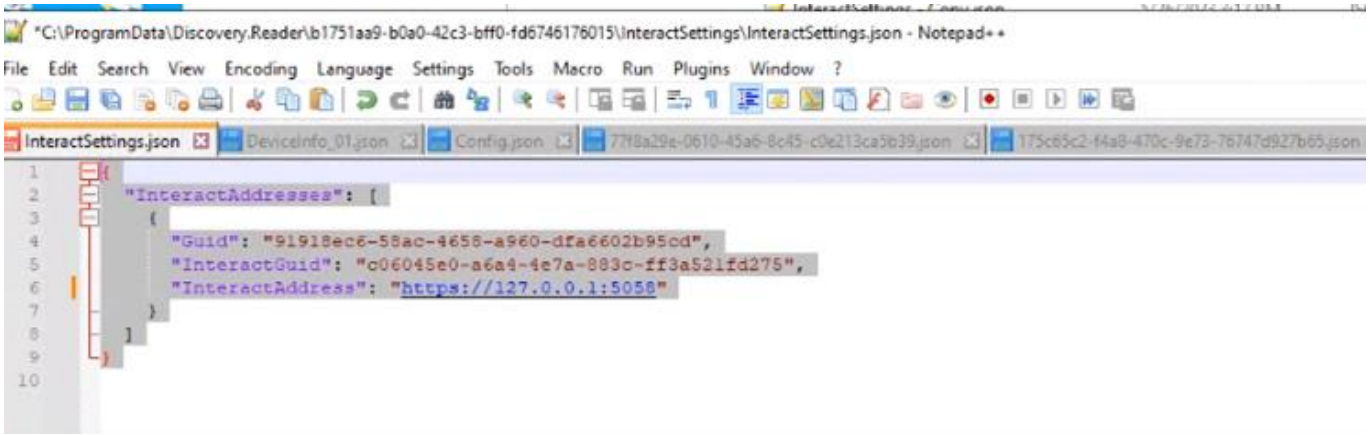
- Перейдите директорию **Reader/Settings** в общей директории сервера.
- Откройте файл **ReaderConfig.json** от имени администратора.
- Измените все значения **http** на **https** и добавьте следующие две строчки с названием ключа и сертификата.

  Обратите внимание, далее приведен пример - наименования файлов и расширения у вас будут свои.

```
"CertificatePemFileName" : "rootCA.pem",
"CertificateKeyFileName " : "rootCA.key"
```



- Если вы первый раз запускаете сервер, то в начале необходимо его запустить, затем остановить через диспетчер задач.
- Если у вас уже был запущен сервер, то этого делать не нужно.
  Вместо этого:
  - Перейдите в каталог C:\ProgramData\Discovery.Reader\b1751aa9-b0a0-42c3-bff0-fd6746176015\InteractSettings.
  - Откройте файл **InteractSettings.json**.
  - Измените значение **http** на **https**.
  - Сохраните файл.



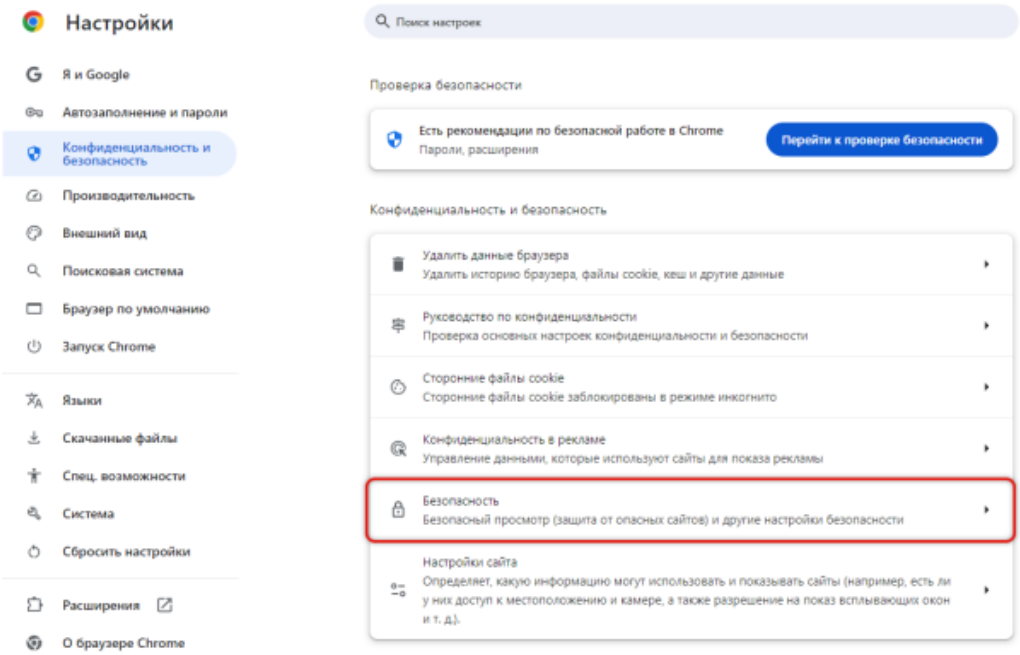
- Перезапустите сервер.

2.7.2. Дополнительная настройка для самоподписанных сертификатов

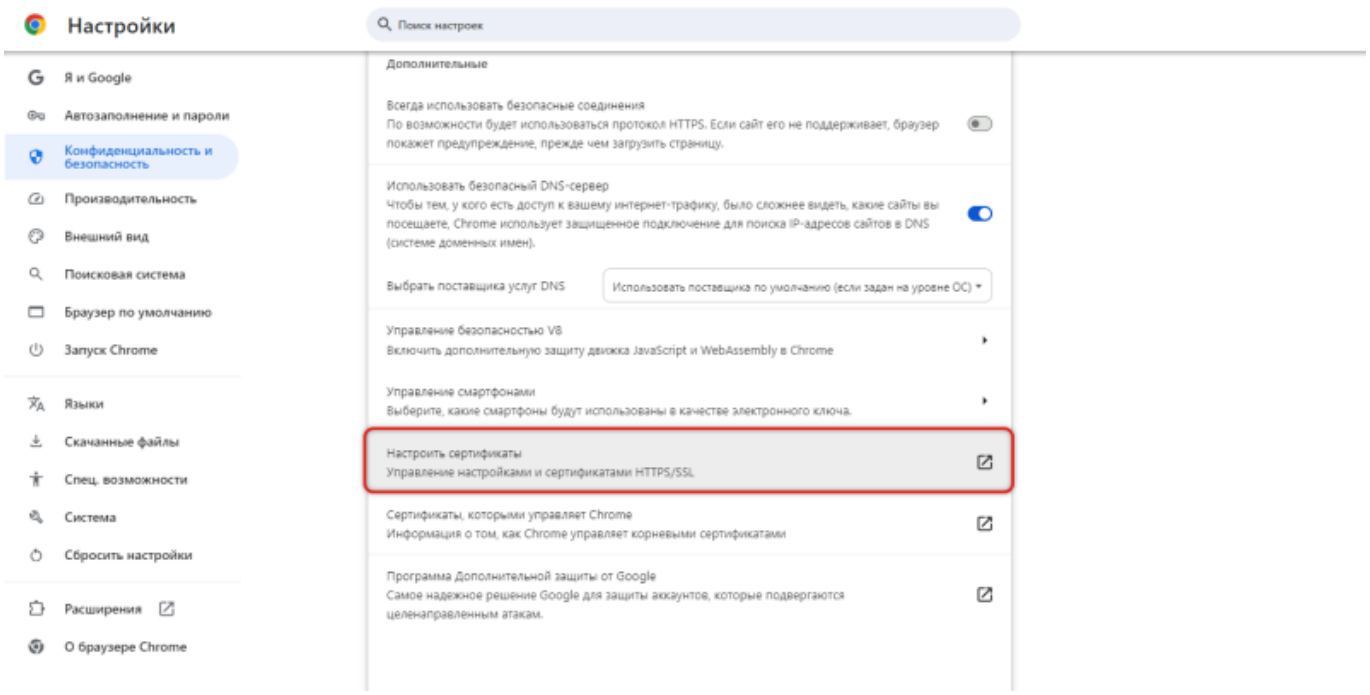
По требованиям безопасности SSL сертификат **должен быть выдан удостоверяющим центром (CA)**. Однако вы можете использовать самоподписанный сертификат, например, на этапе

тестирования. В этом случае следуйте инструкции ниже для добавления сертификата в доверенные в браузерах пользователей, работающих с системой.

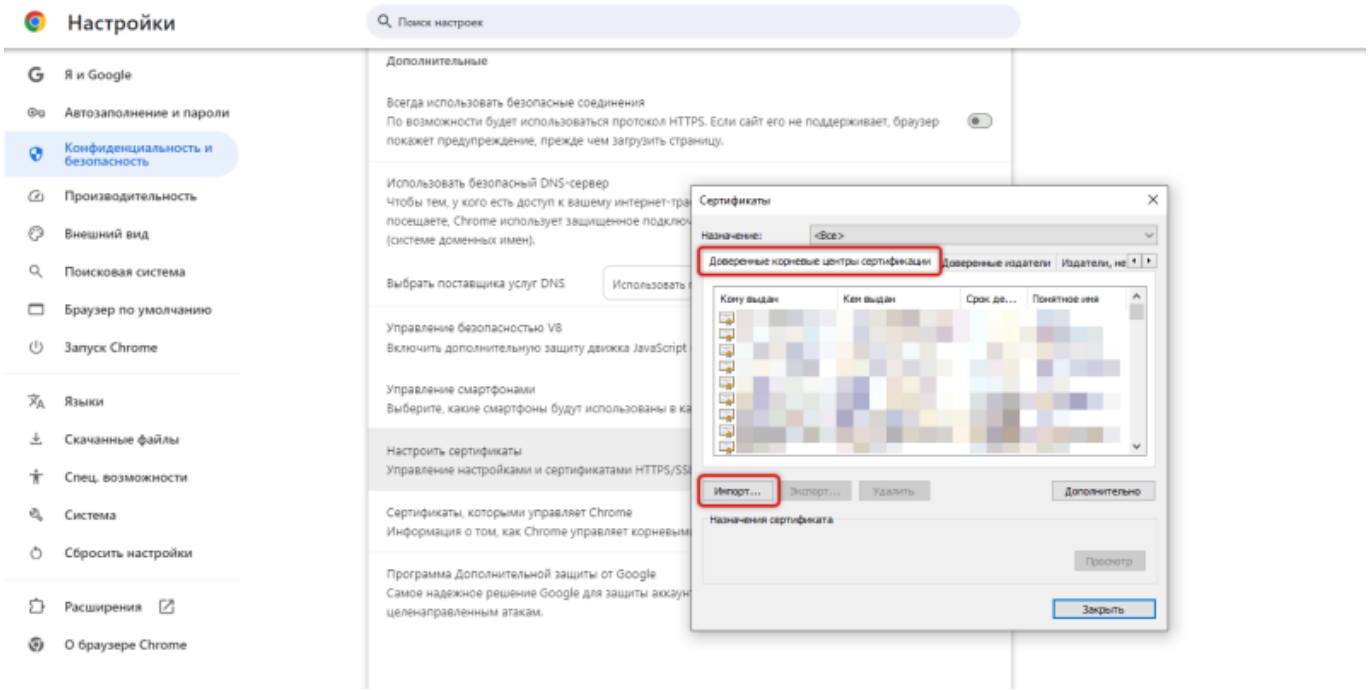
- Откройте браузер.
- Введите в адресную строку адрес сервера вместе с портом, например: **https://*.***.***.** /login/b1751aa9-b0a0-42c3-bff0-fd6746176015**
- Откройте настройки браузера.
- Перейдите на вкладку **Конфиденциальность и безопасность**.
- Выберите пункт **Безопасность**.



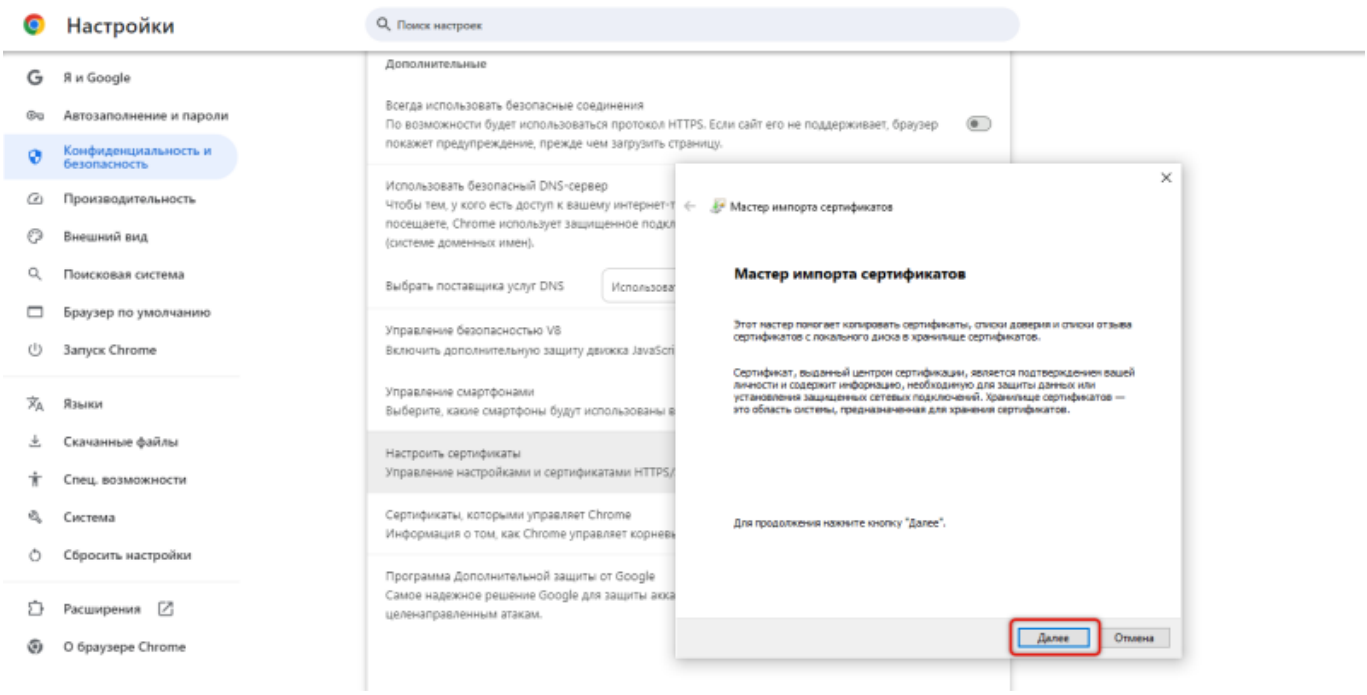
- Прокрутите страницу вниз и нажмите на пункт **Настроить сертификаты**.



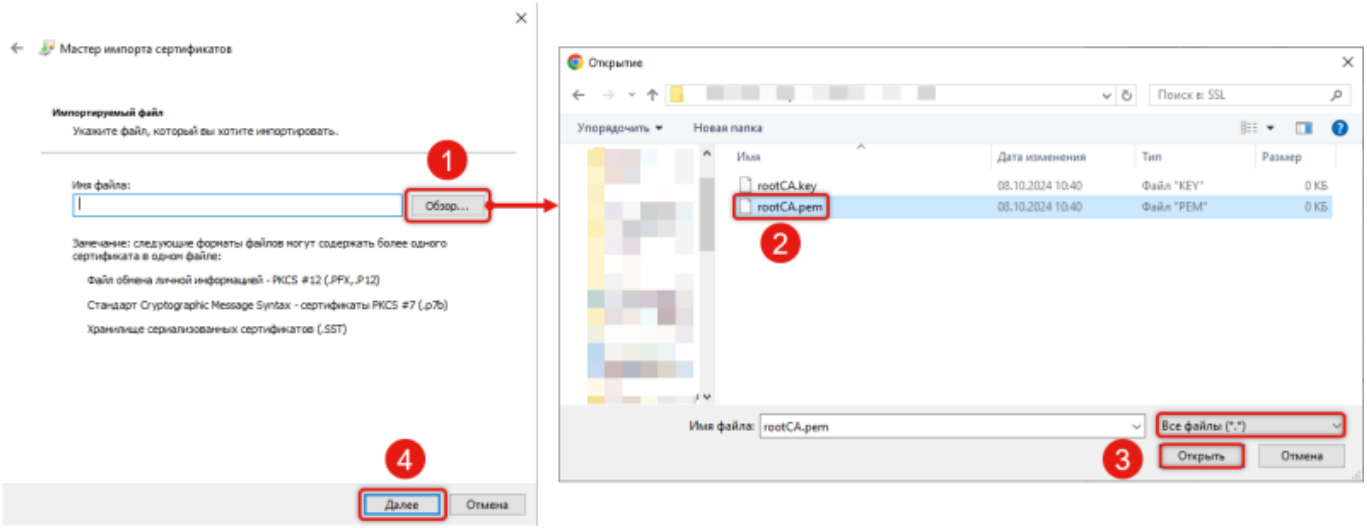
- Перейдите на вкладку **Доверенные корневые центры сертификации** и нажмите кнопку **Импорт**.



- Откроется мастер импорта сертификатов.
- Нажмите кнопку **Далее**.



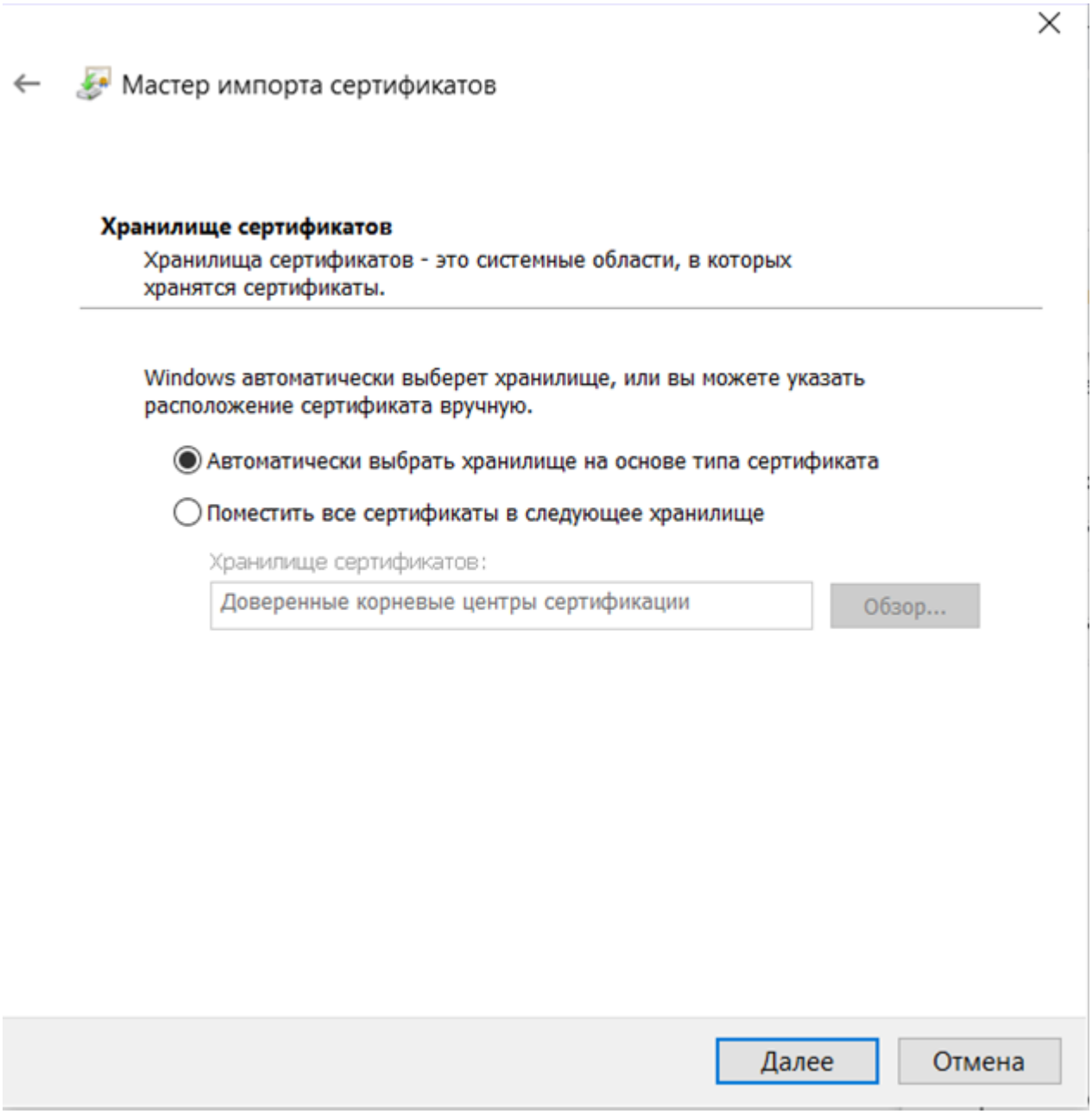
- Выберите импортируемый файл:
    - Нажмите кнопку **Обзор**.
    - Выберите файл сертификата на своем устройстве и нажмите кнопку **Открыть**. Если файл сертификата не отображается в директории, то включите отображение **Все файлы (*.*)**.
      Обратите внимание, далее приведен пример - наименования файлов и расширения у вас будут свои.
- Нажмите кнопку **Далее** в мастере импорта.



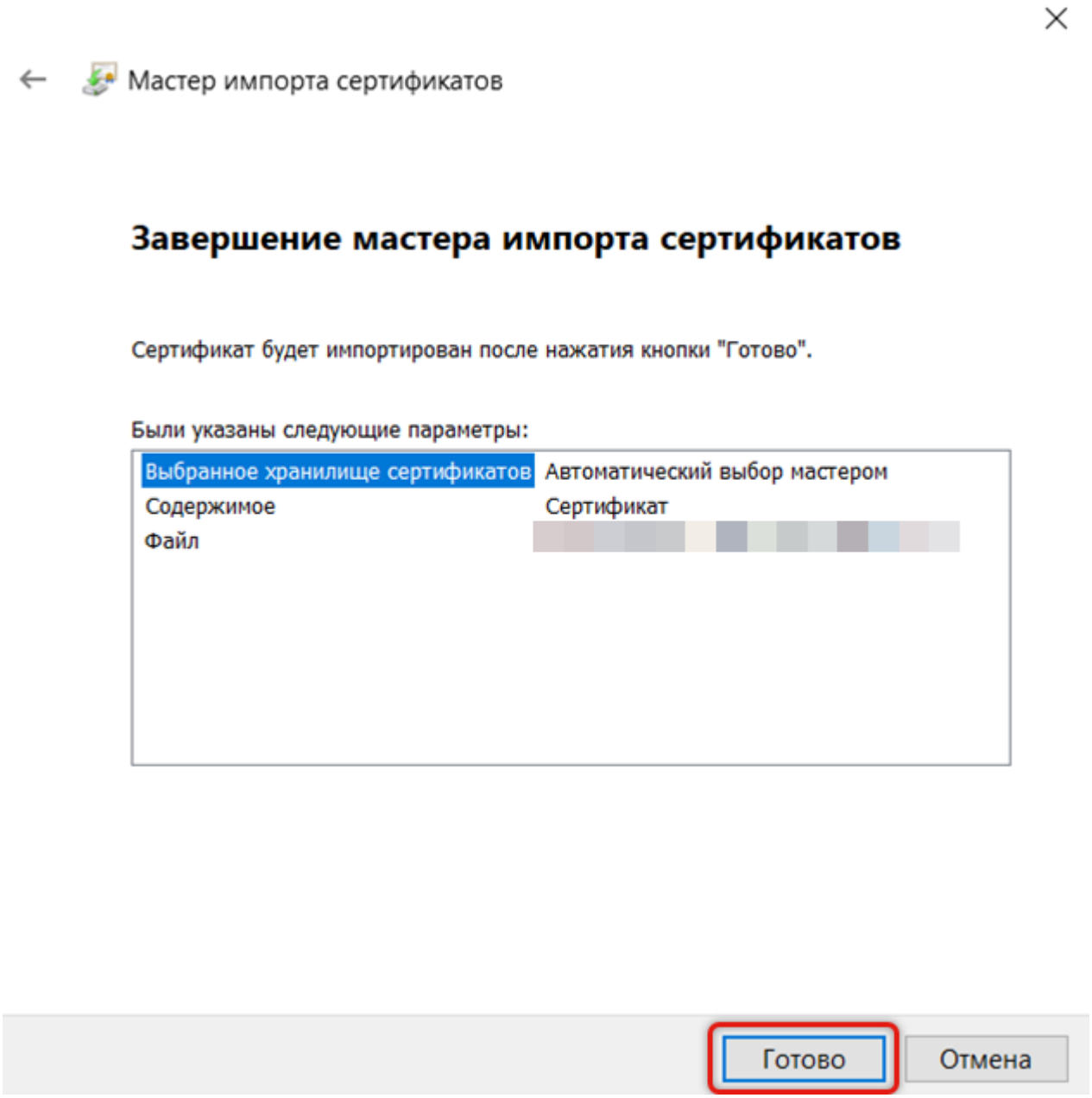
- Выберите хранилище сертификатов.

  Если политика вашей компании позволяет, то можно выбрать **Автоматический выбор хранилища**.

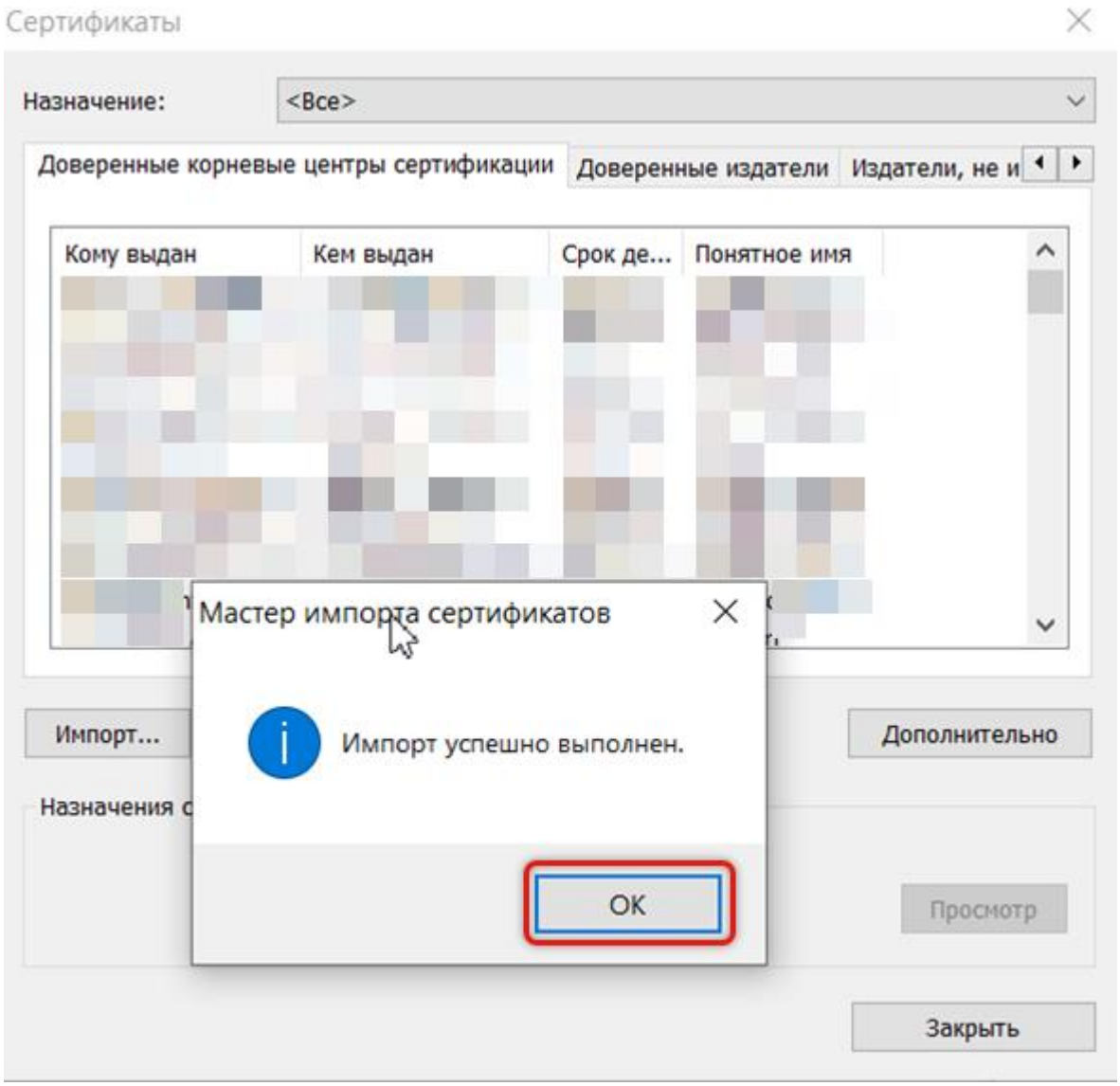
- Нажмите кнопку **Далее**.



- Проверьте правильность настройки.
- Если все верно, то нажмите кнопку **Готово**.



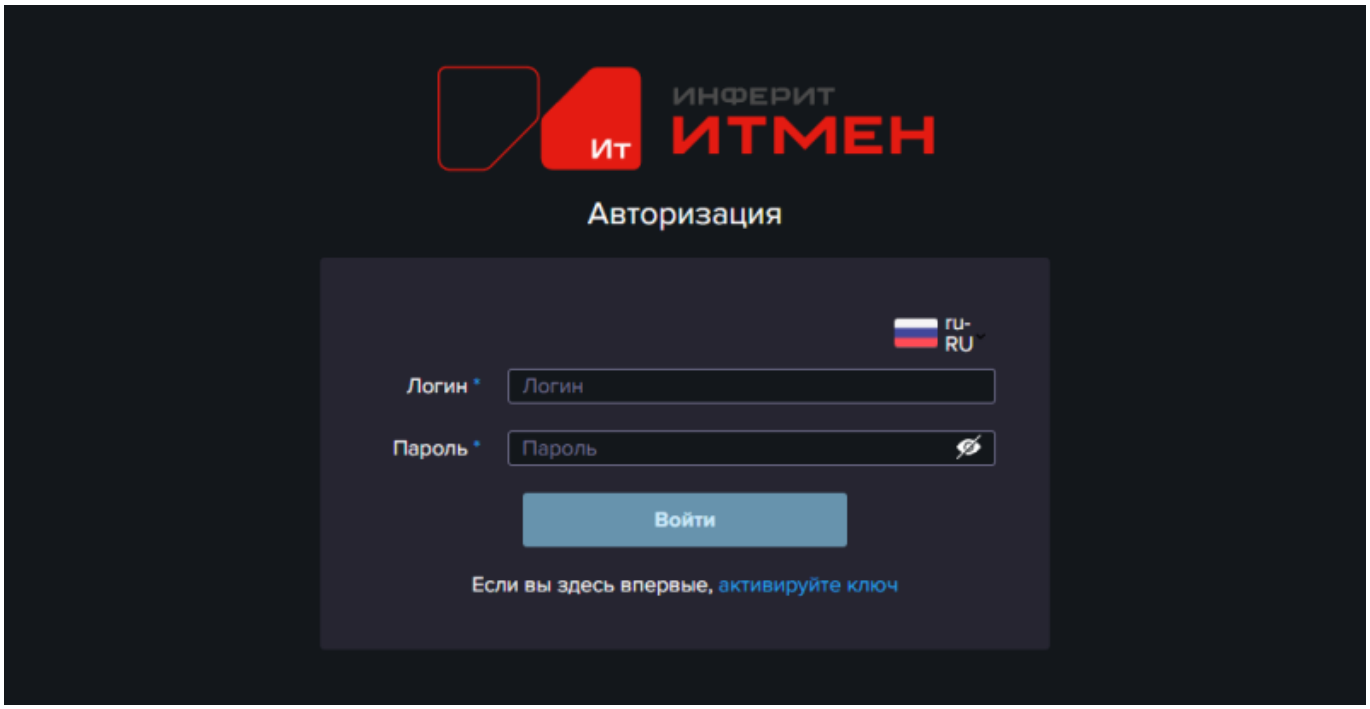
- Окно мастера установки закроется и отобразится сообщение об успешной установке сертификата.
- Нажмите **ОК** в сообщении.



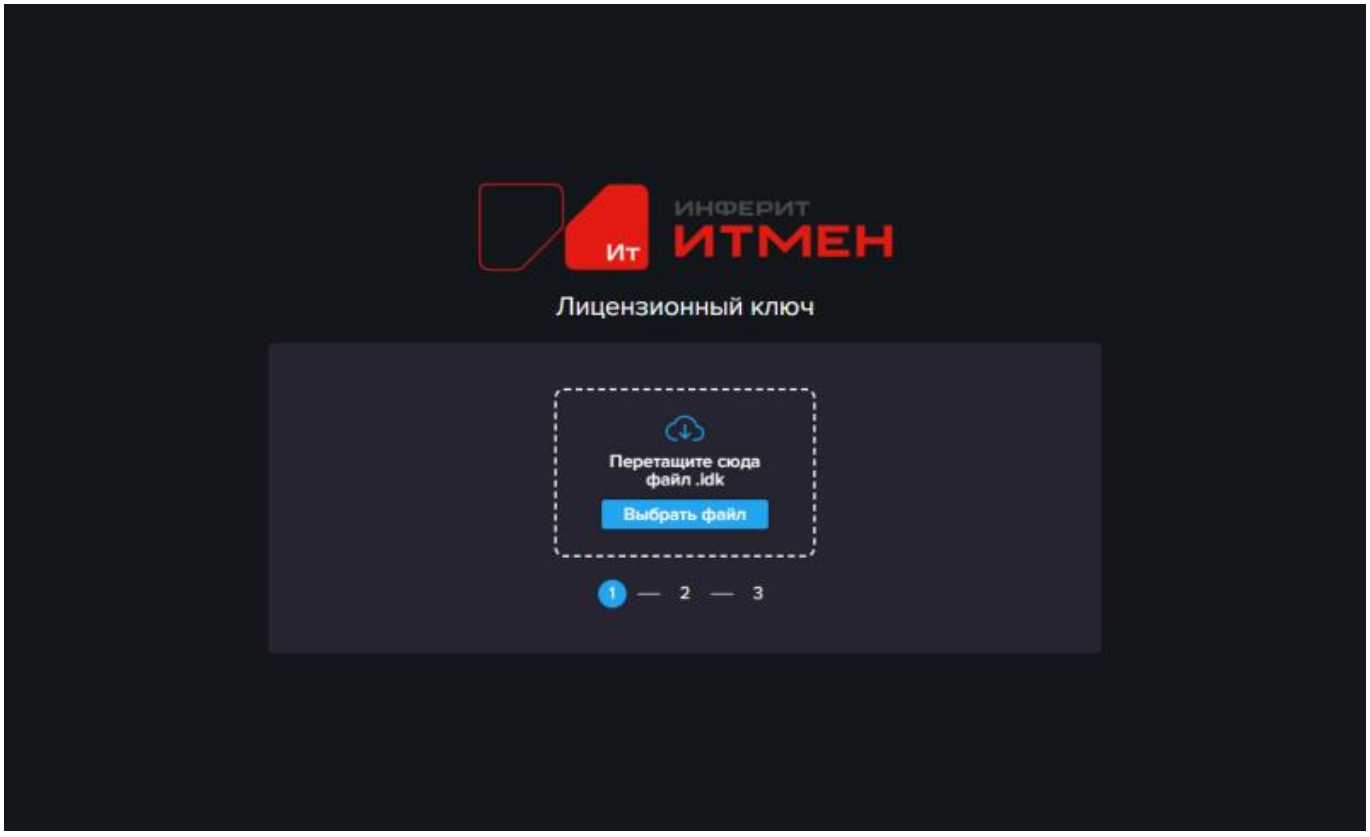
2.8. Запуск сервера на ОС Windows

После того, как сервер установлен:

- Запустите сервер, для этого в директории, куда он был установлен, запустите файл: **Discovery.Server**.
- Откройте браузер и введите публичный адрес вашего компьютера, на который вы установили сервер.
- Откроется окно авторизации.

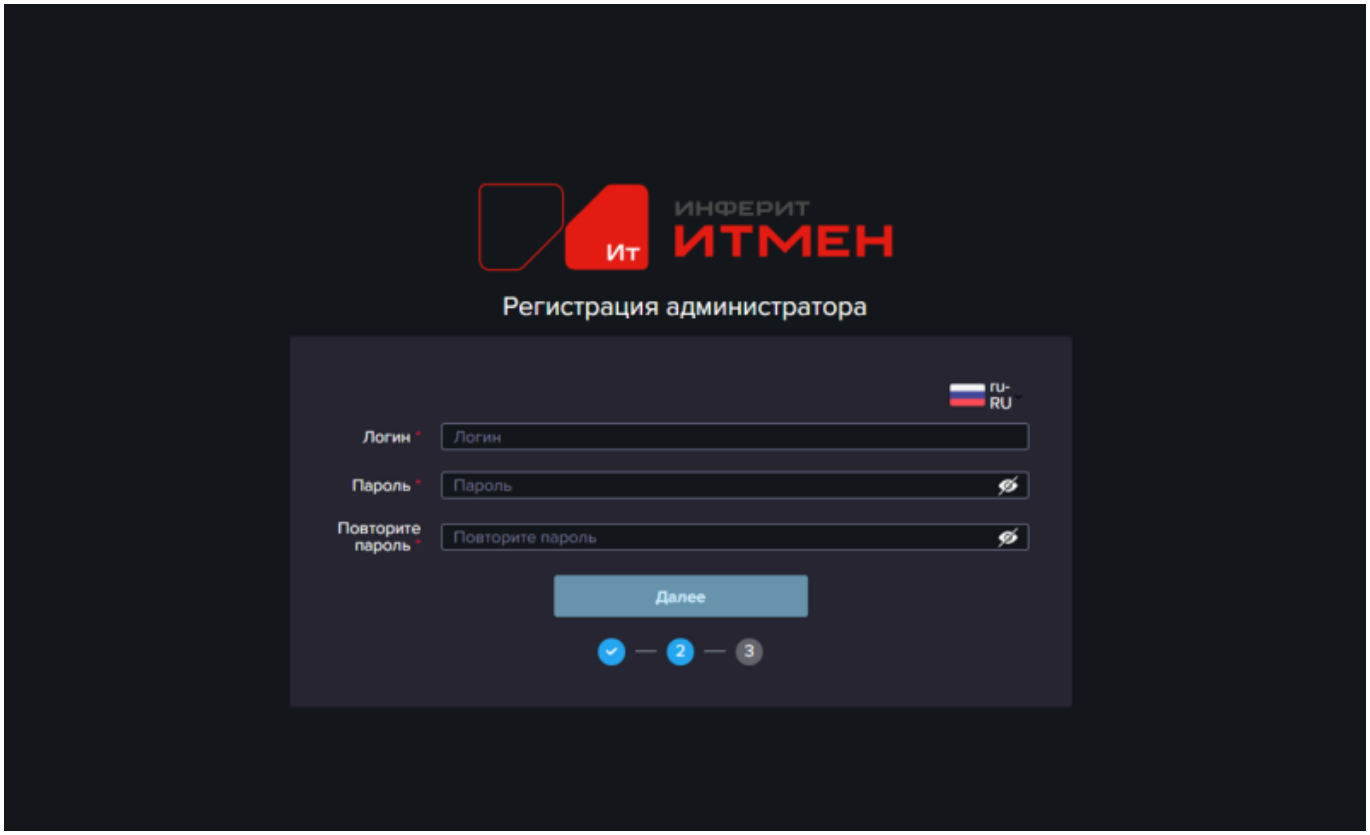


- Нажмите ссылку **активируйте ключ**.
- Откроется окно ввода ключа.

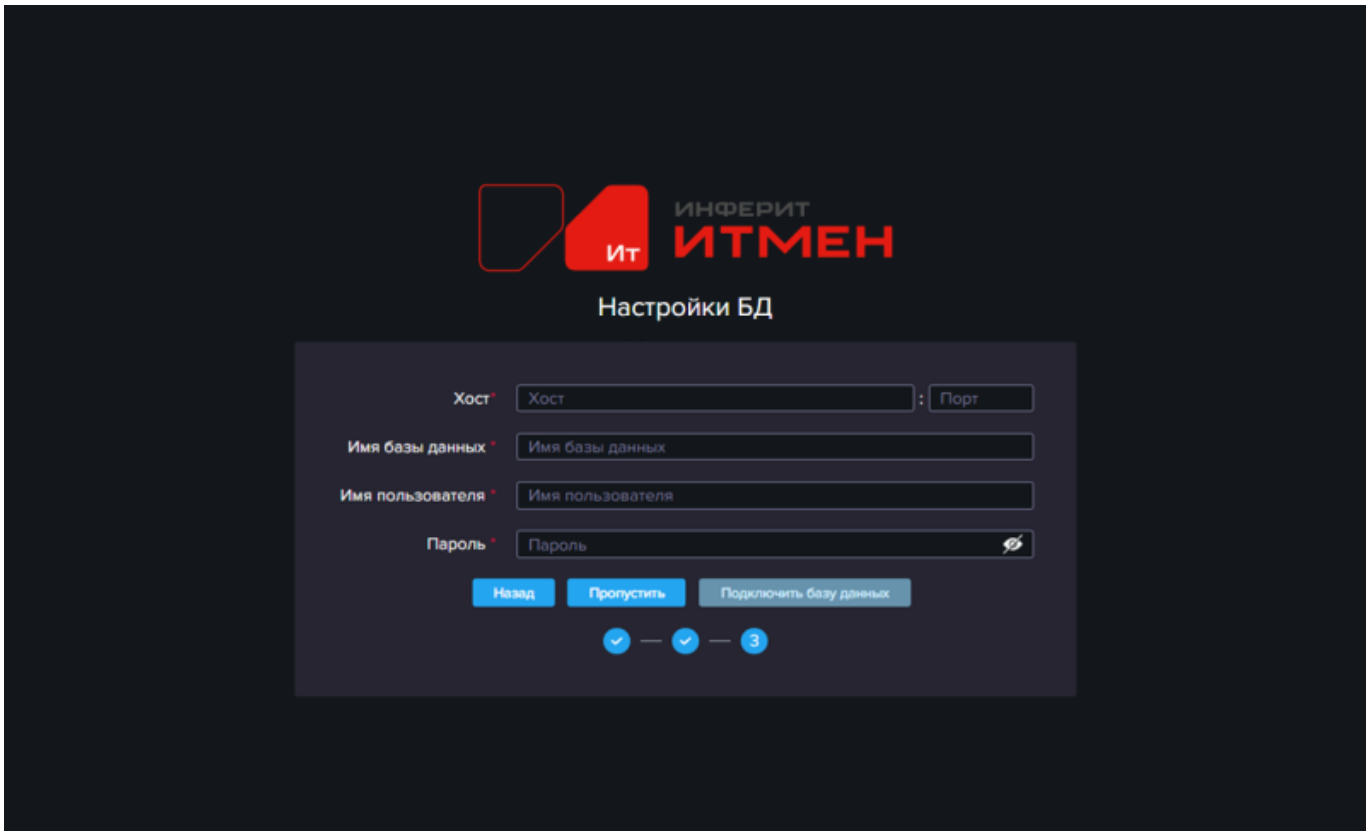


- Загрузите файл с ключом, который вы получили вместе с системой.

- Откроется окно регистрации администратора.



- Зарегистрируйте учетную запись администратора сервера, для этого:
    - Введите логин и пароль. Не забудьте сохранить авторизационные данные в надежном месте.
    - Нажмите кнопку **Далее**.
- Откроется окно для ввода данных СУБД, в которую в дальнейшем будут сохраняться данные.



- Заполните данные СУБД и нажмите кнопку **Подключить базу данных**.
  Вы можете пропустить этот шаг и ввести данные позднее. В этом случае нажмите кнопку **Пропустить**.
- После прохождения авторизации вам откроется окно первого входа, где необходимо нажать кнопку **Далее** и перейти в систему.

3. Установка на Linux

3.1. Порядок действий при установке в ОС Linux

1. Определите компьютер, на которой будете устанавливать сервер, там будут храниться данные сервера и инвентаризации.
   Настройте к нему доступ по статическому IP адресу.
2. Организуйте хранение данных:
   - Установите БД **Postgre**
   - Создайте в БД пользователя с правами **суперадмин**, логин и пароль этого

пользователя потребуются при установке системы.

3. Установите **Net core 7.0** на компьтеры, где будет установлен сервер и агенты.
4. Выпустите **SSL сертификат**, см. требования к сертификату.
5. Установите nginx на компьютер, где будет установлен сервер, и выполните настройку.
6. Установите Discovery Server
7. Установите Discovery Interact
8. Установите агентов
9. Запустите Discovery Server
10. Проверьте работу

3.2. Дистрибутив

Для установки системы на Linux вам потребуется:

1. Ключ
2. Пакет приложения в zip, который содержит:
   - **DiscoveryServer_***-Linux.zip** или **.deb**
   - **DiscoveryWeb_***.zip** или **.msi**
   - interct.deb
   - agent.deb

3.3. Установка PostgreSQL на ОС Linux

Для установки выполните команды:

```
$ sudo apt update
$ sudo apt-get install postgresql-12
```

Далее запустите сервис и проверьте его статус с помощью команд:

```
$ sudo systemctl start postgresql.service
$ sudo systemctl status postgresql.service
```

При установке PostgreSQL по умолчанию создается пользователь postgres, но вы можете создать своего пользователя. Обратите внимание, у пользователя должны быть права суперадмина.

Это можно сделать с помощью команд:

```
$ sudo -u postgres createuser -s -i -d -r -l -w superadmin
$ sudo -u postgres psql -c "ALTER ROLE <> WITH PASSWORD '<>';"
sudo -u postgres createuser
```

3.4. Установка и настройка nginx на ОС Linux

- Установите **nginx** на компьтер, где будет установлен сервер, для этого выполните стандартные действия:
    - Обновите локальный индекс пакетов командой:

```
$ sudo apt update
```

    - Установите nginx командой:

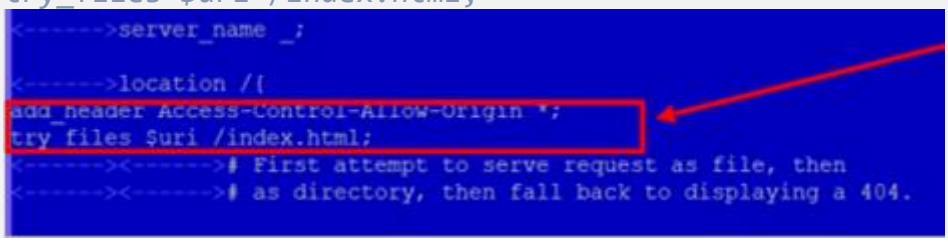
```
$ sudo apt install nginx
```

- Скопируйте директорую **html** из предоставленного дистрибутива **DiscoveryWeb_***.zip** в директорию установленного nginx **/var/www/html**.

  Например, с помощью WinCSP.

- Внесите изменения в настройки **nginx** в файле **sites-available**:

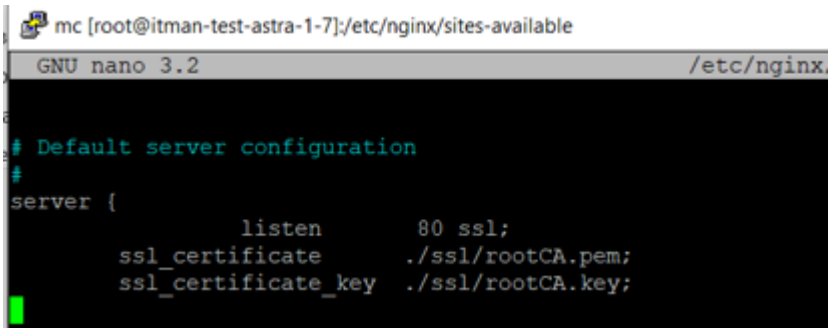
```
add_header Access-Control-Allow-Origin *;
try_files $uri /index.html;
```



- Создайте директорию для хранения SSL сертификата и перенесите в нее сертификат и ключ.

- Добавьте в файле **sites-available** данные о SSL сертификате:

```
server {
  listen       80 ssl;
  ssl_certificate      ./ssl/rootCA.pem;
  ssl_certificate_key  ./ssl/rootCA.key;
```



- Если у вас будут свои разрешенные порты, то вы их можете изменить их в этом файле **/www/html/config.json.**

3.5. Установка Discovery Server на ОС Linux

3.5.1. Установка приложения из deb пакета

- Остановите nginx с помощью команды:

```
$ sudo systemctl stop nginx
```

- Перенесите директорию **html** из дистрибутива **DiscoveryWeb_*** ** в директорию **/var/www** (полностью заменяя существующую там папку **html**).
- Выполните следующую команду для начала установки дистрибутива (возможности конфигурации установки описаны далее):

```
sudo dpkg -i DiscoveryServer_LinuxNative_***.deb
```

При установке пакета будет запущено три сервиса systemd, статус которых можно проверить следующими командами:

```
sudo systemctl status discovery-reader.service
sudo systemctl status discovery-writer.service
sudo systemctl status discovery-server.service
```

Запуск/ перезапуск/ остановка также производится через стандартные команды systemd.

Конфигурация установки

Вы можете сконфигурировать установку двумя способами:

- С помощью интерактивного меню, которое доступно с помощью команды:

```
sudo dpkg-reconfigure discovery-server
```

- С помощью вспомогательного скрипта конфигурирования:

```
./deb/usr/local/bin/configure-all.sh -s /var/tmp/discovery-server -d
/var/tmp/discovery-reader -p 5047 -o 5087 -r 5097 -v VDI_
```

Скрипт может быть также применен в одну строчку с командой установки:

```
sudo dpkg -i DiscoveryServer_LinuxNative_***.deb &&  sudo
./deb/usr/local/bin/configure-all.sh -s /var/tmp/discovery-server -d
/var/tmp/discovery-reader -p 5047 -o 5087 -r 5097 -v VDI_
```

Параметры командной строки имеют формат:

-s | --server_directory	**Директория временных файлов для сервера (по-умолчанию /va**
-d | --reader_writer_directory	Директория временных файлов для модулей Reader и Writer (по-ум /var/tmp/discovery-reader).
-p | --server_port	Порт который открывает сервер (по-умолчанию 5047).
-o | --reader_port	Порт который открывает Reader (по-умолчанию 5087).
-r | --writer_port	Порт который открывает Writer (по-умолчанию 5097).
-v | --vdi_prefix	Vdi префикс (by default VDI_).
-h | --help	Показать данное сообщение и выйти.

Если в дальнейшем вы хотите изменить разрешенные порты на свои, то это можно сделать в файле **/var/www/html/config.json**.

```
{
  "BASE_API_PORT": "5047",
  "READER_API_PORT": "5087",
  "WRITER_API_PORT": "5097",
  "DEFAULT_INSTACE": "BaseInstance",
  "FLAG":""
}
```

Чтобы проверить результат, откройте файлы:

- **/opt/Discovery.Server/Settings/ServerConfig.json**
- **/opt/Discovery.Reader/Settings/ReaderConfig.json**

Ошибки установки

Важно! Если установка не запустилась необходимо проверить следующие данные:

- Наличие **aspnetcore-runtime-7.0**

  Для установки выполните команды:

```
// Добавляем репазиторий майкрософт в sorcelist.
wget https://packages.microsoft.com/config/ubuntu/20.04/packages-
microsoft-prod.deb -O packages-microsoft-prod.deb
```

```
sudo dpkg -i packages-microsoft-prod.deb
rm packages-microsoft-prod.deb
// Обновляем список пакетов в репазиториях
sudo apt-get update
// устанавливаем дотнет.
sudo apt-get install aspnetcore-runtime-7.0
```

- Версия **libssl** не соответствует (более новые версии linux могут не содержать libssl нужной (более старой) версии, необходимой для aspnetcore).

  Для установки выполните команды:

```
wget
http://archive.ubuntu.com/ubuntu/pool/main/o/openssl/libssl1.1_1.1.1-
1ubuntu2.1~18.04.23_amd64.deb
sudo dpkg -i libssl1.1_1.1.1-1ubuntu2.1~18.04.23_amd64.deb
rm libssl1.1_1.1.1-1ubuntu2.1~18.04.23_amd64.deb
```

- Добавьте в файл данные:

```
add_header Access-Control-Allow-Origin *;
try_files $uri /index.html;
```

3.5.2. Установка приложения из архива

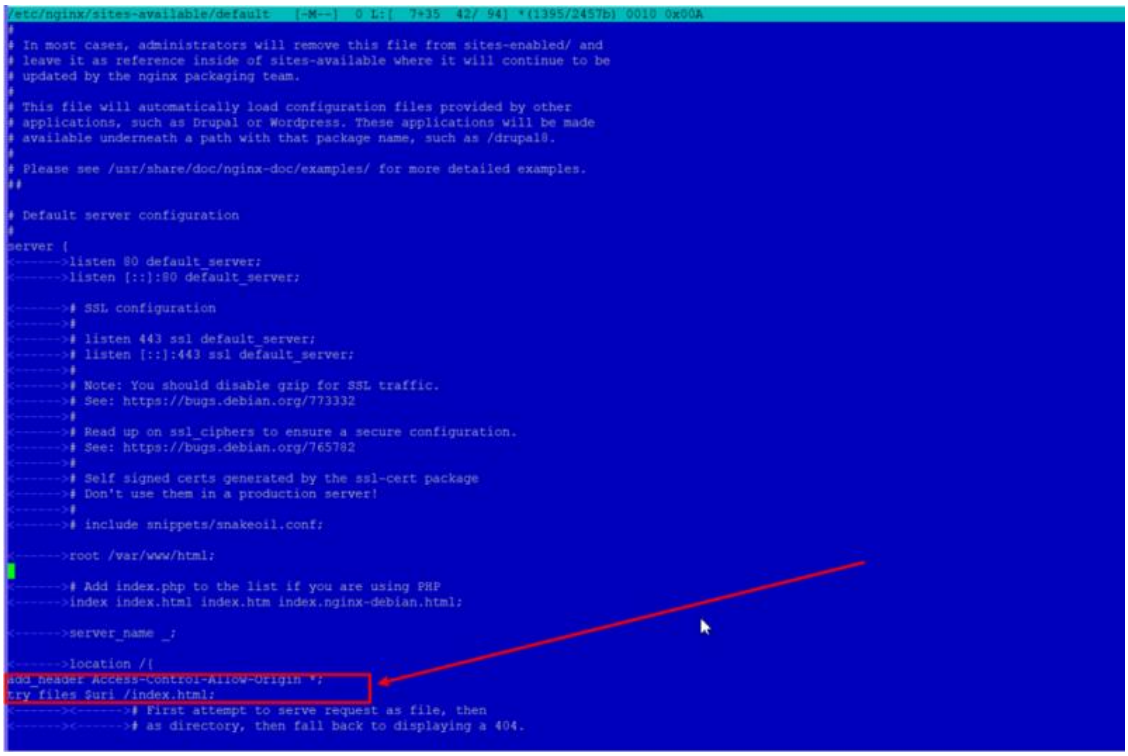
- Остановите nginx с помощью команды:

```
$ sudo systemctl stop nginx
```

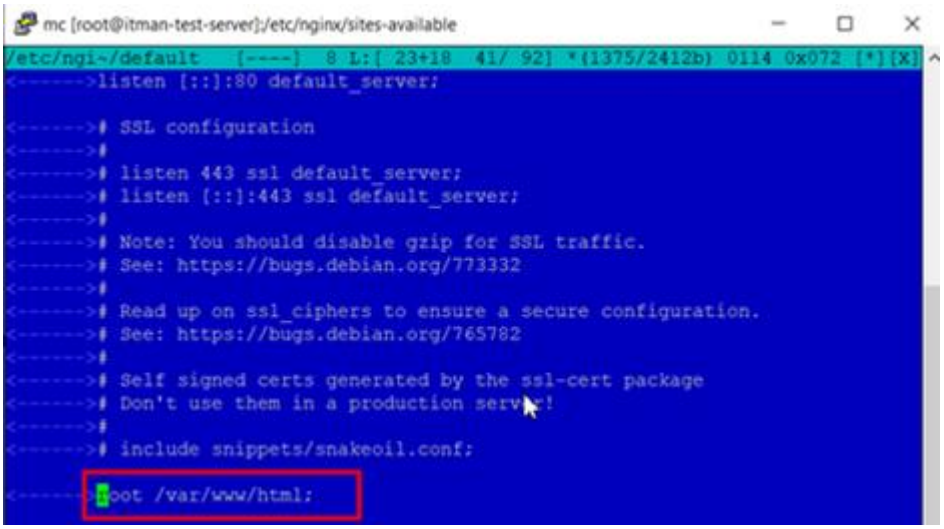
- Откройте директорию с папками с помощью команды:

```
$ sudo mc
```

- Проверьте, в какой каталог необходимо перенести данных WEB, для этого:
  - Перейдите в **/etc/nginx/sites-available**.
  - Откройте файл **default** и добавьте в него данные:



- Посмотрите, какой адрес указан в разделе **root** - это необходимо для следующего шага.



- Откройте файл **...\Server-Linux\Server-Linux\Settings\ServerConfig.**.
- В поле **CommonDirectory** укажите директорию, в которую будет установлен сервер, например, "/home/itmans"

```
...
"CommonDirectory": "/home/itmans",
...
```

- Скопируйте содержимое каталога **iTManDiscovery-Web/html** в директорию, указанную в разделе root файла **default** (см. предыдущий шаг).

- Откройте директорию **Server-***-Linux.**
- Откройте файл **..\Server-Linux\Reader-Linux\Settings\ReaderConfig.json**.
- В поле **CommonDirectory** укажите директорию, в которую будет установлен модуль Reader, например, "/home/itmanr"

```
...
"CommonDirectory": "/home/itmanr",
...
```

- Скопируйте архив каталога **Server-Linux** (версии могут отличаться) в любую отдельную директорию, где будут хранится данные для настройки системы, например, **/home/project** (например с помощью WinCSP).
- Разархивируйте данные в эту директорию или создайте поддиректорию **Server-Linux**, в зависимости как вам удобно будет с ней работать.
- В общей директории **Server-Linux** запустите файл **Discovery.Server.exe**.
  В случае, если запуск не произошел (ничего не происходит), то проверьте права доступа, они должны соответствовать: **rwxr-xr-x**.
  Если права другие, то для всех директорий, с которыми будете работать, используйте команду:

```
$ chmod -R 755 НазваниеДиректории/
```

- Выйдите в раздел команд linux.
- Запустите nginx с помощью следующей команды:

```
$ sudo systemctl start nginx
```

3.6. Установка Discovery Interact

3.6.1. Установка на Linux

Для установки интеракта на Linux используйте пакет **interact.deb** из дистрибутива.

У администратора должна была возможность запустить установку от имени пользователя **root**.

Далее:

- С помощью команды **$ cd** перейдите в директорию, в которую вы сохраните установщик **interact.deb**.

- Запустите установщик с помощью команды ниже, при этом подставьте в нее:
    - Вместо "DiscoveryInteract_LinuxNative_N.NN.NN.deb" актуальное название файла дистрибутива.
    - Параметры установки (см. описание ключей ниже).

```
$ sudo dpkg -i packs/ DiscoveryInteract_LinuxNative_N.NN.NN.deb && sudo
/usr/local/bin/configure-interact.sh --common_directory
/var/tmp/discovery-platform/interact2/ --company_guid NNNNNNNN-NNNN-
NNNN-NNNN-NNNNNNNNNNNN --self_port_first 5078 --self_port_last 5079 --
https_
server_ip 0.0.0.0 --https_server_port 5058 --main_server_ip 127.0.0.1 --
main_server_port 5087
```

Ключи для интеракта:

-c | --company_guid	**GUID ключ вашей компании (вида NNNNNNNN-NNNN-NNNN-NNNN-NNNNNNNNNNNN), который вы получили при покупке системы.**
-d | --common_directory	Путь к директории с временными файлами.
-f | --self_port_first	Стартовый TCP порт интеракта, по умолчанию 5078.
-l | --self_port_last	Конечный TCP порт интеракта, по умолчанию 5079.
-i | --https_server_ip	IP адрес, на который агент откроет https сервер. Обычно: • 0.0.0.0 - если нужно, чтобы сервер был доступен из вне. • 127.0.0.1 - только для локального хоста.
-p | --https_server_port	Открываемый https порт, по умолчанию 5058.
-m | --main_server_ip	IP адрес, по которому обращаться к серверу.
-s | --main_server_port	Порт, по которому обращаться к серверу, по умолчанию 5087.
-h | --help	Show this message and exit.

- Проверьте статус установки с помощью команды:

```
$sudo systemctl status discoveri-platform-interact.service
```

- Создайте директорию **SSL** для интеракта.

  Вы можете использовать директорию **SSL** по умолчанию или создать свою в директории **ect**.

- Перенесите в нее ключи.
- Перейдите в директорию **ect/discovery-platfom** и откройте файл **interact.json**
- Вносите в него следующие значения (названия ключей и сертификатов указаны для примера, вам необходимо вписать соответствующие данные):

```
    "sslCertificateKey" : "/etc/SSL/rootCA.key",
    "sslCertificate" : " /etc/SSL/rootCA.pem"
```

- Сохраните файл.
- Перезапустите интеракт с помощью команды:

```
$ sudo systemctl restart discovery-platform-interact
```

3.6.2. Установка на RedHat

Для установки интеракта на RedHat используйте пакет **interact.rpm** из дистрибутива.

У администратора должна была возможность запустить установку от имени пользователя **root**.

Далее:

- С помощью команды **$ cd** перейдите в директорию, в которую вы сохраните установщик **interact.rpm**.
- Запустите установщик с помощью команды (вместо 4.17.0 подставьте соответстювущую версию):

```
$ sudo rpm -i /home/itman/discovery-platform-interact-4.17.0.rpm
```

- Измените параметры установки (значения ключей такие же как для linux, см. инструкцию выше):

```
$ sudo rpm -i
home/ge/rpmbuild/RPMS/x86_64/DiscoveryInteract_LinuxNative_N.NN.NN.rpm
&& sudo /usr/local/bin/configure-interact.sh --common_directory
/var/tmp/discovery-platform/interact2/ --company_guid NNNNNNNN-NNNN-
NNNN-NNNN-NNNNNNNNNNNN--self_port_first 5078 --self_port_last 5079 --
https_server_ip 0.0.0.0 --https_server_port 5058
```

- Создайте директорию **SSL** для интеракта.

  Вы можете использовать директорию **SSL** по умолчанию или создать свою в директории **ect**.

- Перенесите в нее ключи.
- Перейдите в директорию **ect/discovery-platfom** и откройте файл **interact.json**
- Вносите в него следующие значения (названия ключей и сертификатов указаны для примера, вам необходимо вписать соответствующие данные):

```
    "sslCertificateKey" : "/etc/SSL/rootCA.key",
    "sslCertificate" : " /etc/SSL/rootCA.pem"
```

- Сохраните файл.
- Перезапустите интеракт.

3.7. Установка Discovery Agent на ОС Linux

В системе есть агенты двух типов:

- **DiscoveryAgentNative** – агент с функцией сетевой инвентаризации, позволяет сканировать сеть, AD и SQL.
- **DiscoveryAgentLiteNative** – агент без функции сетевой инвентаризации. Такой агент будет сканировать только компьютер, на котором он установлен.

При установке используйте файл в соответствии с типом агента.

Помимо этого, агент может быть:

- **Конечной точкой обработки данных** - в этом случае агент может только отправлять данные интеракту.
- **Работать в цепочке агентов** - в этом случае агент может получать данные от других агентов и передавать их интеракту..

При установке это повлияет на настройку портов агента.

3.7.1. Настройка учетных записей

Установка агента должна запускаться под учетной записью, позволяющей sudo повышать права без необходимости ввода интерактивного пароля. Для установки пакета агента инвентаризации требуются **root** привилегии уровня.

При дальнейшей работе агенты инвентаризации должны работать как **root** для установки и запуска служб агента на локальном устройстве.
При работе агента под другой ролью, мы не гарантируем инвентаризацию данных, так как у системы может не хватить прав.

3.7.2. Получение файла установки

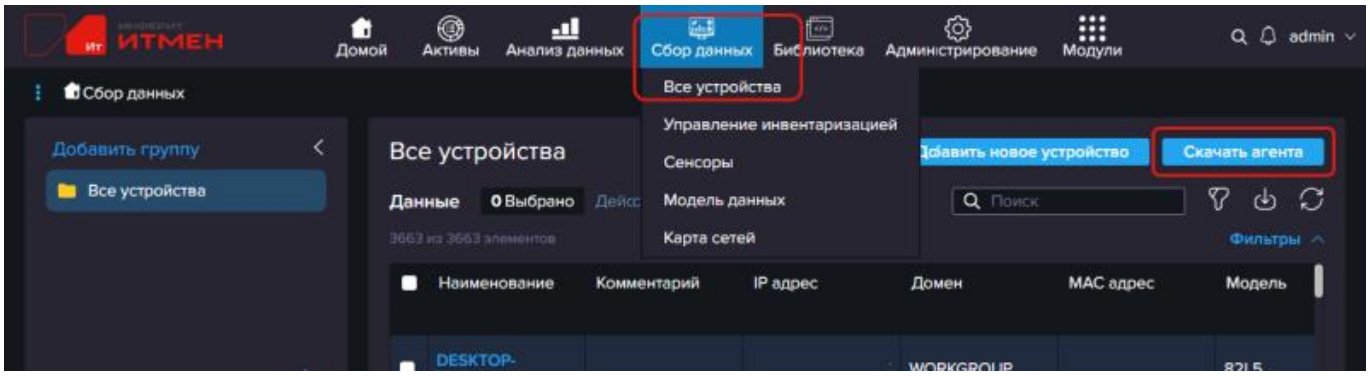
Перед установкой получите файл агента для установки.
Для этого:

- Возьмите файл из дистрибутива.
- Или скачайте через веб-интерфейс системы в разделе **Сбор данных / Все устройства**.

Обратите внимание, агенты бывают двух типов, вам нужно использовать только один из файлов, в зависимости от функций, которые будет выполнять агент:

- **DiscoveryAgentNative.deb** – агент с функцией сетевой инвентаризации, позволяет сканировать сеть, AD и SQL.
- **DiscoveryAgentLiteNative.deb** – агент без функции сетевой инвентаризации. Такой агент будет сканировать только компьютер, на котором он установлен.

После получения файла разархивируйте его.

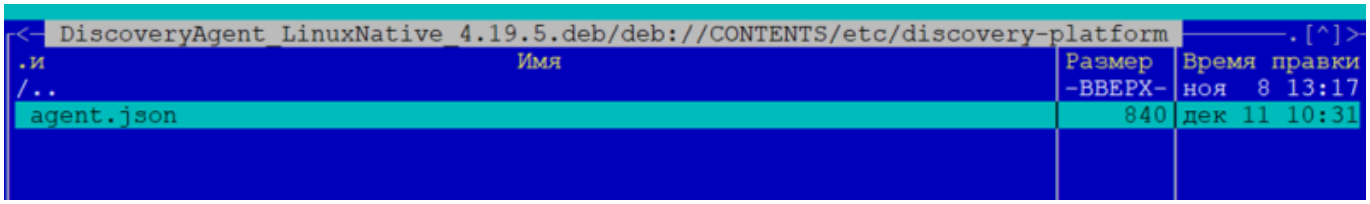


Вы можете установить агент двумя способами:

- Через интерфейс.
- В тихом режиме.

3.7.3. Установка через интерфейс

- С помощью команды **$ sudo mc** перейдите в директорию, в которую вы сохранили файл установки агента.
- Откройте пакет **deb/Contens/ect/discovery-platform/agent.json**



- Введите данные в файле:
    - CompanyGUID - GUID ключ **вашей компании** (вида NNNNNNNN-NNNN-NNNN-NNNN-NNNNNNNNNNNN), который вы получили при покупке системы.
    - IP адрес, на котором установлен интеракт (если интеракт установлен на том же компьютере, что и агент, то оставьте значение по умолчанию).
    - Порт интеракта, к которому будут подключены агенты (по умолчанию 5078)
    - Признак открытия портов агента:
        - **false** - порты закрыты, в этом случае агент является конечной точкой и на него не могут передавать данные другие агенты.
        - **true** - порты открыты, агент может получать данные от других агентов (то

есть является участником цепочки).

- Стартовый порт прослушивания: по умолчанию 5088.
- Конечный порт прослушивания: по умолчанию 5089.
- Метки - специальные метки (теги) компьютера, на который устанавливается агент (если есть необходимость), например, VDI. В дальнейшем они используются при обогащении данных в процессе инвентаризации.

Вы сможете изменить настройки установки в дальнейшем в файле config_agent.json.

- Сохраните файл и проверьте статус установки с помощью команды:

```
$sudo systemctl status discovery-platform-agent.service
```

- Перезапустите агент с помощью команды:

```
$sudo systemctl restart discovery-platform-agent.service
```

3.7.4. Тихая установка агента

Для тихой установке выполните команду ниже, подставив в нее:

- Вместо "DiscoveryAgent_LinuxNative_N.NN.N.deb" актуальное название файла дистрибутива.
- Параметры установки (см. описание ключей ниже).

Установка для deb:

```
sudo dpkg -i DiscoveryAgent_LinuxNative_N.NN.N.deb  && sudo
/usr/local/bin/configure-agent.sh --common_directory /var/tmp/discovery-
platform/agent/ --company_guid NNNNNNNN-NNNN-NNNN-NNNN-NNNNNNNNNNNN --
agent_ip 127.0.0.1 --agent_port 5078 --self_port_is_open fa
lse --self_port_first 5088 --self_port_last 5089 --agent_tags \"vdi\"
```

Установка для rpm:

```
sudo rpm -i
/home/ge/rpmbuild/RPMS/x86_64/DiscoveryAgent_LinuxNative_N.NN.NN.rpm &&
sudo /usr/local/bin/configure-agent.sh --common_directory
/var/tmp/discovery-platform/agent2/ --company_guid NNNNNNNN-NNNN-NNNN-
NNNN-NNNNNNNNNNNN --agent_ip 127.0.0.1 --agent_port 5078 --
self_port_is_open fa
lse --self_port_first 5088 --self_port_last 5089 --agent_tags \"vdi\"
```

Ключи для агента:

-d | --	**Рабочий каталог агента, укажите путь сохранения временных данных при**

common_directory	**инвентаризации.**
-c | --company_guid	GUID ключ **вашей компании** (вида NNNNNNNN-NNNN-NNNN-NNNN-NNNNNNNNNNNN), который вы получили при покупке системы.
-a | --agent_ip	IP адрес родительского элемента (агент или интеракт), который должен получать данные от настраиваемого агента.
-p | --agent_port	Порт родительского элемента (агент или интеракт), который должен получать данные от настраиваемого агента (по умолчанию для интеракта: 5078, для агента-родителя: 5088).
-o | --self_port_is_open	Признак открытия портов агента: • **false** - порты закрыты, в этом случае агент является конечной точкой и на него не могут передавать данные другие агенты. • **true** - порты открыты, агент может получать данные от других агентов, то есть участвует в цепочке.
-f | --self_port_first	Начальный и конечный порты настраиваемого агента, открытые на прослушивание. Используется, если включено открытие портов агента, то есть агент участвует в цепочке и может получать данные от других агентов. Проверьте, что указанные порты не заняты. Например, если сервер и агент установлены на одном компьютере, и вы укажите порт, который занят сервером, то агент не будет работать корректно. Если открытие портов выключено, то настроенные порты игнорируются, даже если заполнены.
-l | --self_port_last	
-t | --agent_tags	Специальные метки (теги) компьютера, на который устанавливается агент (если есть необходимость), например, VDI. В дальнейшем они используются при обогащении данных в процессе инвентаризации. Перечисляются в виде строки (двойные кавычки необходимо экранировать. Например \"tag1\",\"vdi\").
-h | --help	Показать список доступных параметров и их значения по умолчанию.

Вы можете использовать краткие или полные имена ключей, например, -c и --company_guid -

равноценные ключи. Можно указывать только нужные параметры, которые необходимо изменить все остальные будут установлены значениями по умолчанию.

3.7.5. Ошибки установки

Если установка не выполнилась, то:

- Проверьте, создались ли данные в директории, которую вы указали в строке **CommonDirectory**. Если данных нет, то возможно у вас нет прав для записи в эту директорию
- Если данные создались, то проверьте запустился ли ваш агент через команду **$ htop**
- Если данные создались, и агент запустился, то проверьте корректность заполнения адреса в строках **ServerAddress** и **InteractServerAddress**.

3.7.6. Настройка цепочки агентов

По умолчанию предполагается, что агенты передают собранные данные интеракту.

Вы можете настроить цепочку агентов, в которой агенты будут передавать данные по цепочке друг другу, а последний будет передавать данные интеракту.
Это используется, например, в случае если интеракт и агенты находятся в разных локальных сетях без доступа во внешнюю сеть.
Например, если в вашей компании несколько офисов, и в каждом из них только один компьютер имеет доступ во внешнюю сеть.

В таком случае:

- На компьютер, имеющий доступ к внешней сети, устанавливается агент, который выполняет роль шлюза.
- Все агенты внутри локальной сети отправляют данные на агент-шлюз.
- Далее агент-шлюз передает данные или интеракту или (как показано на схеме ниже), другому агенту-шлюзу, который в свою очередь уже передает их интеракту.

3.8. Настройка SSL сертификата для ОС Linux

Для обеспечения безопасности системы необходимо настроить работу компонентов системы по HTTPS протоколу. Для этого необходим SSL сертификат, выданный удостоверяющим центром.

Подключение SSL сертификата выполняется в процессе установки сервера и интеракта. Если вы выполнили соответствующие инструкции, то дополнительных действий не требуется.

В случае, если вы все сделали по инструкциям, но система не работает по HTTPS, - проверьте все этапы настройки, описанные далее.

3.8.1. Проверка основных настроек

Перед началом работы:

- Выпустите ключ и сертификат для SSL.
  Обратите внимание на требования к SSL сертификату.
- Остановите **Discovery Server** и **Reader** в диспетчере задач, если они уже были установлены и запущены.

Далее:

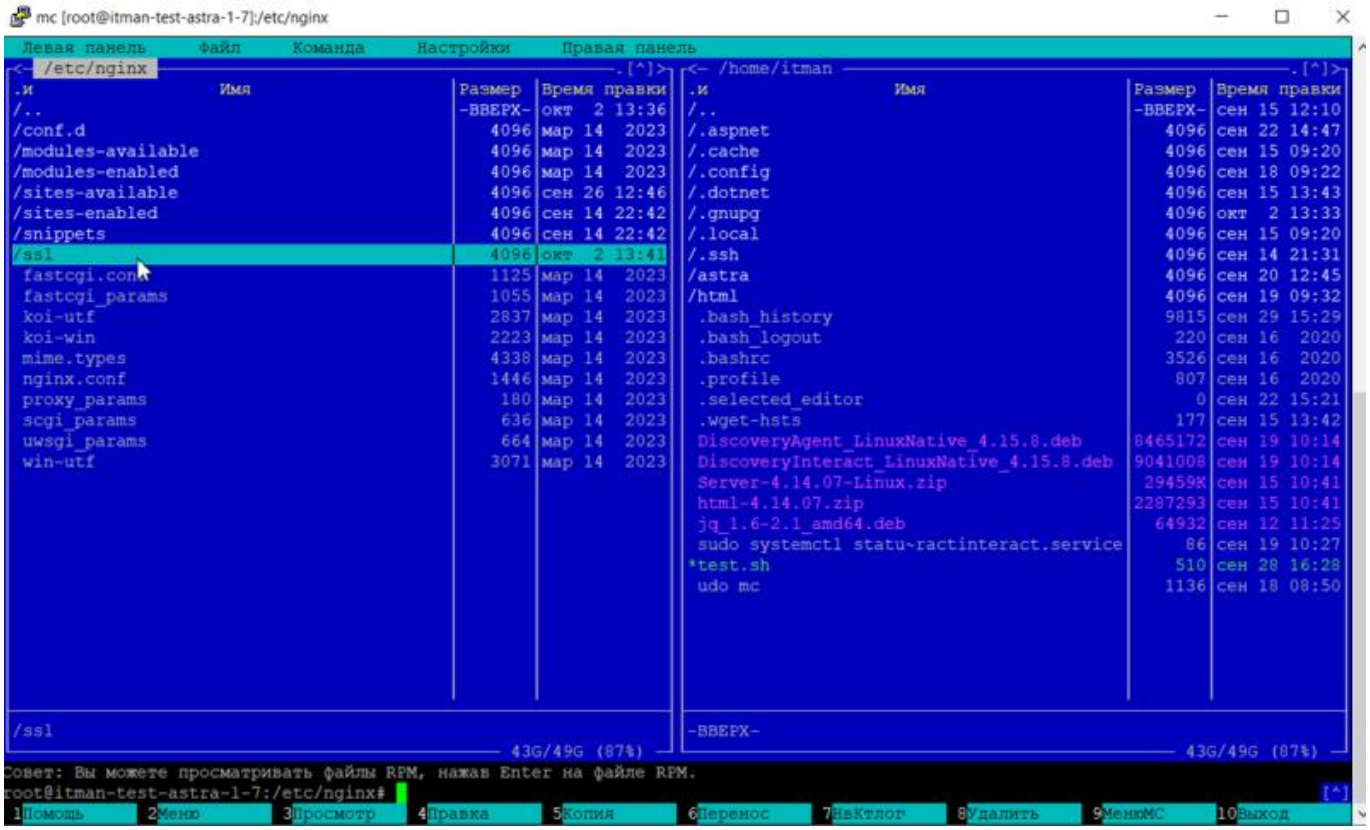
- Разместите сертификаты в доверенный и промежуточный каталоги: **/usr/local/share/ca-certificates**
- Обновите сертификаты командой:

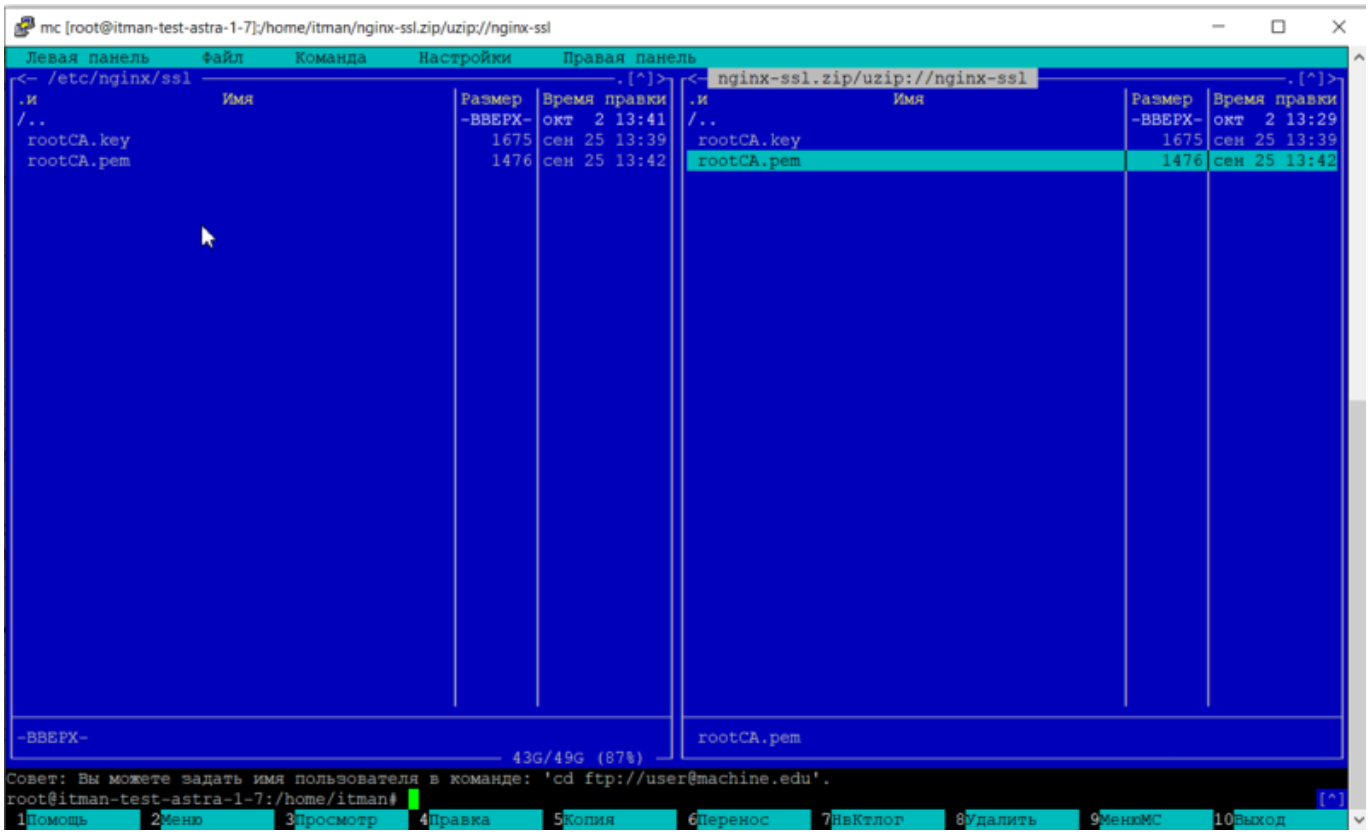
```
update-ca-certificates --verbose
```

- Проверьте результат с помощью команды (в конце должны появиться новые записи):

```
awk -v cmd='openssl x509 -noout -subject' '
/BEGIN/{close(cmd)};{print | cmd}' < /etc/ssl/certs/ca-certificates.crt
```

- Настройте отображение веб-сервиса.
  Для этого:
  - Откройте директорию, в которую установлен **nginx**.
  - Создайте в ней директорию с названием **SSL**.
  - Скопируйте в нее файлы приватного ключа сертификата.



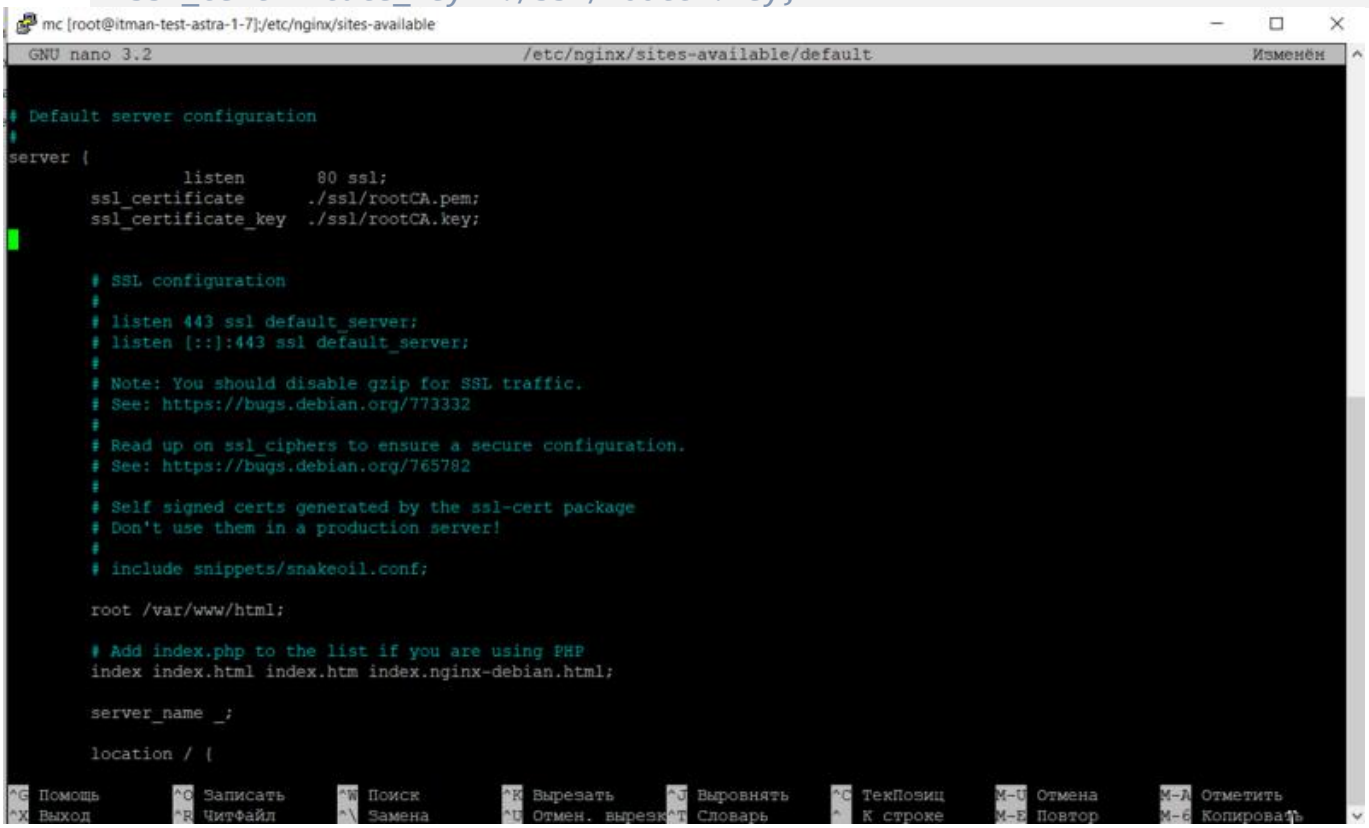


- Перейдите в директорию **sites-anailable** в **nginx.**

- Отредактируйте файл конфигурации **default**, добавив в него данные:

  Обратите внимание, далее приведен пример - наименования файлов и расширения у вас будут свои.

```
server {
  listen       80 ssl;
ssl_certificate      ./ssl/rootCA.pem;
  ssl_certificate_key  ./ssl/rootCA.key;
```



- Настройте передачу данных.

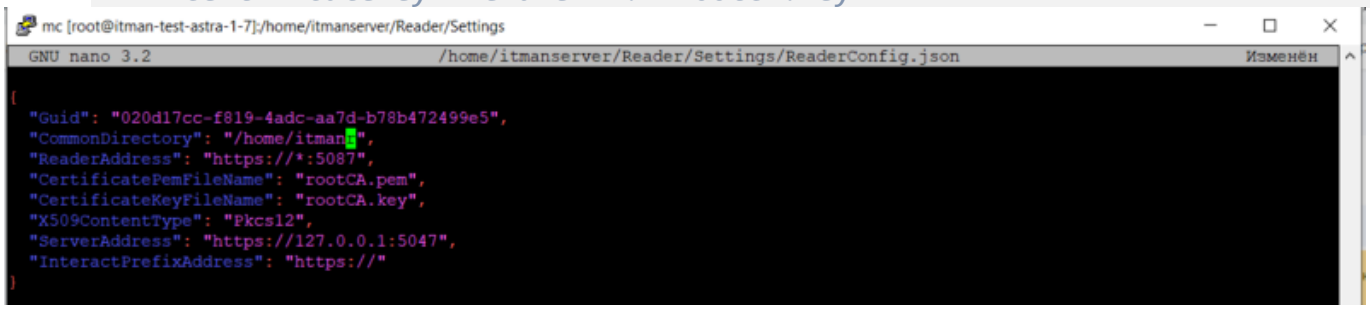
  Для этого:

  - Откройте директорию сервера (та директория из которой вы будете запускать сервер).
  - Создайте в ней директорию с названием **SSL**, если это не было сделано ранее.
  - Скопируйте в нее файлы приватного ключа сертификата.

- Перейдите директорию **Reader** в общей директории сервера.
- Создайте в ней директорию с названием **SSL**, если это не было сделано ранее.
- Скопируйте в нее файлы приватного ключа сертификата.

- Перейдите директорию **Reader/Settings**
- Отредактируйте файл **ReaderConfig.json**:
    - Замените все значения «http» на значения «https».
    - Добавьте в него две строчки с названием ключа и сертификата.
      Обратите внимание, далее приведен пример - наименования файлов и расширения у вас будут свои.

```
"CertificatePemFileName" : "rootCA.pem",
"CertificateKeyFileName " : "rootCA.key"
```



- Перейдите в директорию **Settings** в общей директории сервера.
- Отредактируйте файл **ServerConfig.json**:
    - Замените все значения «http» на значения «https».
    - Добавьте в него две строчки с названием ключа и сертификата.
      Обратите внимание, далее приведен пример - наименования файлов и расширения у вас будут свои.

```
"CertificatePemFileName" : "rootCA.pem",
" CertificateKeyFileName " : "rootCA.key"
```

- Создайте директорию **SSL** для интеракта.
  Вы можете использовать директорию по умолчанию или создать свою в каталоге **etc**.

- Скопируйте в нее файлы приватного ключа сертификата.
- Перейдите в директорию **ect/discovery-platfom.**
- Отредактируйте файл **interact**, добавив в него данные.

  Обратите внимание, далее приведен пример - наименования файлов и расширения у вас будут свои.

```
    "sslCertificateKey" : "/etc/SSL/rootCA.key",
    "sslCertificate" : " /etc/SSL/rootCA.pem"
  }
},
"httpServer": {
  "host": "0.0.0.0",
  "port": 5058,
  "sslCertificateKey": "/etc/discovery-platform/rootCA.key",
  "sslCertificate": "/etc/discovery-platform/rootCA.pem"
},
```

- Перезапустите интеракт с помощью команды:

```
$ sudo systemctl restart discovery-platfom-interact
```

- Перейдите в общую директорию сервера.
- Перезапустите сервер с помощью файла **runServer**.

3.8.2. Дополнительная настройка для самоподписанных сертификатов

По требованиям безопасности SSL сертификат **должен быть выдан удостоверяющим центром (CA)**. Однако вы можете использовать самоподписанный сертификат, например, на этапе тестирования. В этом случае добавьте сертификат в доверенные в браузерах пользователей, работающих с системой, аналогично тому, как это делается при установке на Windows.

При установке агента, который выполняет роль шлюза, и получает данные от других агентов:

- Используйте тип агента **DiscoveryAgentNative.**
- Заполните данные родительского элемента - интеракта или другого агента-шлюза, который будет получать данные от настраиваемого агента:
    - **agent_ip** - IP адрес родительского элемента.
    - **agent_port** - порт родительского элемента (по умолчанию для агента-родителя: 5088).
- Включите открытие портов агента **(self_port_is_open=true)**.
- Укажите диапазон портов настраиваемого агента (**self_port_first, self_port_last**), на которые будут присылать данные другие агенты.

При установке агента, который является конечным (собирает данные только с того устройства, на котором установлен, и не получает данные от других агентов):

- Используйте тип агента **DiscoveryAgentLightNative.**
- Заполните данные родительского элемента - **агента-шлюза**, который будет получать

данные от настраиваемого агента:

- **agent_ip** - IP адрес родительского элемента.
- **agent_port** - порт родительского элемента.

- Выключите открытие портов агента **(self_port_is_open=true)**.
- Порты настраиваемого агента (**self_port_first, self_port_last**), на которые будут присылать данные другие агенты, можно оставить со значениями по умолчанию. Они будут игнорироваться.

Вы также можете отредактировать эти данные после установки в файлах конфигурации.

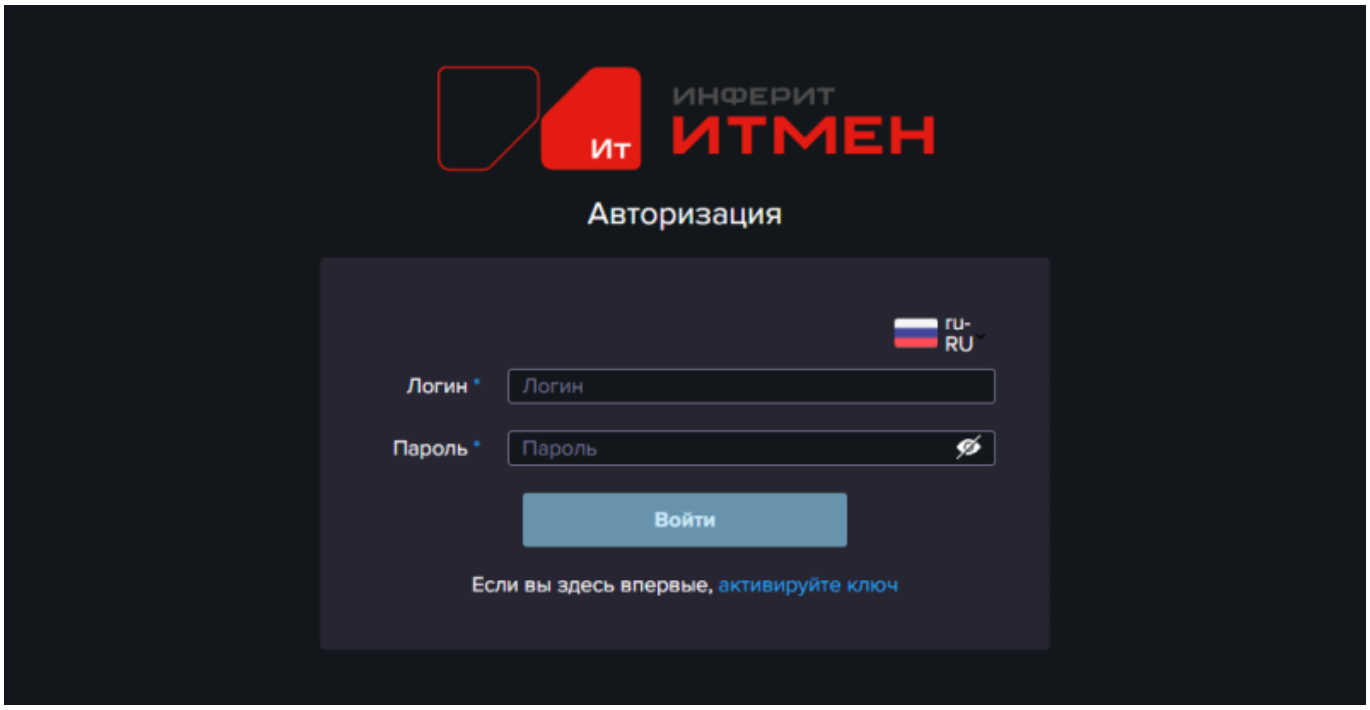
3.9. Запуск сервера на ОС Linux

После того, как сервер установлен:

- Запустите сервер, для этого в директории, куда он был установлен, запустите: **runServer**.
- Из командной строки запустите nginx:

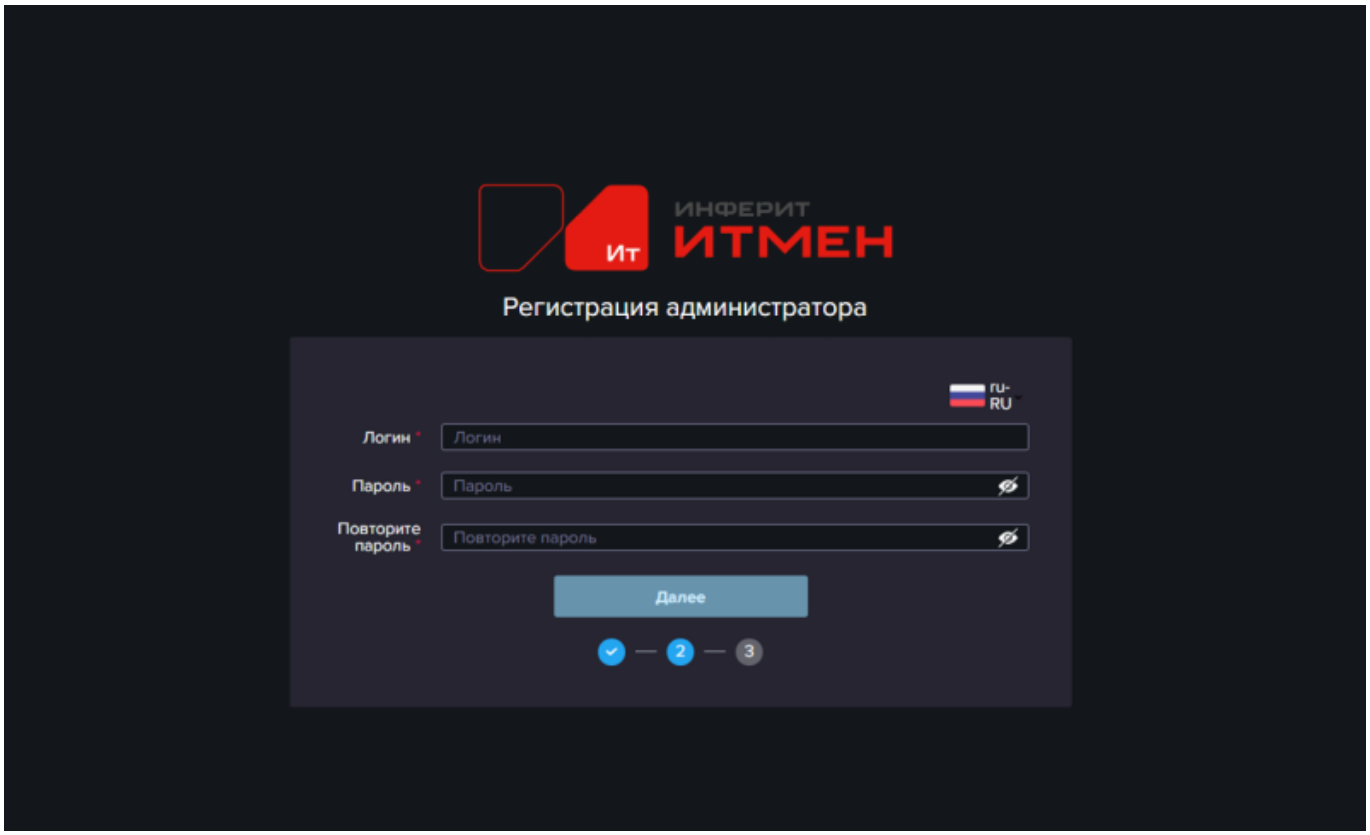
```
sudo systemctl start nginx
```

- Откройте браузер и введите публичный адрес вашего компьютера, на который вы установили сервер.
- Откроется окно авторизации.

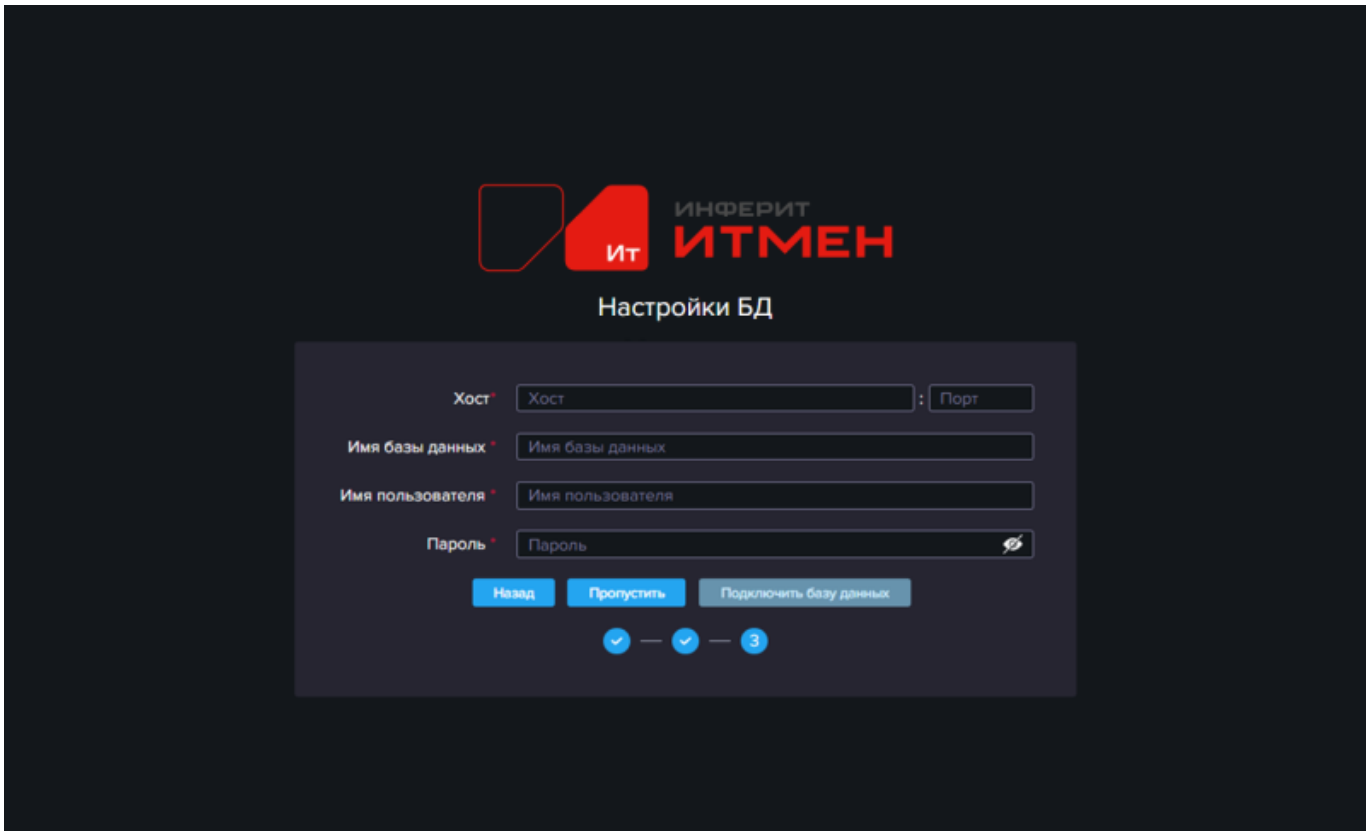


- Нажмите ссылку **активируйте ключ**.
- Откроется окно ввода ключа.

- Загрузите файл с ключем, который вы получили вместе с системой.
- Откроется окно регистрации администратора.



- Зарегистрируйте учетную запись администратора сервера, для этого:
    - Введите логин и пароль. Не забудьте сохранить авторизационные данные в надежном месте.
    - Нажмите кнопку **Далее**.
- Откроется окно для ввода данных СУБД, в которую в дальнейшем будут сохраняться данные.



- Заполните данные СУБД и нажмите кнопку **Подключить базу данных**.

  Вы можете пропустить этот шаг и ввести данные позднее. В этом случае нажмите кнопку **Пропустить**.

- После прохождения авторизации вам откроется окно первого входа, где необходимо нажать кнопку **Далее** и перейти в систему.

3.9.1. Запуск сервера, как службы

Для запуска сервера, как службы:

- Выполните команды, чтобы создать файл, присвоить права и затем открыть файл:

```
$ touch /etc/systemd/system/discovery-platform.reader.service
$ chmod 664 /etc/systemd/system/discovery-platform.reader.service
@ nano /etc/systemd/system/discovery-platform.reader.service
```

- Внесите в файл следующие данные:

```
[Unit]
Description=discovery-platform.reader
After=network.target
[Service]
Type=simple
```

```
User=root
ExecStart=/home/itman/Server-4.16-Linux/Reader/Discovery.Reader
&/dev/null&
[Install]
WantedBy=multi-user.target
```

- Сохраните и закройте файл.
- Введите в текстовом редакторе заново команды:

```
$ touch /etc/systemd/system/discovery-platform.server.service
$ chmod 664 /etc/systemd/system/discovery-platform.server.service
$ nano /etc/systemd/system/discovery-platform.server.service
```

- Внесите в файл следующие данные:

```
[Unit]
Description=discovery-platform.server
After=network.target

[Service]
Type=simple
User=root
ExecStart=/home/itman/Server-4.16-Linux/Discovery.Server &/dev/null&

[Install]
WantedBy=multi-user.target
```

- Сохраните и закройте файл.
- Перезапустите менеджер служб с помощью команды:

```
systemctl deamon-reload
```

4. Проверка работы системы

4.1. Начало работы

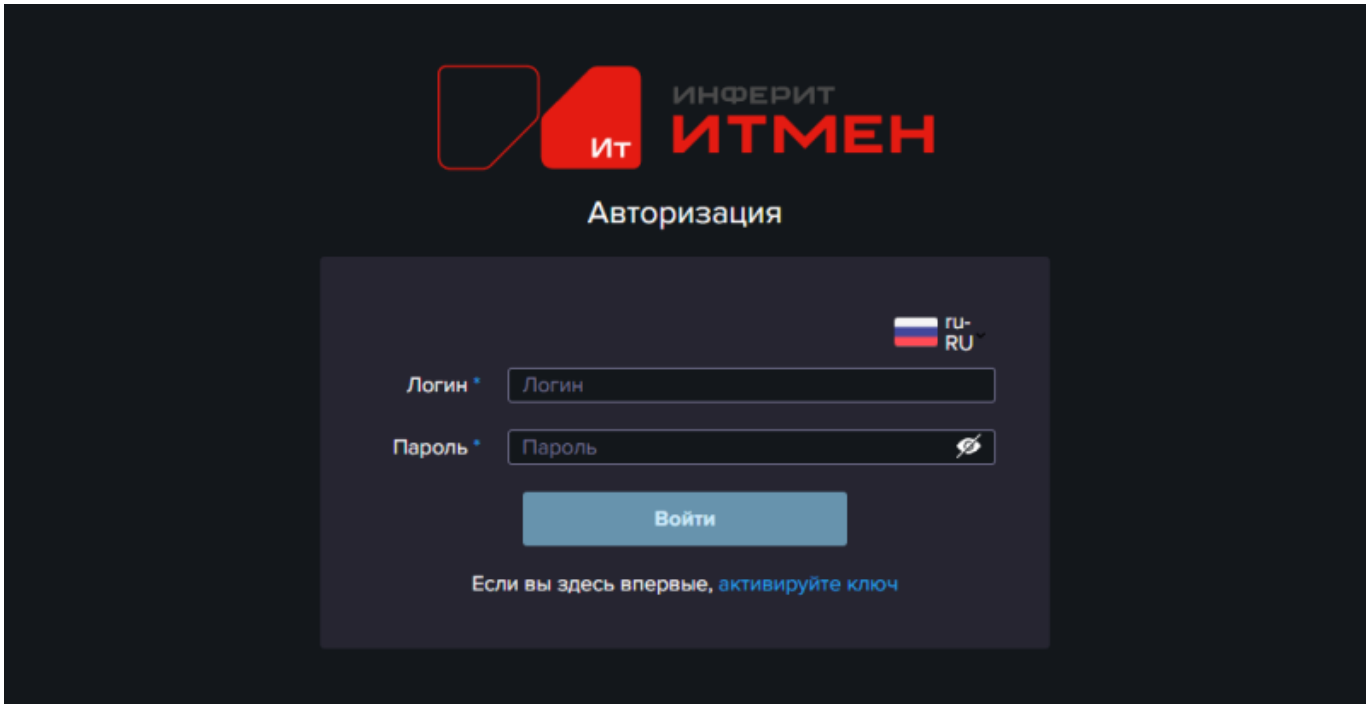
- Откройте браузер и введите публичный адрес вашего компьютера, на который вы установили сервер.

  Если вы видите ошибку и веб-интерфейс системы не открывается, то:

  - Проверьте, что вы ввели адрес с HTTPS.
  - Если указан HTTPS, то проверьте настройку SSL сертификата.
  - Если ошибка осталась, проверьте настройки портов в файле конфигурации nginx.

- В случае, если веб-интефрейс доступен по HTTPS, откроется окно авторизации.

- Пройдите авторизацию с учетными данными администратора.

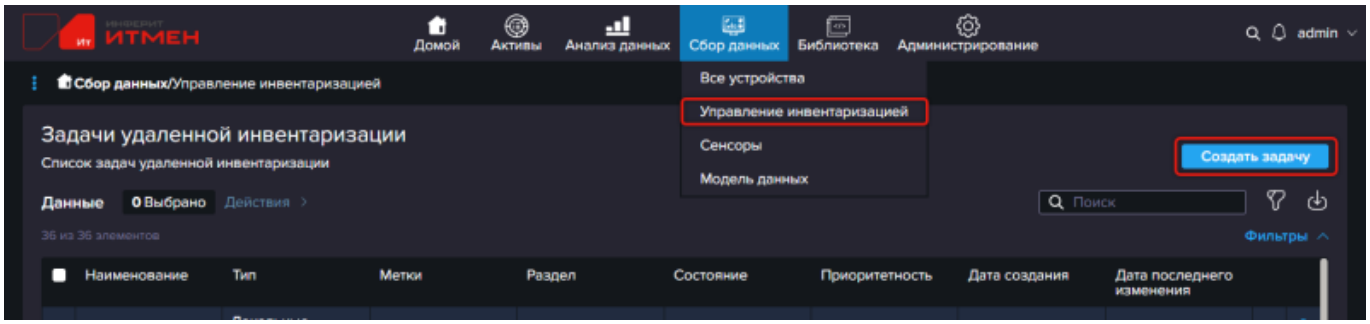


4.2. Проверка подключения СУБД

- Перейдите в раздел **Администрирование / Типы устройств**.
- Проверьте, что в таблице отображаются предустановленные типы устройств:
    - Если данные есть, то значит СУБД подключена корректно.
    - В случае отсутствия данных проверьте настройки подключения к базе в разделе **Администрирование / Настройки БД**.

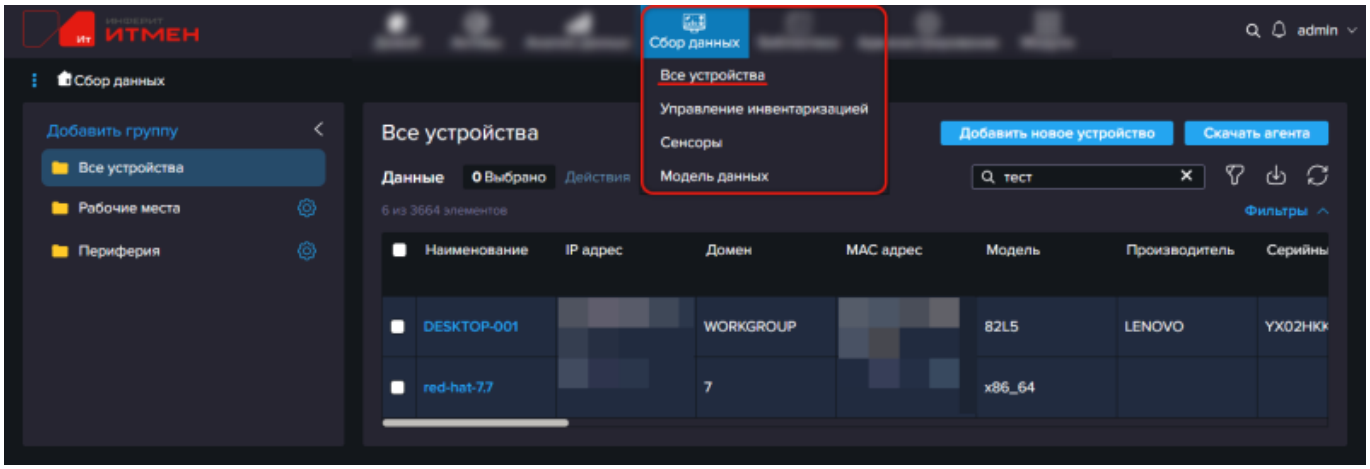
4.3. Проверка работы агентов

- Перейдите в раздел **Сбор данных / Управление инвентаризацией.**
- Нажмите кнопку **Создать задачу.**



- В открывшемся окне нажмите кнопку **Сбор данных инвентаризации с агента**.
- Откроется мастер создания задачи.
- Заполните шаги мастера.
  Для проверки работы достаточно:
  - На шаге **Настройки задачи** заполните **Наименование задачи**.
  - На шаге **Настройка агентов** включите **Выбрать всех агентов**.
  - Остальные настройки мастера оставьте по умолчанию.
  - Нажмите кнопку **Сохранить** на последнем шаге мастера.
- Дождитесь завершения выполнения задачи.
  Если вы не заполняли дату старта, то при сохранении задачи дата старта заполнится текущей датой и временем, и задача запустится сразу после создания. Если вы заполнили дату создания, то задача запустится в указанные дату и время. Вы также можете

  принудительно запустить задачу, нажав иконку в строке задачи.
- Проверьте результат выполнения задачи:
  - Перейдите в раздел **Сбор данных / Все устройства**.
  - В случае корректной работы системы после выполнения задач в разделе появится список найденных устройств.
  - Если данные не появились, или они неполные, проверьте: историю выполненеия задачи, логи.



5. Файлы конфигурации системы

5.1. Файл конфигурации сервера: ServerConfig.json

Основные характеристики:

Назначение	Содержит настройки Discovery Server
Наименование	**ServerConfig.json**
Расположение по умолчанию	Для ОС Windows: **C:\Program Files\Discovery Server\Server\Settings** Для ОС Linux: **/opt/Discovery.Server/Settings/**

Параметры:

- **Guid** - уникальный идентификатор конфигурации.
- **CommonDirectory** - путь к рабочей директории сервера, в которую сохраняются данные транзакций, логи.
  Формат зависит от операционной системы.
  - Пример для Windows: **C:\\ProgramData\\Discovery.Server**
  - Пример для Linux: **/home/itmans** или **/var/tmp/discovery-server**
- **ServerAddress** - порт, на котором работает сервер обработки данных, в формате: "https://*:5047", где "5047" это номер порта.
- **CertificateFileName** - название файла SSL сертификата.
- **CertificateKeyFileName** - название файла ключа SSL.

- **X509ContentType** - формат сертификата x509.
- **EnKey** - настройка полного шифрования данных:
  - **true** - шифрование включено,
  - **false** - шифрование выключено.
- **ReaderPrefixAddress** - протокол взаимодействия с модулем Reader, должен быть равен "https://" при использовании SSL.
- **ReaderAddress** - IP адрес модуля Writer, укажите "127.0.0.1", если модуль расположен на том же компьютере, что и сервер.
- **ReaderPort** - порт модуля Reader.
- **WriterPrefixAddress** - протокол взаимодействия с модулем Reader, должен быть равен "https://" при использовании SSL.
- **WriterAddress** - IP адрес модуля Writer, укажите "127.0.0.1", если модуль расположен на том же компьютере, что и сервер.
- **WriterPort** - порт модуля Writer.

Пример заполнения для ОС Windows:

```
{
  "Guid": "NNNN-NNNN-NNNN-NNNN-NNNNNNNNNNNN",
  "CommonDirectory": "C:\\ProgramData\\Discovery.Server",
  "ServerAddress": "https://*:5047",
  "CertificateFileName": "rootCA.pem",
  "CertificateKeyFileName": "rootCA.key",
  "X509ContentType": "Pkcs12",
  "EnKey" : true,
  "ReaderPrefixAddress": "https://",
  "ReaderAddress": "127.0.0.1",
  "ReaderPort": 5087,
  "WriterPrefixAddress": "https://",
  "WriterAddress": "127.0.0.1",
  "WriterPort": 5097
}
```

5.2. Файл конфигурации модуля Reader: ReaderConfig.json

Основные характеристики:

Назначение	Содержит настройки Discovery Reader
Наименование	**ReaderConfig.json**
Расположение по умолчанию	Для ОС Windows: C:\Program Files\Discovery Server\Server\Reader\Settings Для ОС Linux: /opt/Discovery.Reader/Settings/

Параметры:

- **Guid** - уникальный идентификатор конфигурации.
- **CommonDirectory** - адрес директории, в которую сохраняются данные модуля, включая транзакции, логи.
  Формат зависит от операционной системы.
  - Пример для Windows: **C:\\ProgramData\\Discovery.Reader**
  - Пример для Linux: **/home/itmanr** или **/var/tmp/discovery-reader**
- **ReaderAddress** - порт, на котором работает Reader.
- **CertificateFileName** - название файла SSL сертификата.
- **CertificateKeyFileName** - название файла ключа SSL.
- **X509ContentType** - формат сертификата x509.
- **EnKey** - определяет шифрование данных:
  - **true** - включает полное шифрование данных,
  - **false** - отключает шифрование (файлы будут расшифрованы).
- **ServerAddress** - IP адрес и порт сервера.
- **WriterAddress** - IP адрес и порт модуля Writer.
- **InteractPrefixAddress** - протокол взаимодействия с модулем Interact, должен быть равен "https://" при использовании SSL.
- **VdiSyncPrefix** - тег, который будет указываться при настройке обогащения в инвентаризации, например: VDI_

Пример заполнения для ОС Windows

```
{
  "Guid": "NNNN-NNNN-NNNN-NNNN-NNNNNNNNNNNN",
```

```
  "CommonDirectory": "C:\\ProgramData\\Discovery.Reader",
  "ReaderAddress": "https://*:5087",
  "CertificateFileName": "rootCA.pem",
  "CertificateKeyFileName": "rootCA.key",
  "X509ContentType": "Pkcs12",
  "EnKey" : true,
  "ServerAddress": "https://127.0.0.1:5047",
  "WriterAddress": "https://127.0.0.1:5097",
  "InteractPrefixAddress": "https://",
  "VdiSyncPrefix": "VDI_"
}
```

5.3. Файл конфигурации логов модуля Reader: LoggingSettings.json

Основные характеристики:

Назначение	Содержит настройки логирования для модуля Discovery Reader
Наименование	**LoggingSettings.json**
Расположение по умолчанию	Для ОС Windows: C:\ProgramData\Discovery Server\Server\Reader\Settings Для ОС Linux: /opt/Discovery.Reader/Settings/

Параметры:

- **Verbose** - определяет максимальный уровень логирования.
  Варианты значений:
  - 0 - Trace
  - 1 - Debug
  - 2 - Info
  - 3 - Warning
  - 4 - Error
  - 5 - Critical

Например, если, выставить 2 (Info), то логироваться будут только сообщения Info, Warning, Error, Critical.

- **DaysPerLog** - определяет через сколько дней будет заведен новый файл current.log, а старый будет переименован в формат {дата_создания_лога}_{дата_окончания_лога}.log

Пример заполнения для ОС Windows / Linux

```
{
  "Verbose": 2,
  "DaysPerLog": 3
}
```

5.4. Файл конфигурации модуля Interact: config_interact.json

Основные характеристики:

Назначение	Содержит настройки для модуля Discovery Interact
Наименование	**config_interact.json**
Расположение по умолчанию	Для ОС Windows: C:\Program Files\Discovery Server\Server\InteractV2\etc Для ОС Linux: /ect/discovery-platform/

Параметры:

- **commonDirectory** - путь установки интеракта в формате соответствующей операционной системы, например:
  - Для Windows: **C:\\ProgramData\\discovery-platform\\interact**
  - Для Linux: **/var/tmp/discovery-platform/interact**
- **companyGuid** - GUID ключ **вашей компании** (вида NNNNNNNN-NNNN-NNNN-NNNN-NNNNNNNNNNNN), который вы получили при покупке системы.
- **transactionPackagesLimit** - количество архивов с результатами сбора, после которого будет уведомлен сервер что необходимо забрать пакеты.
- **transactionPackagesTimeout** - промежуток времени, через который интеракт проверяет, есть ли готовые к отправки пакеты с результатами.
- **syncModelInTransaction**
- **selfOpenedTcpPorts** - настройки портов:
  - **first** - стартовый TCP порт, по которому можно подключится к интеракту.
  - **last** – конечный TCP порт, по которому можно подключится к интеракту.
- **selfOpenedUdpPort:**
  - **isNeedOpenUdpServerPort**
  - **port**

- **webSocket**
  - **port**
- **httpServer** - настройки подключения к интеракту:
  - **host** - адрес который отрывает интеракт для подключения по https.
  - **port** - порт интеракта для подключения по https.
  - **sslCertificateKey** - путь к ключу для SSL сертификата в формате соответствующей операционной системы, например:
    - Для Windows: etc\\rootCA.key
    - Для Linux:
  - **sslCertificate** – путь к SSL сертификату в формате соответствующей операционной системы, например:
    - Для Windows: etc\\rootCA.pem
    - Для Linux:
- **mainServer** - настройки подключения к серверу и его модулям:
  - **serverHost** - адрес Reader, на который отправляются данные.
  - **serverPort** - порт сервера.
  - **serverPath** - адрес, по которому интеракт обращается в случае готовности пакета.
- **logger** - настройки логирования:
  - **verbose** - уровень логирования, варианты значений:
    - 0 - Trace
    - 1 - Debug
    - 2 - Info
    - 3 - Warning
    - 4 - Error
    - 5 - Critical
  - **directory** - название директории, в которую будут записываться логи.

  - **sink**
  - **maxFiles**
- **timeDivision**
  - **allocation**
  - **distributionAlgorithm**
  - **frameLength**
  - **heartBeatTimeSlotsRange**
  - **timeSlotCount**
  - **timeSlotNumber**
- **tags -** специальные метки (теги) компьютера, на который устанавливается интеракт, например, VDI. В дальнейшем они используются при обогащении данных в процессе инвентаризации.

Пример заполнения для ОС Windows

```
{
  "commonDirectory": "C:\\ProgramData\\discovery-platform\\interact",
  "companyGuid": "NNNNNNNN-NNNN-NNNN-NNNN-NNNNNNNNNNNN",
  "transactionPackagesLimit": 5,
  "transactionPackagesTimeout": 60,
  "syncModelInTransaction" : 5,
  "selfOpenedTcpPorts": {
    "portRange": {
      "first": 5078,
      "last": 5079
    }
  },
  "selfOpenedUdpPort":
  {
      "isNeedOpenUdpServerPort": false,
      "port" : 5090
  },
  "webSocket":
  {
      "port" : 30001
  },
  "httpServer":
  {
    "host": "0.0.0.0",
```

```
    "port": 5058,
    "sslCertificateKey" : "etc\\rootCA.key",
    "sslCertificate" : "etc\\rootCA.pem"
  },
  "mainServer": {
    "serverHost": "127.0.0.1",
    "serverPort": 5087,
    "serverPath": "/api/Interact/TaskData/TransactionsIsReady"
  },
  "logger": {
    "verbose": 0,
    "directory": "logs",
    "sink": "FileTxt",
    "maxFiles": 25
  },
  "timeDivision": {
    "allocation": "dynamic",
    "distributionAlgorithm": "Random",
    "frameLength": 120,
    "heartBeatTimeSlotsRange": [
        0,
        1200
    ],
    "timeSlotCount": 1200,
    "timeSlotNumber": -1
  },
  "tags": [
    "VDI"
  ]
}
```

5.5. Файл конфигурации агента: config_agent.json

Основные характеристики:

Назначение	Содержит настройки для агента
Наименование	**config_agent.json**
Расположение по умолчанию	Для ОС Windows в зависимости от типа агента: - C:\Program Files\Discovery.Agent.V2 - C:\Program Files\Discovery.Agent.Light.V2 Для ОС Linux: /ect/discovery-platform/

Параметры:

- **commonDirectory** – рабочий каталог агента, укажите путь сохранения временных данных

при инвентаризации в формате соответствующей операционной системы, например:

- Для Windows: **C:\\ProgramData\\Discovery.Agent.V2** или **C:\\ProgramData\\Discovery.Agent.Light.V2** в зависимости от версии агента.
- Для Linux: **/var/tmp/discovery-platform/agent/**

- **companyGuid** – GUID ключ **вашей компании** (вида NNNNNNNN-NNNN-NNNN-NNNN-NNNNNNNNNNNN), который вы получили при покупке системы.
- **interactOrParentAgentAddress** – данные родительского элемента (агента или интеракта), который должен получать данные от настраиваемого агента:
  - **host** - IP адрес.
  - **port** - открытый порт (по умолчанию для Interact: 5078, для агента-родителя: 5088).
- **selfOpenedTcpPorts**:
  - **isNeedOpenTcpServerPorts** – признак открытия портов агента:
    - **true** – порты открыты, агент может получать данные от других агентов, то есть участвует в цепочке агентов.
    - **false** – порты закрыты, в этом случае агент является конечной точкой и на него не могут передавать данные другие агенты.
  - **portRange** – начальный (first) и конечный (last) порты настраиваемого, открытые на прослушивание. Испольльзуется, если агент может получать данные от других агентов (self_port_is_open=true), то есть участвует в цепочке агентов. В ином случае игнорируется, даже если заполнен. Проверьте, что указанные порты не заняты. Например, если сервер и агент установлены на одном компьютере, и вы укажите порт, который занят сервером, то агент не будет работать корректно.
- **threadsCount –** количество потоков, используемых при сетевой инвентаризации.
- **timeDivision**:
  - **allocation** – определяет источник настроек timeDivision и может быть равен:
    - **static** – использовать текущие настройки timeDivision,
    - **dynamic** – запрашивать параметры у родителя, а именно агента или

интеракта.

- **distributionAlgorithm –** алгоритм выдачи тайм-слотов.
- **frameLength** – количество секунд, через которые агент передает свой статус интеракту.
- **heartBeatTimeSlotsRange –** интервал тайм слотов, которые можно раздавать дочерним элементам.
- **timeSlotCount –** количество тайм слотов в периоде.
- **timeSlotNumber –** номер тайм-слота, который раздает родитель.

- **logger** – настройки логов:
  - **verbose** - глубина логирования, варианты значений:
    - 0 - Trace
    - 1 - Debug
    - 2 - Info
    - 3 - Warning
    - 4 - Error
    - 5 - Critical
  - **directory** – путь к директории, в которую сохраняются логи в формате соответствующей операционной системы, например:
    - Для Windows: logs
    - Для Linux: /var/log/discovery-platform/agent
  - **maxFilesCount** – максимальное количество файлов с логами в директории.
  - **sink** – "FileTxt", куда выводить логи.
  - **ttl** – максимальное время жизни файлов логов в часах.
- **tags** – специальные метки (теги) компьютера, на который устанавливается агент (если есть необходимость), например, VDI. В дальнейшем они используются при обогащении данных в процессе инвентаризации.

Пример заполнения для ОС Windows

```
{
  "commonDirectory": "C:\\ProgramData\\Discovery.Agent.V2",
  "companyGuid": "NNNNNNNN-NNNN-NNNN-NNNN-NNNNNNNNNNNN",
  "interactOrParentAgentAddress": {
    "host": "127.0.0.1",
    "port": 5078,
    "udpPort": 5078
  },
  "selfOpenedTcpPorts": {
    "portRange": {
      "first": 5088,
      "last": 5089
    },
    "isNeedOpenTcpServerPorts": false
  },
  "logger": {
    "verbose": 0,
    "directory": "logs",
    "sink": "FileTxt",
    "maxFilesCount": 25,
    "maxFilesSize": 256,
    "ttl": 0
  },
  "tags": [
    "",
    "",
    ""
  ],
  "timeDivision": {
    "allocation": "dynamic",
    "distributionAlgorithm": "RoundRobin",
    "frameLength": 120,
    "timeSlotCount": 1200,
    "heartBeatTimeSlotsRange": [
      0,
      1199
    ],
    "timeSlotNumber": -1
  },
  "selfOpenedUdpPort": {
    "isNeedOpenUdpServerPort": false,
    "port": 5078
  }
}
```

5.6. Файл конфигурации nginx: config.json

Основные характеристики:

Назначение	Содержит настройки портов **nginx** для web-интерфейса
Наименование	**config.json**
Расположение по умолчанию	В директории установленного **nginx** /www/html/config.json

Параметры:

- **BASE_API_PORT** - порт для работы с сервером.
- **READER_API_PORT** - порт для работы с модулем Reader.
- **WRITER_API_PORT** - порт для работы с модулем Writer.
- **DEFAULT_INSTACE** - варианты значений:
  - **BaseInstance** - служебная настройка, не меняйте ее значение.
- **FLAG** - служебная настройка, не меняйте ее значение.

Пример заполнения

```
{
  "BASE_API_PORT": "5047",
  "READER_API_PORT": "5087",
  "WRITER_API_PORT": "5097",
  "DEFAULT_INSTACE": "BaseInstance",
  "FLAG":""
}
```

5.7. Файл конфигурации модуля Writer: WriterConfig.json

Основные характеристики:

Назначение	Содержит настройки Discovery Writer
Наименование	**WriterConfig.json**
Расположение по умолчанию	Для ОС Windows: C:\Program Files\Discovery Server\Server\Writer\Settings Для ОС Linux: /opt/Discovery.Writer/Settings/

Параметры:

- **Guid** - уникальный идентификатор конфигурации.
- **CommonDirectory** - адрес директории, в которую сохраняются данные модуля, включая транзакции, логи (**совпадает с директорией Reader**).

Формат зависит от операционной системы.

  - Пример для Windows: **C:\\ProgramData\\Discovery.Reader**
  - Пример для Linux: **/home/itmanr** или **/var/tmp/discovery-reader**

- **WriterAddress** - порт, на котором работает Writer.
- **CertificateFileName** - название файла SSL сертификата.
- **CertificateKeyFileName** - название файла ключа SSL.
- **X509ContentType** - формат сертификата x509.
- **EnKey** - определяет шифрование данных:
  - **true** - включает полное шифрование данных,
  - **false** - отключает шифрование (файлы будут расшифрованы).
- **VdiSyncPrefix** - тег, который будет указываться при настройке обогащения в инвентаризации, например: VDI_

Пример заполнения для ОС Windows

```
{
  "Guid": "NNNN-NNNN-NNNN-NNNN-NNNNNNNNNNNN",
  "CommonDirectory": "C:\\ProgramData\\Discovery.Reader",
  "WriterAddress": "https://*:5097",
  "CertificateFileName": "rootCA.pem",
  "CertificateKeyFileName": "rootCA.key",
  "X509ContentType": "Pkcs12",
  "EnKey": true,
  "VdiSyncPrefix": "VDI_"
}
```