Оглавление

1 Общие сведения о документе

Информация о документе:

Наименование документа:	Руководство по установке системы для администратора		
Номер версии документа:	**Версия 5.0**		
Документ подготовил:	**Владислав Никитин** **Камиля Фахрутдинова**	**Дата создания версии:**	**13.10.2021**

Информация о версиях документа:

№ вер	Дата создания версии	Кем создана версия	Краткое описание причин и состава изменений документа	Наименование файла
1.0	**18.05.2021**	**Владислав Никитин** **Камиля Фахрутдинова**		**Руководство по установке системы для администратора**
2.0	**20.07.2021**	**Владислав Никитин** **Камиля Фахрутдинова**		**Руководство по установке системы для администратора**
3.0	**11.08.2021**	**Владислав Никитин** **Камиля Фахрутдинова**		**Руководство по установке системы для администратора**
4.0	**06.09.2021**	**Владислав Никитин** **Камиля Фахрутдинова**		**Руководство по установке системы для администратора**
5.0	**13.10.2021**	**Владислав Никитин** **Камиля Фахрутдинова**		**Руководство по установке системы для администратора**

2 Введение

Настоящий документ является руководством по установке программы для ЭВМ «Система IT Infrastructure Manager (ITIM) 2021» (Далее - Система) для администратора.

2.1 Требования к программному обеспечению

Требования к программному обеспечению серверной части:

Компонент	**Конфигурация**
Операционная система	AstraLinux, АльтLinux, РедОС, RosaLinux
СУБД	Mysql
Общесистемное ПО	• Ansible semaphore • Zabbix • Gitlab

Требования к аппаратному обеспечению серверной части:

Компонент	**Минимальная конфигурация**
Процессор	4 core CPU 2,2Ghz+
Оперативная память	8 Gb
Жесткий диск	200Gb

2.1.1 Клиентская часть

Для работы с Системой IT Infrastructure Manager (ITIM) 2021 рабочие станции пользователей должны удовлетворять следующим минимальным требованиям:

Компонент	**Минимальная конфигурация**
Процессор	Pentium 4
Оперативная память	512 Mb
Жесткий диск	350 Mb
Браузер	Firefox, Chrome

Для развертывания системы необходимо, чтобы были свободными следующие порты: 80, 3000, 3022, 3080, 3306, 5001, 10051, 53659, 53660.

3 Установка системы

3.1 Установка системы на примере Astra-Linux

1. Скачайте пакет itim-astra.deb, в консоли перейдите в папку со скачанным архивом. Например, с помощью команды с cd.
2. Запустите установку, выполнив команду:

   sudo apt-get install ./itim-astra.deb

3. Во время установки необходимо ввести ip-адрес и порт узла, на котором устанавливается система.

```
(Чтение базы данных … на данный момент установлено 112477 файлов и каталогов.)
Подготовка к распаковке /home/developer/itim-astra.deb …
Распаковывается itim (1.0) …
Настраивается пакет itim (1.0) …
Выполнение posinst
Введите IP адрес установки системы: 10.10.167.236
Введите порт для приложения ITIM: 82
```

Рис 1. Пример ввода IP адреса и порта.

4. После завершения установки убедитесь в наличии следующих контейнеров: mysql, itim-web, zabbix-web, zabbix-server, semaphore, gitlab (их статус должен находиться в состоянии UP), командой:

   sudo docker ps -a

```
STATUS                          PORTS                                          NAMES
Up 22 seconds                   0.0.0.0:82->80/tcp                             itim-web
Up 26 seconds                   0.0.0.0:53659->8080/tcp, 0.0.0.0:53660->8443/tcp   zabbix-web
Up 31 seconds                   0.0.0.0:10051->10051/tcp                       zabbix-server
Up 34 seconds (health: starting) 443/tcp, 0.0.0.0:3022->22/tcp, 0.0.0.0:3080->80/tcp  gitlab
Up 19 seconds                   0.0.0.0:49562->3000/tcp                        semaphore
Up About a minute (healthy)     0.0.0.0:3306->3306/tcp, 33060/tcp              mysql
```

Рис 2. Пример вывода команды docker ps -a

5. Убедитесь в запущенном (активном) состоянии сервиса itim.service командой:

   systemctl status itim

```
● itim.service - ITIM .net API running on Ubuntu.
   Loaded: loaded (/etc/systemd/system/itim.service; enabled; vendor preset: enabled)
   Active: active (running) since Thu 2021-07-22 18:14:34 MSK; 44s ago
 Main PID: 24367 (ITEM Zabbix)
    Tasks: 18 (limit: 4915)
   Memory: 99.6M
      CPU: 2.197s
   CGroup: /system.slice/itim.service
           └─24367 /srv/itim/publish/ITEM Zabbix --urls=http://10.10.167.236:5001/
developer@ASTRSLKI:~$
```

Рис 3. Пример вывода команды systemctl status itim

3.2 Установка системы на примере РедОС

1. Скачайте пакет itim-1.0-1.x86_64.rpm, в консоли перейдите в папку со скачанным архивом. Например, с помощью команды с cd.
2. Запустите установку, выполнив команду:

sudo yum install ./itim-1.0-1.x86_64.rpm

```
[root@localhost developer]# yum install ./itim-1.0-1.x86_64.rpm
```

Рис 4. Пример ввода команды

3. После завершения работы yum менеджера, запустите скрипт, конфигурации системы, выполнив команду: введите ip-адрес и порт узла, на котором устанавливается система.

sudo /tmp/itim/deploy-RedOS.sh

```
[root@localhost developer]# /tmp/itim/deploy-RedOS.sh
Выполнение posinst
Введите IP адрес установки системы: 10.10.167.249
Введите порт для приложения ITIM: 8282
```

Рис 5. Пример ввода команды

4. После завершения установки убедитесь в наличии следующих контейнеров: mysql, itim-web, zabbix-web, zabbix-server, semaphore, gitlab (их статус должен находиться в состоянии UP), командой:

sudo docker ps -a

```
STATUS                          PORTS                                                NAMES
Up 22 seconds                   0.0.0.0:82->80/tcp                                   itim-web
Up 26 seconds                   0.0.0.0:53659->8080/tcp, 0.0.0.0:53660->8443/tcp     zabbix-web
Up 31 seconds                   0.0.0.0:10051->10051/tcp                             zabbix-server
Up 34 seconds (health: starting) 443/tcp, 0.0.0.0:3022->22/tcp, 0.0.0.0:3080->80/tcp gitlab
Up 19 seconds                   0.0.0.0:49562->3000/tcp                              semaphore
Up About a minute (healthy)     0.0.0.0:3306->3306/tcp, 33060/tcp                    mysql
```

Рис 6. Пример вывода команды docker ps -a

5. Убедитесь в запущенном (активном) состоянии сервиса itim.service командой:

sudo systemctl status itim

```
developer-Virtual-Machine developer # systemctl status itim
● itim.service - ITIM .net API running on Ubuntu.
   Loaded: loaded (/etc/systemd/system/itim.service; enabled; vendor preset: disabled)
   Active: active (running) since Вт 2021-08-24 16:47:22 MSK; 14s ago
 Main PID: 19719 (ITEM Zabbix)
    Tasks: 18 (limit: 512)
   CGroup: /system.slice/itim.service
           └─19719 /srv/itim/publish/ITEM Zabbix --urls=http://10.10.167.238:5001/

авг 24 16:47:22 developer-Virtual-Machine systemd[1]: Started ITIM .net API running on Ubuntu..
авг 24 16:47:25 developer-Virtual-Machine dotnet-example[19719]: 2021-08-24 16:47:25.8891|INFO|Microsoft.Hosting
авг 24 16:47:25 developer-Virtual-Machine dotnet-example[19719]: 2021-08-24 16:47:25.9894|INFO|Microsoft.Hosting
авг 24 16:47:25 developer-Virtual-Machine dotnet-example[19719]: 2021-08-24 16:47:25.9931|INFO|Microsoft.Hosting
авг 24 16:47:25 developer-Virtual-Machine dotnet-example[19719]: 2021-08-24 16:47:25.9931|INFO|Microsoft.Hosting
```

Рис 7. Пример вывода команды systemctl status itim

3.3 Установка системы на примере Alt

1. Скачайте пакет itim-1.0-1.x86_64.rpm, в консоли перейдите в папку со скачанным архивом. Например, с помощью команды cd.
2. Запустите установку, выполнив команду:

sudo apt-get install ./itim-1.0-1.x86_64.rpm

```
[root@ASTRNSTII ~]# sudo apt-get install ./itim-1.0-1.x86_64.rpm
```

Рис 8. Пример ввода команды

3. После завершения работы apt-get менеджера, запустите скрипт, конфигурации системы, выполнив команду: введите ip-адрес и порт узла, на котором устанавливается система.

sudo /tmp/itim/deploy.sh

```
[root@ASTRNSTII ~]# /tmp/itim/deploy.sh
Выполнение posinst
Введите IP адрес установки системы: 10.10.167.275
```

Рис 9. Пример ввода команды

4. После завершения установки убедитесь в наличии следующих контейнеров: mysql, itim-web, zabbix-web, zabbix-server, semaphore, gitlab (их статус должен находиться в состоянии UP), командой:

sudo docker ps -a

```
STATUS                          PORTS                                                NAMES
Up 22 seconds                   0.0.0.0:82->80/tcp                                   itim-web
Up 26 seconds                   0.0.0.0:53659->8080/tcp, 0.0.0.0:53660->8443/tcp     zabbix-web
Up 31 seconds                   0.0.0.0:10051->10051/tcp                             zabbix-server
Up 34 seconds (health: starting) 443/tcp, 0.0.0.0:3022->22/tcp, 0.0.0.0:3080->80/tcp gitlab
Up 19 seconds                   0.0.0.0:49562->3000/tcp                              semaphore
Up About a minute (healthy)     0.0.0.0:3306->3306/tcp, 33060/tcp                    mysql
```

Рис 10. Пример вывода команды docker ps -a

5. Убедитесь в запущенном (активном) состоянии сервиса itim.service командой:

systemctl status itim

```
[root@ASTRNSTII ~]# systemctl status itim.service
● itim.service - ITIM .net API running on Ubuntu.
     Loaded: loaded (/etc/systemd/system/itim.service; enabled; vendor preset: disabled)
     Active: active (running) since Mon 2021-08-30 17:29:12 MSK; 21h ago
   Main PID: 2800 (ITEM Zabbix)
      Tasks: 18 (limit: 4693)
     Memory: 41.0M
     CGroup: /system.slice/itim.service
             └─2800 /srv/itim/publish/ITEM Zabbix --urls=http://10.10.167.27:5001/

авг 30 17:29:12 ASTRNSTII systemd[1]: Started ITIM .net API running on Ubuntu..
авг 30 17:29:37 ASTRNSTII dotnet-example[2800]: 2021-08-30 17:29:36.8374|INFO|Microsoft.Hosting.Life
авг 30 17:29:37 ASTRNSTII dotnet-example[2800]: 2021-08-30 17:29:37.2134|INFO|Microsoft.Hosting.Life
авг 30 17:29:37 ASTRNSTII dotnet-example[2800]: 2021-08-30 17:29:37.3612|INFO|Microsoft.Hosting.Life
авг 30 17:29:37 ASTRNSTII dotnet-example[2800]: 2021-08-30 17:29:37.3612|INFO|Microsoft.Hosting.Life
lines 1-14/14 (END)
```

Рис 11. Пример вывода команды systemctl status itim

3.4 Установка системы на примере Rosa

1. Скачайте пакет itim-Rosa-1.0-1.x86_64.rpm, в консоли перейдите в папку со скачанным архивом. Например, с помощью команды с cd.
2. Запустите установку, выполнив команду:

sudo urpmi ./itim-Rosa-1.0-1.x86_64.rpm

```
developer@developer-Virtual-Machine ~ $ sudo urpmi ./itim-1.0-1.x86_64.rpm
```

Рис 12. Пример ввода команды

3. После завершения работы urpmi менеджера, запустите скрипт, конфигурации системы, выполнив команду: введите ip-адрес и порт узла, на котором устанавливается система.

sudo /tmp/itim/deploy.sh

```
developer@developer-Virtual-Machine ~ $ sudo /tmp/itim/deploy.sh
Выполнение posinst
Введите IP адрес установки системы: 10.10.167.238
Введите порт для приложения ITIM: 8282_
```

Рис 13. Пример ввода команды

4. После завершения установки убедитесь в наличии следующих контейнеров: mysql, itim-web, zabbix-web, zabbix-server, semaphore, gitlab (их статус должен находиться в состоянии UP), командой:

sudo docker ps -a

```
STATUS                          PORTS                                                    NAMES
Up 22 seconds                   0.0.0.0:82->80/tcp                                       itim-web
Up 26 seconds                   0.0.0.0:53659->8080/tcp, 0.0.0.0:53660->8443/tcp         zabbix-web
Up 31 seconds                   0.0.0.0:10051->10051/tcp                                 zabbix-server
Up 34 seconds (health: starting) 443/tcp, 0.0.0.0:3022->22/tcp, 0.0.0.0:3080->80/tcp     gitlab
Up 19 seconds                   0.0.0.0:49562->3000/tcp                                  semaphore
Up About a minute (healthy)     0.0.0.0:3306->3306/tcp, 33060/tcp                        mysql
```

Рис 14. Пример вывода команды docker ps -a

5. Убедитесь в запущенном (активном) состоянии сервиса itim.service командой:

sudo systemctl status itim

```
[root@ASTRNSTII ~]# systemctl status itim.service
● itim.service - ITIM .net API running on Ubuntu.
     Loaded: loaded (/etc/systemd/system/itim.service; enabled; vendor preset: disabled)
     Active: active (running) since Mon 2021-08-30 17:29:12 MSK; 21h ago
   Main PID: 2800 (ITEM Zabbix)
      Tasks: 18 (limit: 4693)
     Memory: 41.0M
     CGroup: /system.slice/itim.service
             └─2800 /srv/itim/publish/ITEM Zabbix --urls=http://10.10.167.27:5001/

авг 30 17:29:12 ASTRNSTII systemd[1]: Started ITIM .net API running on Ubuntu..
авг 30 17:29:37 ASTRNSTII dotnet-example[2800]: 2021-08-30 17:29:36.8374|INFO|Microsoft.Hosting.Life
авг 30 17:29:37 ASTRNSTII dotnet-example[2800]: 2021-08-30 17:29:37.2134|INFO|Microsoft.Hosting.Life
авг 30 17:29:37 ASTRNSTII dotnet-example[2800]: 2021-08-30 17:29:37.3612|INFO|Microsoft.Hosting.Life
авг 30 17:29:37 ASTRNSTII dotnet-example[2800]: 2021-08-30 17:29:37.3612|INFO|Microsoft.Hosting.Life
lines 1-14/14 (END)
```

Рис 15. Пример вывода команды systemctl status itim

4 Первый вход в систему

1. Используя браузер, перейдите по адресу, который был задан при установке

Рис.16 Пример ввода адреса

2. В случае использования порта, отличного от 80, добавьте в конец адреса номер порта, после символа «:»

Рис.17 Пример ввода адреса с портом

3. Используя логин и пароль *itim / itim* , войдите в систему

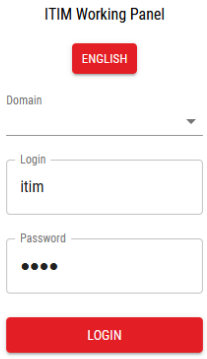

Рис.18 Вид страницы входа в систему

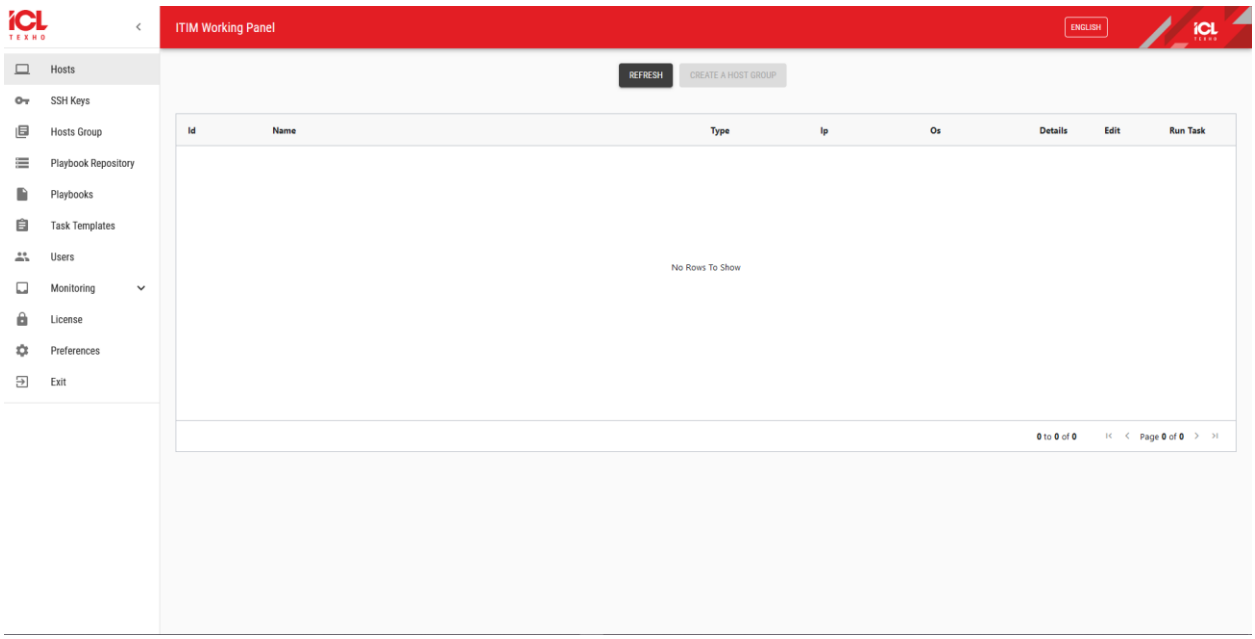

Рис.19 Вид главной страницы системы

5 Добавление пользователей с привязкой к LDAP аккаунту

Для того, чтобы добавить пользователя с привязкой к LDAP-аккаунту, предварительно необходимо настроить LDAP окружение.

4. Запустите терминал, введите команду, для редактирования файла конфигурации:

nano /srv/itim/publish/appsettings.json

```
root@ASTRSLKI:~# nano /srv/itim/publish/appsettings.json
```

Рис.20 Пример ввода команды для редактирования файла конфигурации.

5. Измените параметры в пункте LDAPs, введите адрес узла LDAP сервера, порт, домен, и учётные данные, для авторизации. Сохраните файл и выйдет из редактора.

```
  "LDAPs": [
    {
      "LDAPHost": "testrussiad252.testrussia.local",
      "LDAPPort": 389,
      "LDAPAccount": "testrussia\\SysAdmin",
      "LDAPPassword": "P@ssw0rd",
      "LDAPDomain": "testrussia.local"
    }
  ]
```

Рис.21 Пример заполнения параметров конфигурации LDAP

6. Убедитесь в правильном заполнении и сохранении конфигурации, введите команду:

cat /srv/itim/publish/appsettings.json | grep -i ldap

```
root@ASTRSLKI:~# cat /srv/itim/publish/appsettings.json | grep -i ldap
    "LDAPs": [
        "LDAPHost": "testrussiad252.testrussia.local",
        "LDAPPort": 389,
        "LDAPAccount": "testrussia\\SysAdmin",
        "LDAPPassword": "P@ssw0rd",
        "LDAPDomain": "testrussia.local"
```

Рис.22 Пример вывода команды cat.

7. Для внесения изменений необходимо перезапустить систему ITIM, для этого введите следующую команду:

sudo systemctl restart itim

8. Перейдите во вкладку пользователи, нажмите кнопку добавить пользователя. Укажите имя пользователя, существующего в LDAP сервере. Отметьте галочку “Привязать к LDAP”. Система произведет подключение к LDAP серверу, и привязку.

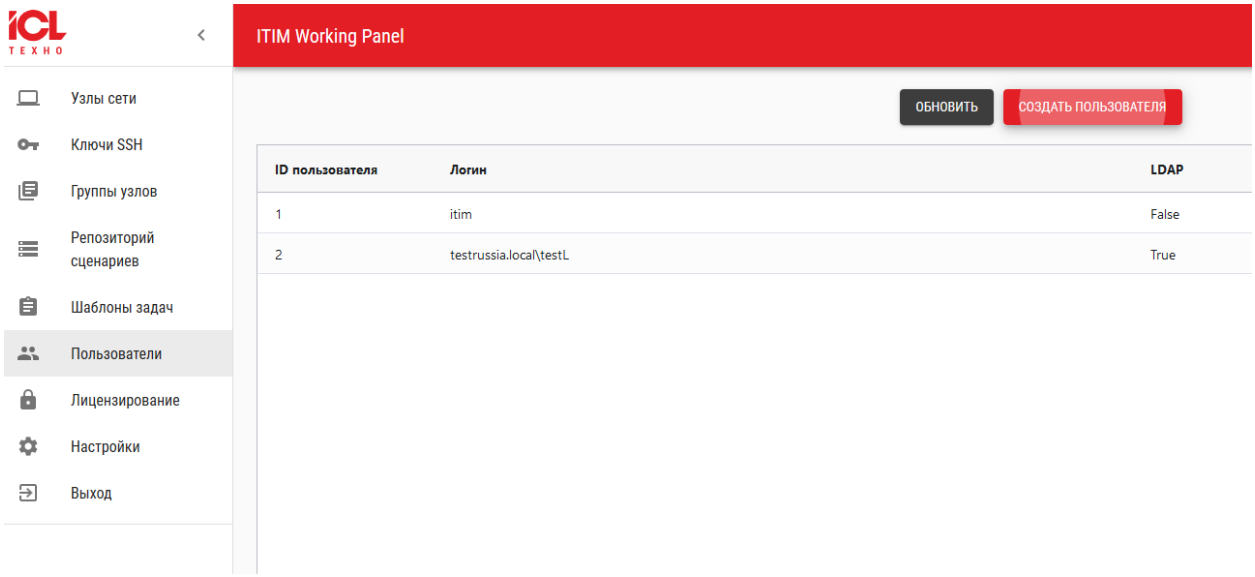

Рис.23 Создание нового пользователя

В случае, если LDAP-аккаунт отсутствует на LDAP-домене, то при создании пользователя в системе будет выводиться ошибка:

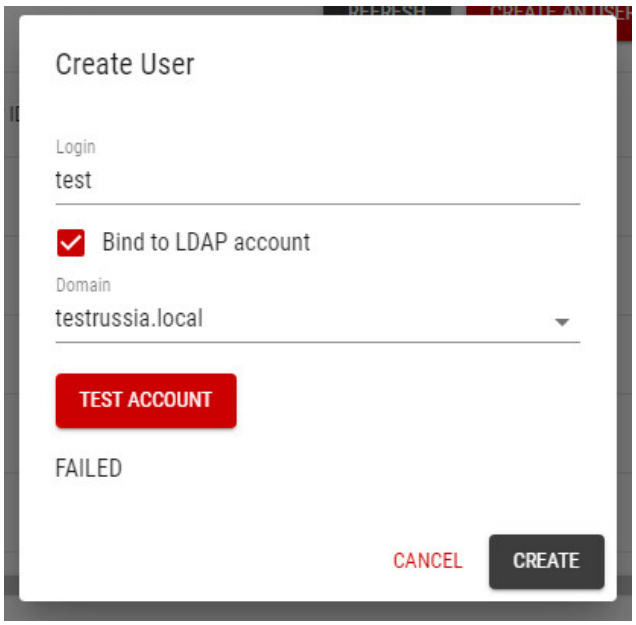

Рис.24 Создание нового пользователя

Если LDAP-аккаунт заведен, то при создании пользователя будет выведено сообщение “CHECKED SUCCESSFUL”

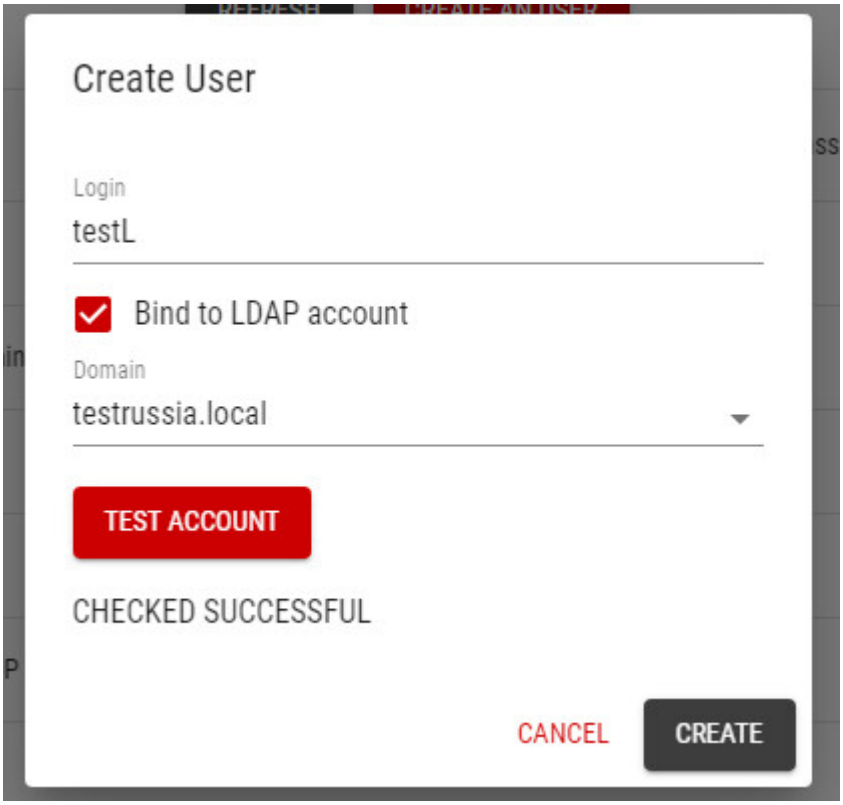

Рис.25 Создание нового пользователя с использование LDAP-аккаунта

Созданный пользователь с привязкой к LDAP-аккаунту может входить в систему через привязанную учетную запись LDAP:

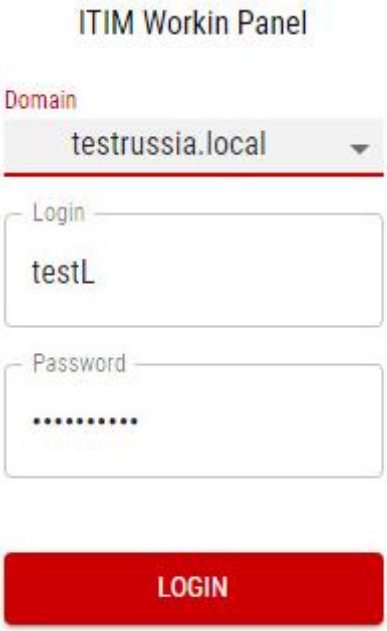

Рис.26 Вход через учетную запись LDAP

6 Настройка Git и репозитория

Перед настройкой Git убедитесь в готовности контейнера с gitlab, для этого введите команду:

sudo docker ps –a

```
STATUS                  PORTS                                                   NAMES
Up 7 minutes            0.0.0.0:8282->80/tcp                                    itim-web
Up 7 minutes            0.0.0.0:53659->8080/tcp, 0.0.0.0:53660->8443/tcp        zabbix-web
Up 7 minutes            0.0.0.0:10051->10051/tcp                                zabbix-server
Up 7 minutes (healthy)  443/tcp, 0.0.0.0:3022->22/tcp, 0.0.0.0:3080->80/tcp     gitlab
Up 7 minutes            0.0.0.0:49562->3000/tcp                                 semaphore
Up 9 minutes (healthy)  0.0.0.0:3306->3306/tcp, 33060/tcp                       mysql
```

Рис.27 Вывод информации о запущенных контейнерах

Статус готового контейнера к использованию обозначается как healthy, в случае starting, подождите некоторое время, после повторите команду.

1. Перейдите на страницу gitlab, указав IP-адрес хоста введенный при установке и порт 3080. Пример: http://10.10.167.27:3080, осуществите вход в систему под пользователем root пароль gitlabPassword

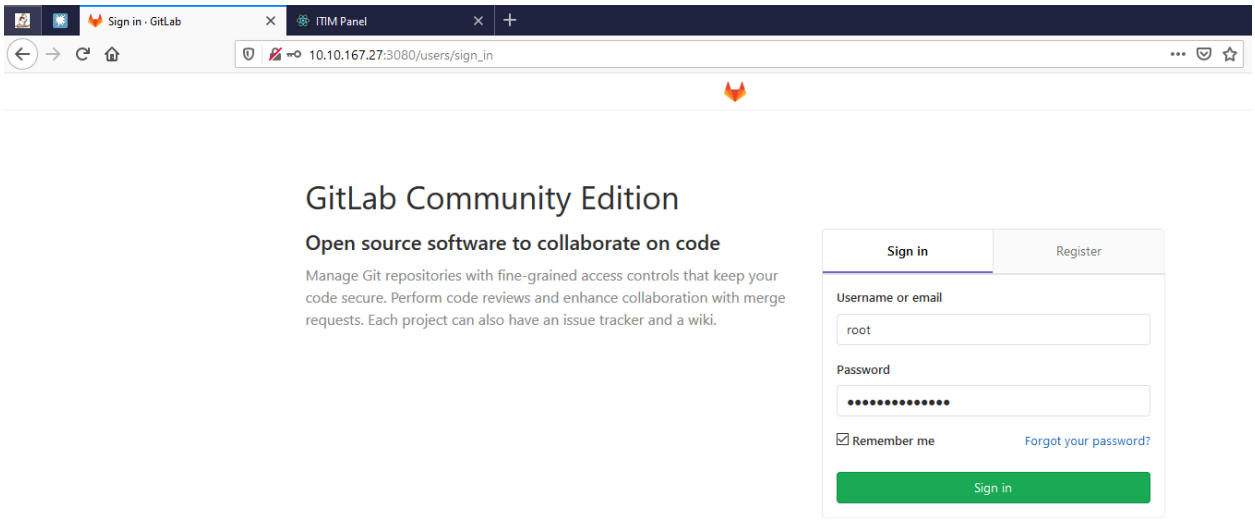

Рис.28 Вход в систему Gitlab

2. Затем необходимо зайти в меню настроек (иконка гаечного ключа), расположенное на верхней панели.

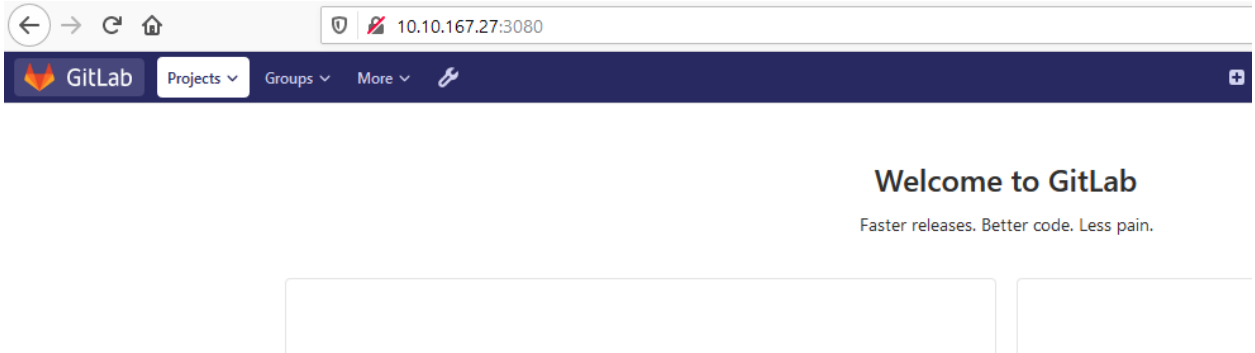

Рис.29 Открытие панели администратора

3. В появившимся боковом меню нажмите на вкладку «Settings», для перехода в настройки gitlab.

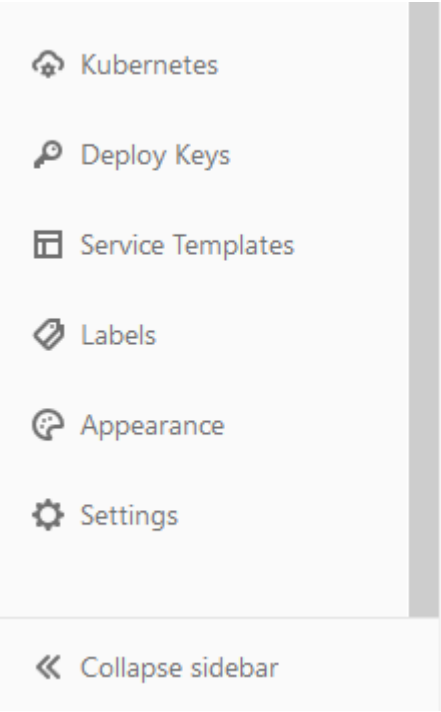

Рис.30 Боковое меню

4. В разделе настроек «Visibility and access controls» нажмите на кнопку “Expand”.

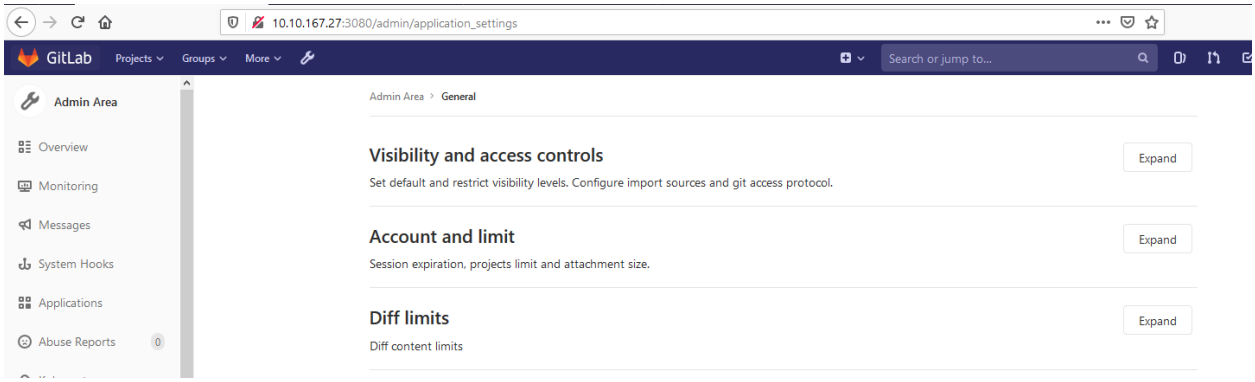

Рис.31 Настройки Gitlab

5. В раскрывшимся меню в пункте “Custom Git clone URL for HTTP(S)” введите IP-адрес хоста, введенный при установке, и порт 3080. Пример: http://10.10.167.27:3080. По окончанию ввода нажмите на кнопку “Save changes”.

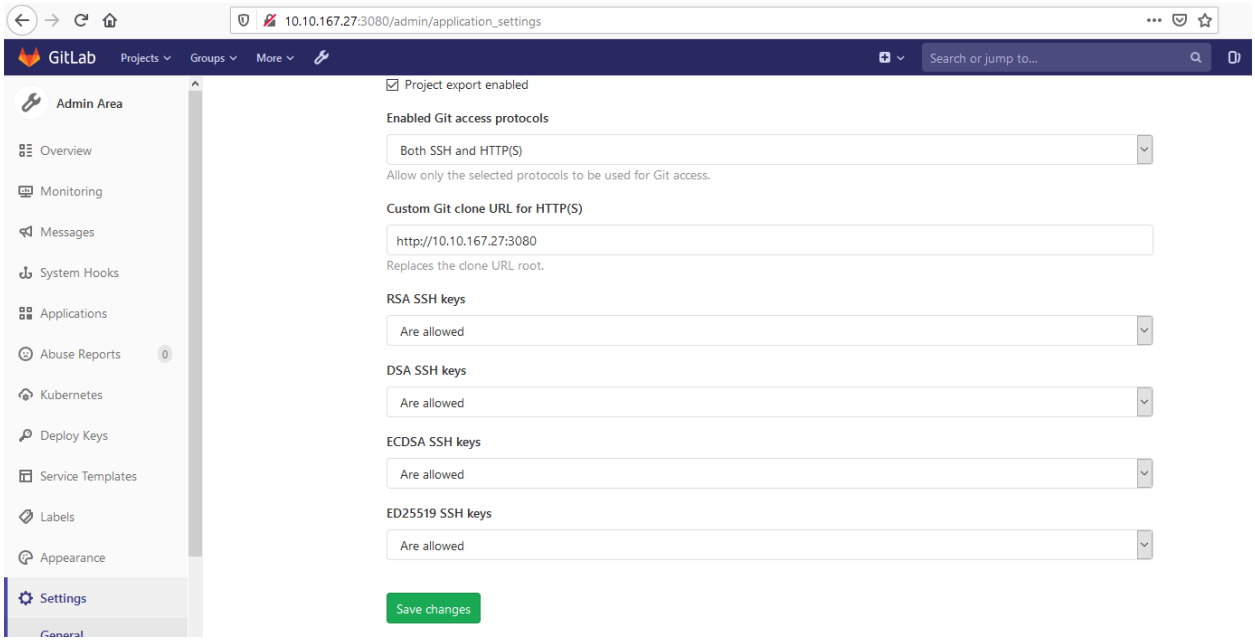

Рис.32 Настройки Gitlab

6. Создайте проект в git. Для этого необходимо нажать кнопку “+” и затем выбрать “New project”.

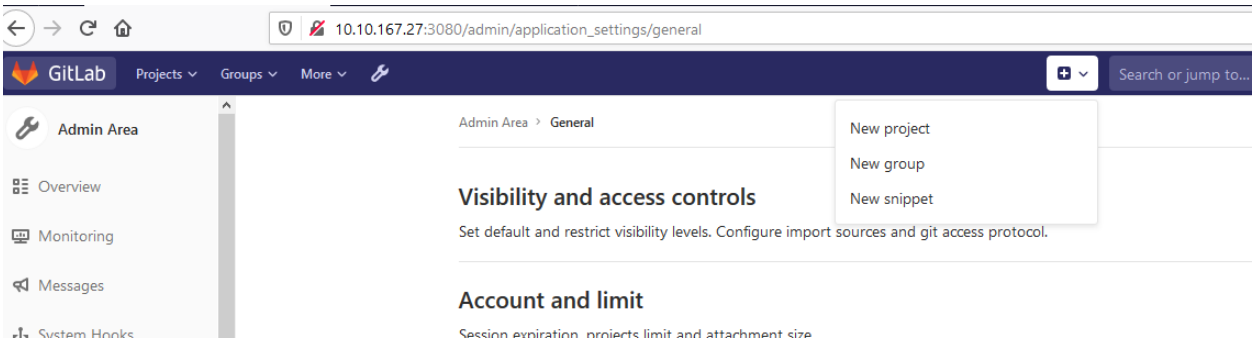

Рис.33 Создание нового проекта в git

7. Вам откроется окно с созданием проекта. Введите название проекта в поле «Project name». Выставите флаг «Visibility Level» в значение «Public». Далее нажмите на кнопку «Create Project».

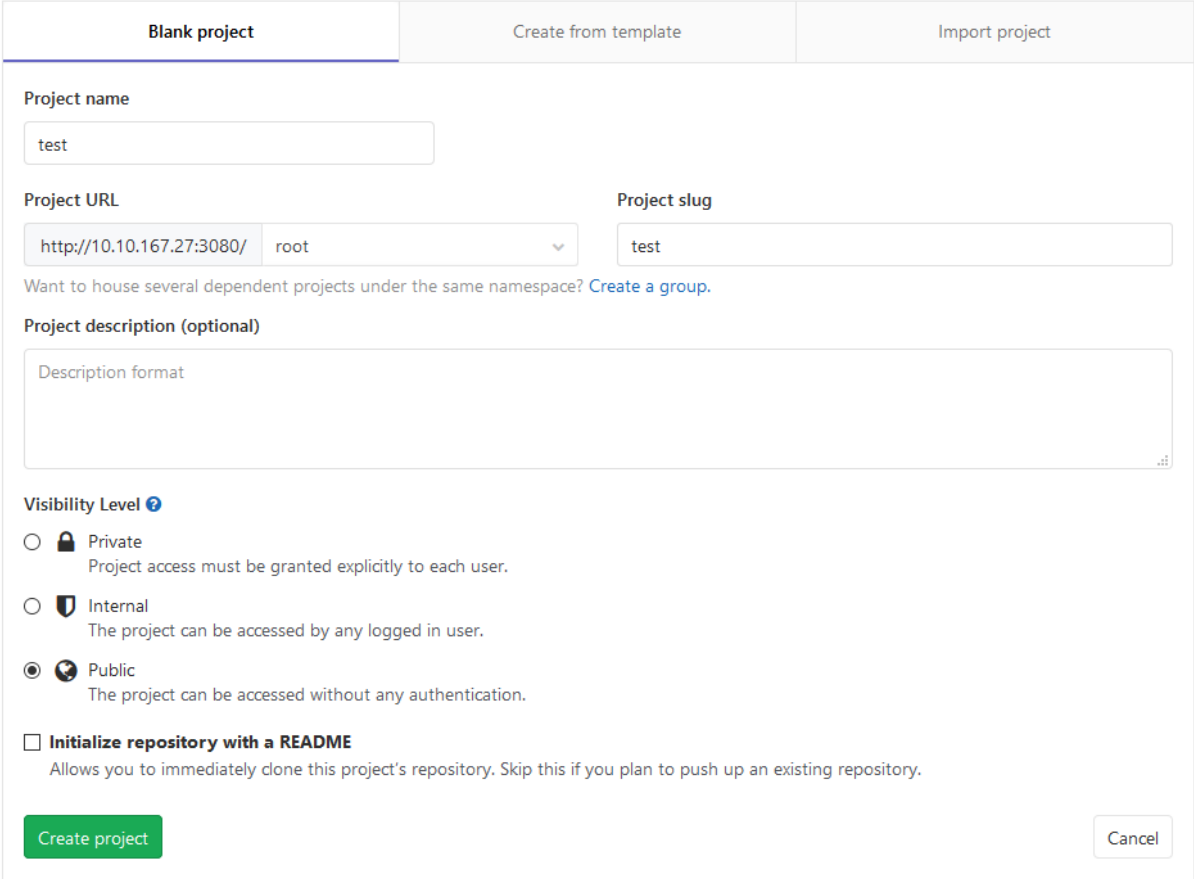

Рис.34 Окно с созданием нового проекта

8. Воспользуйтесь кнопкой “Clone” на главной странице проекта есть. При ее нажатии откроются ссылки для клонирования. Для добавления репозитория в систему ITIM понадобится ссылка HTTP.

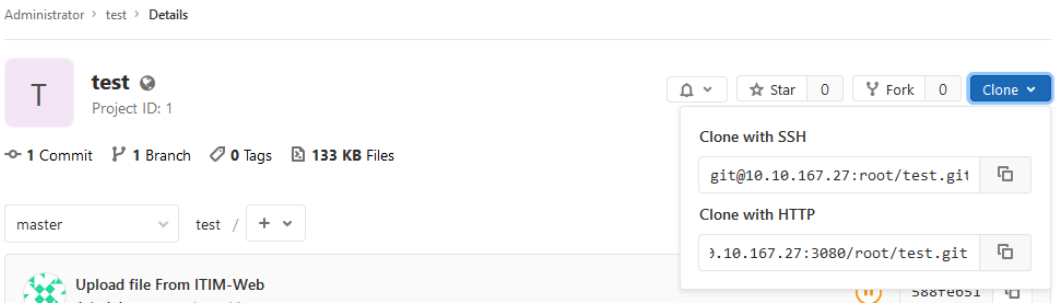

Рис.35 Ссылки для клонирования

9. Для добавления репозитория перейдите в ITIM, на вкладке “Playbook Repository” нажмите на кнопку “Add a new repository”.

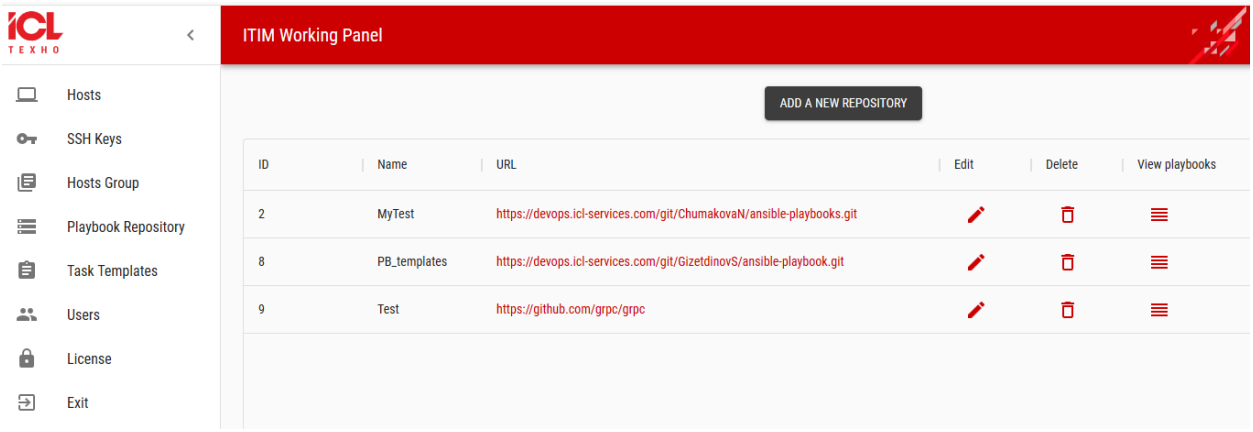

Рис.36 Вкладка “Playbook Repository”

10. В появившимся окне укажите имя репозитория, вставьте ссылку репозитория Git в поле URL, и выберите SSH ключ из списка. Нажмите кнопку «Save». В случае отсутствия ключа SSH, его нужно добавить, добавление ключей SSH описано в разделе 7.

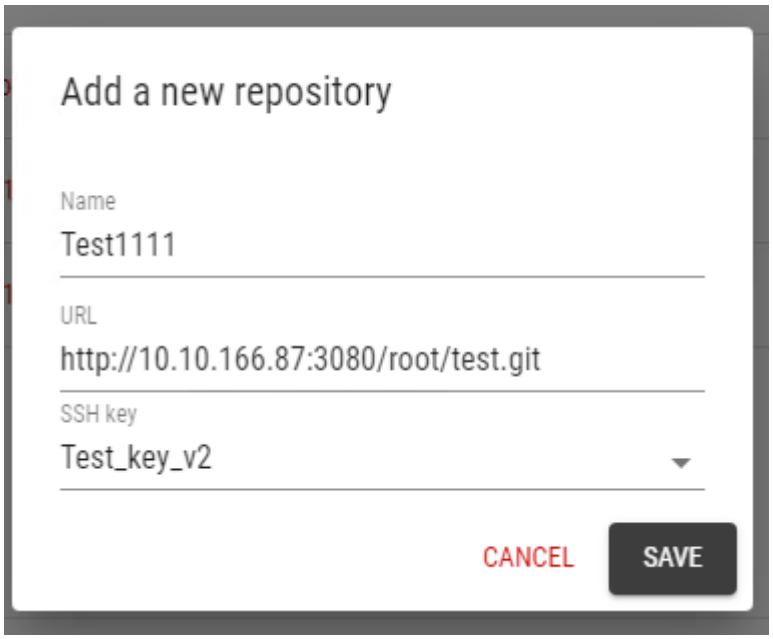

Рис.37 Создание нового репозитория в системе

6.1 Добавление готовых плейбуков

1. Перейдите в раздел «Playbooks» и нажмите на кнопку «Upload Playbook».

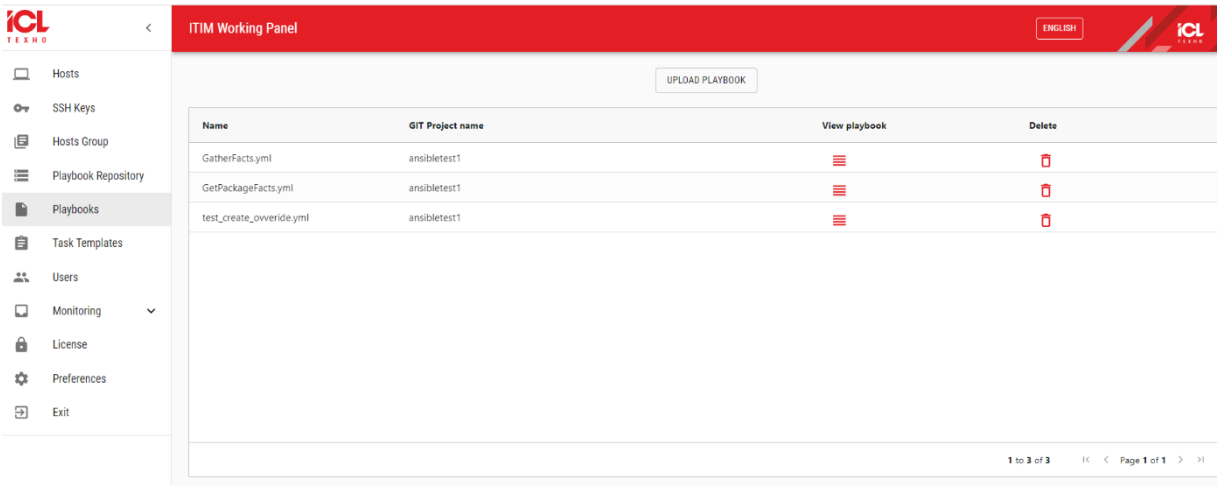

Рис.38 Добавление “Плейбука”

2. Далее в окне «Add playbook» выберите репозиторий и нажав на кнопку «Choose File and Upload» выберите нужный плейбук для добавления.

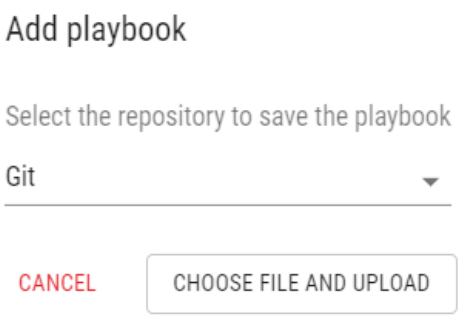

Рис.39 Выбор “Плейбука”

3. В комплекте поставляются два готовых плейбука:

   GatherFacts.yml – получение информации по аппаратной конфигурации хостов;

   GetPackageFacts.yml - получение информации по установленным на хосты пакетам.

7 Получение лицензии на программное обеспечение

Без лицензии функционал программного обеспечения значительно ограничен. Для получения лицензии в системе ITIM необходимо перейти во вкладку “License”.

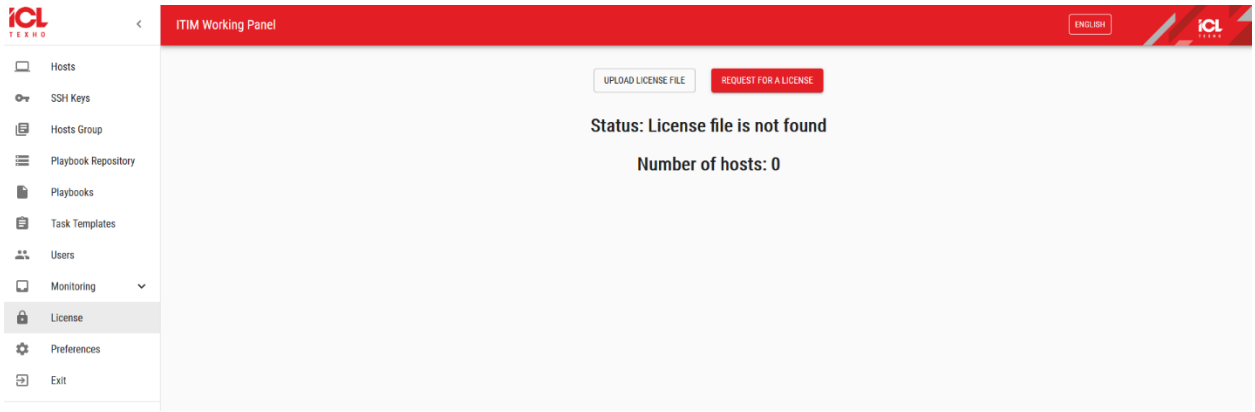

Рис.40 Вкладка “License”

1. Для получения файла запроса лицензии необходимо нажать кнопку “Request for a license”, начнется загрузка файла.

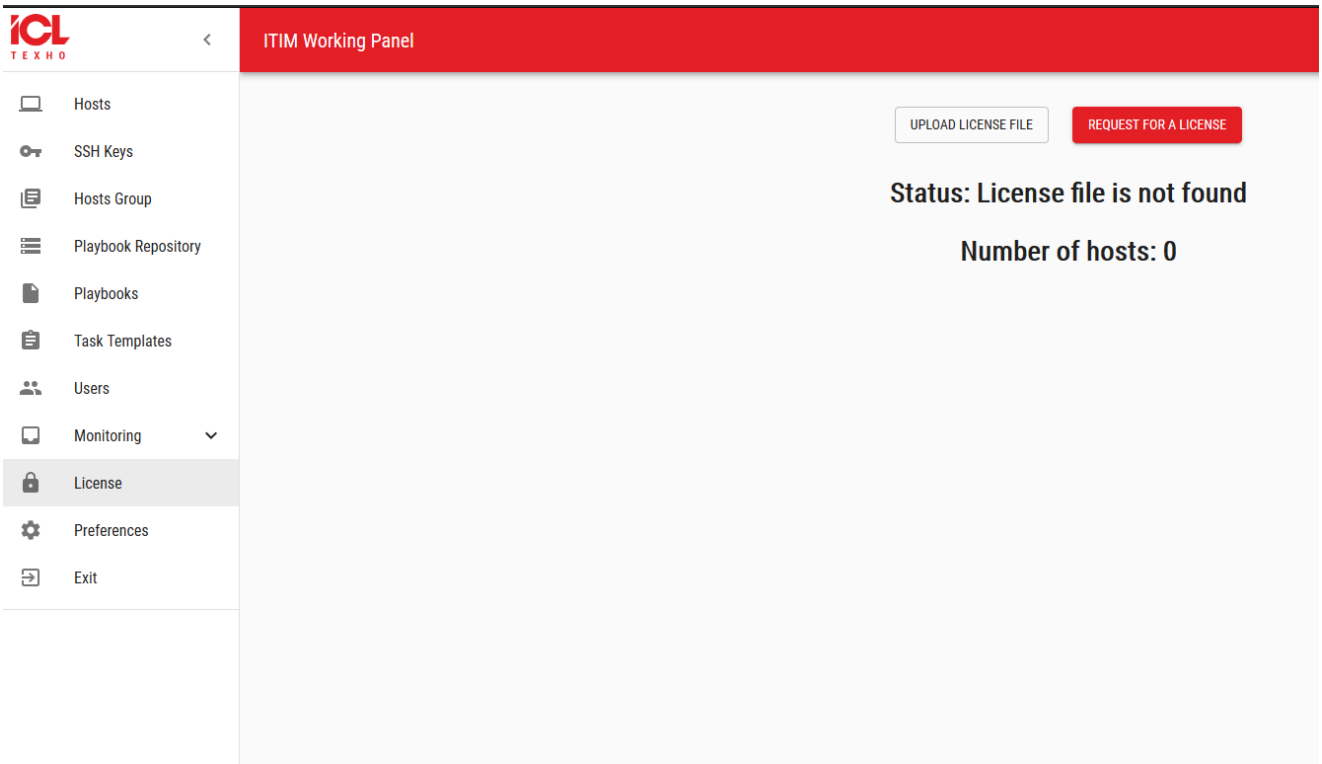

Рис.41 Вкладка “License”

2. После скачивания файла, передайте его компании, поставщику системы. В ответ вы получите файл лицензии.
3. На вкладке “License” нажмите на кнопку “Choose License File” и выберите полученный вами файл лицензии

8 Добавление ssh ключей

С сервера, на котором развернута система необходимо задать ключи на те машины, на которых есть потребность в запуске плейбуков:

1. Сгенерируйте SSH ключи с помощью следующей команды: *ssh-keygen*
2. Самый простой способ скопировать ключ - это использовать утилиту *ssh-copy-id.* Но для работы этого метода нужно иметь пароль доступа к машинам по SSH. Синтаксис команды: *ssh-copy-id username@remote_host*
3. Проверьте соединение по SSH с помощью команды, синтаксис которой выглядит следующим образом: *ssh username@remote_host*
4. При первом подключение введите пароль, далее после перехода на виртуальную машину необходимо выйти из нее (команда exit). И еще раз подключиться по ssh командой, синтаксис которой выглядит следующим образом: *ssh username@remote_host.*
5. В случае, если все шаги были выполнены корректно, пароль далее запрашиваться не будет.
6. Скопируйте публичный и приватный ключ и в системе «ITIM» на вкладке “SSH Keys”, нажмите на кнопку “Add a key”. Укажите название, публичный и приватный ключ и нажмите на кнопку “Save”.

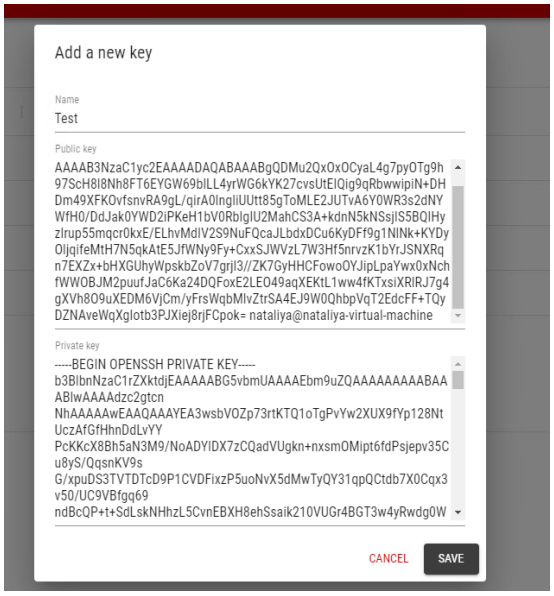

Рис.42 Добавление SSH ключа

9 Настройки обнаружения устройств

Обнаружения устройств и действия по их добавлению в систему выполняются, используя технологии Zabbix. В частности Zabbix API.

Система периодически сканирует указанные в правилах сетевого обнаружения диапазоны IP, пытаясь подключиться к Zabbix агенту по стандартному порту 10050 и получить значение ключа *system.uname*. Частота сканирования по каждому правилу составляет 20 мин, от момента создания или включения правила. Обратите внимание, что одно правило обнаружения всегда обрабатывается одним процессом обнаружения. Диапазон IP адресов не разбивается между несколькими процессами обнаружения. После обнаружения и добавления всех необходимых узлов рекомендуется отключить правило обнаружения, для уменьшения нагрузки на систему.

Когда правило будет добавлено, система автоматически запустит обнаружение и порождение событий, основанных на обнаружении, для дальнейшей их обработки (добавление, удаление узла).

Условия добавления новых узлов:

- сервис узла “Zabbix агента” в состоянии “доступен“.
- значение *system.uname* (ключ Zabbix агента) содержит “Linux” или “Windows”
- Время работы более 5 минут (300 секунд).

Узел будет удален из конфигурации, если сервис “Zabbix агент” 'Недоступен' на протяжении более чем 24 часов (86400 секунд).

9.1 Добавление правила обнаружения устройств

1. Для того чтобы настроить правило обнаружения сети, перейдите во вкладку *Preferences*

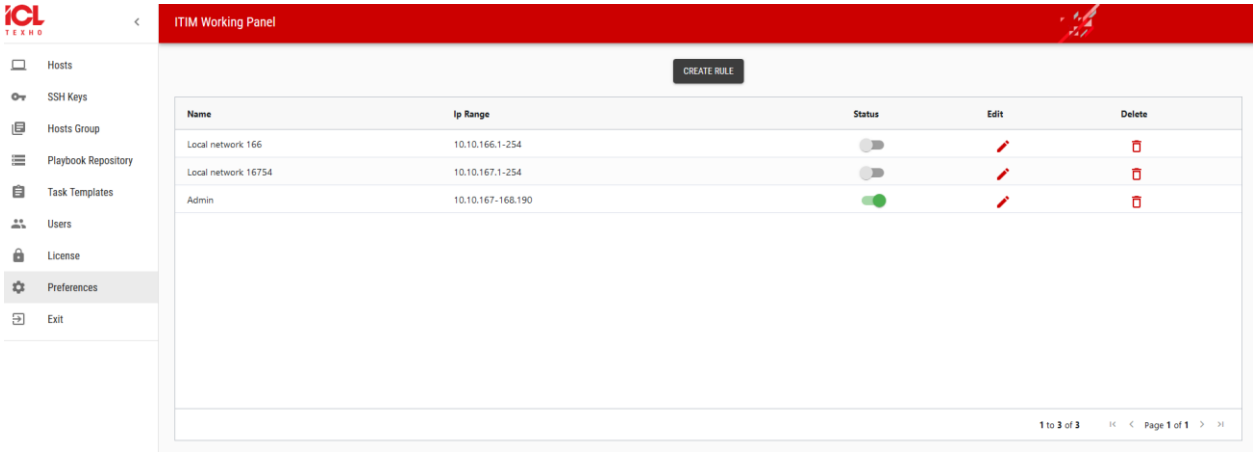

Рис.43 Пример отображения вкладки *Preferences.*

2. Нажмите кнопку *Create Rule*, далее заполните атрибуты правила обнаружения.
3. Заполните поле *Name*, указав в нём уникальное имя правила.
4. Заполните поле Range, указав в нём диапазон IP адресов обнаружения. Может принимать следующие форматы:

   Один IP: 192.168.1.33

   Диапазон IP адресов: 192.168.1–10.1–255.

   Диапазон ограничен общим количеством покрываемых адресов (менее чем 64К).

   Список: 192.168.1.1–255, 192.168.2.1–100, 192.168.2.200

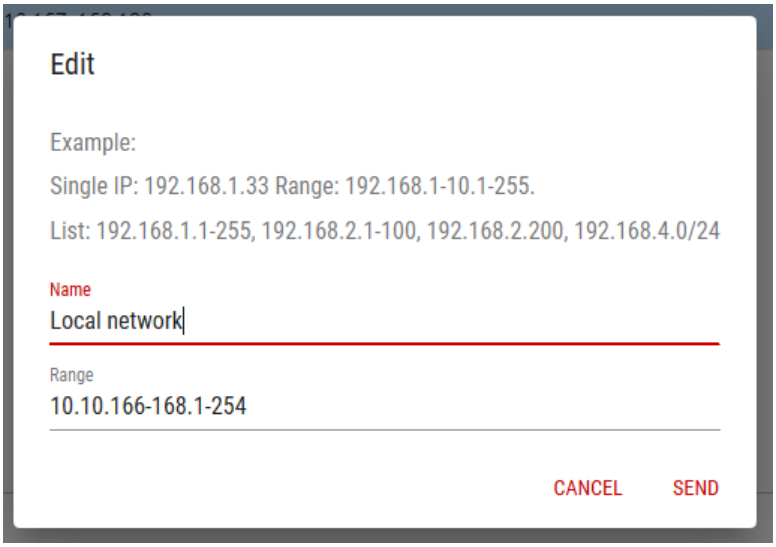

Рис.44 Пример заполнения атрибутов правила обнаружения.

5. Убедившись в правильном заполнении формы, нажмите *Send*, корректно заполненное правило автоматически добавится в таблице.

9.2 Установка Zabbix-agent на Windows

1. Запустите программу установки, нажмите *Next*, ознакомившись с лицензионным соглашением, выберите пункт *«I accept the terms in the License Agreement»* нажмите «*Next*».

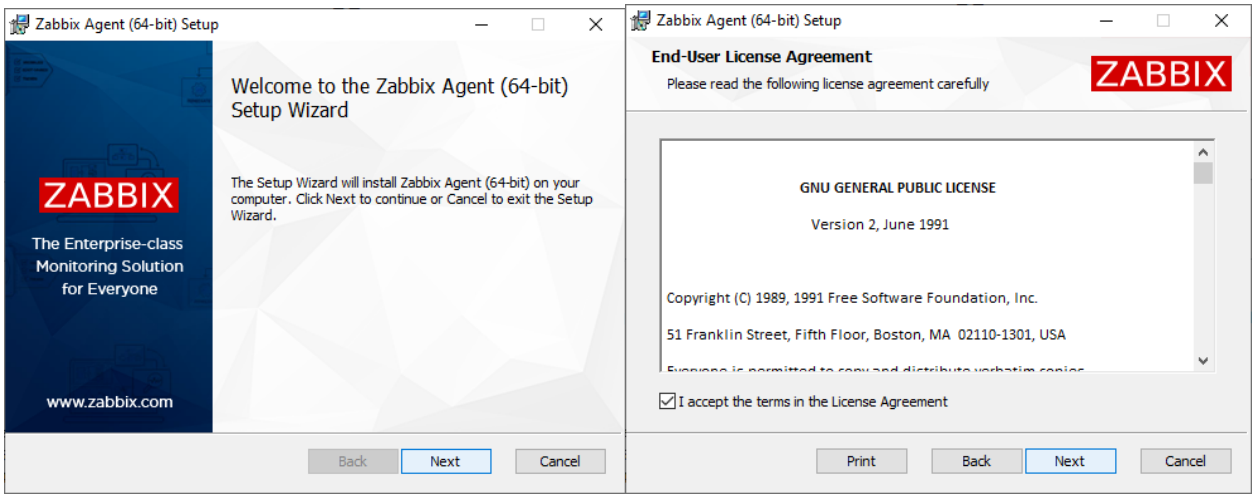

Рис.45 Окно установки и лицензии ПО.

2. Заполните *«Host name»*, указав уникальное имя узла. В поле *«Zabbix server IP»*, укажите IP адрес сервера, на котором развернута система. Указанные в данном окне настройки сохраняются в файле конфигурации Zabbix agent, и доступны к изменению по адресу по умолчанию: *C:\Program Files\Zabbix\zabbix_agentd.conf.*
3. Нажмите «*Next»,* для перехода к следующему окну установки

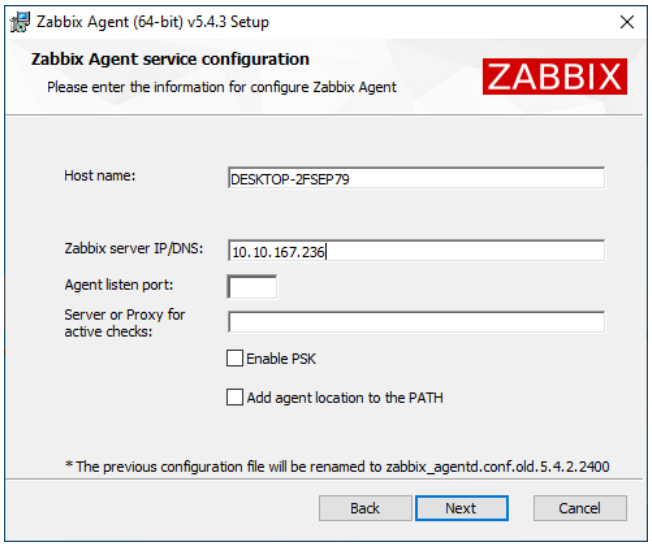

Рис.46 Окно ввода атрибутов

4. При необходимости измените путь установки по умолчанию, нажав Browse. Нажмите «Next», далее «Install». Установка завершена.

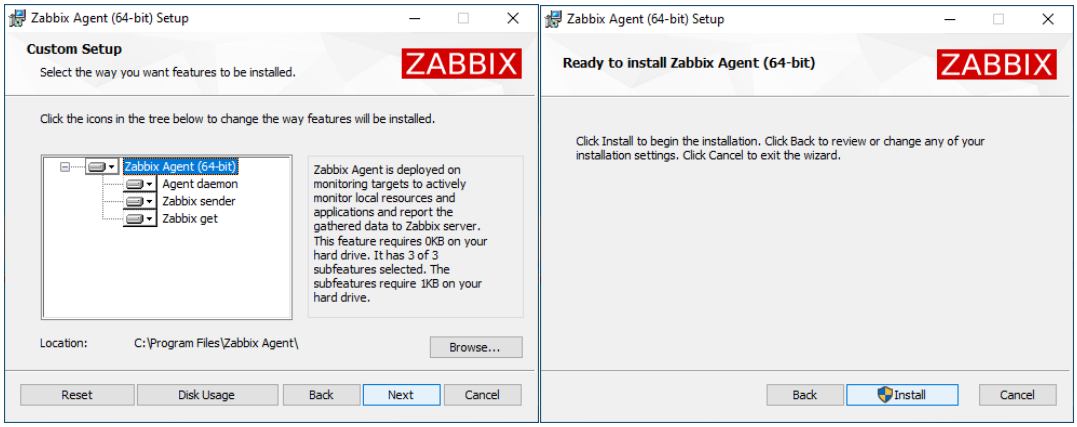

Рис.47 Окно выбора параметров установки.

5. После окончания установки, выполните комбинацию на клавиатуре «Win+R», в поле для ввода появившегося окна наберите services.msc нажмите «ОК».

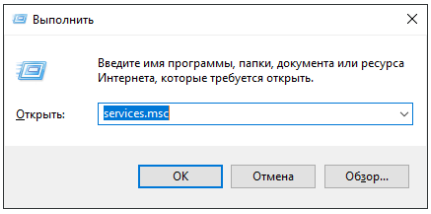

Рис.48 Окно выбора параметров установки.

6. В списке служб выберите «Zabbix Agent», кликнете правой кнопкой мыши по строке, выберите «Остановить».

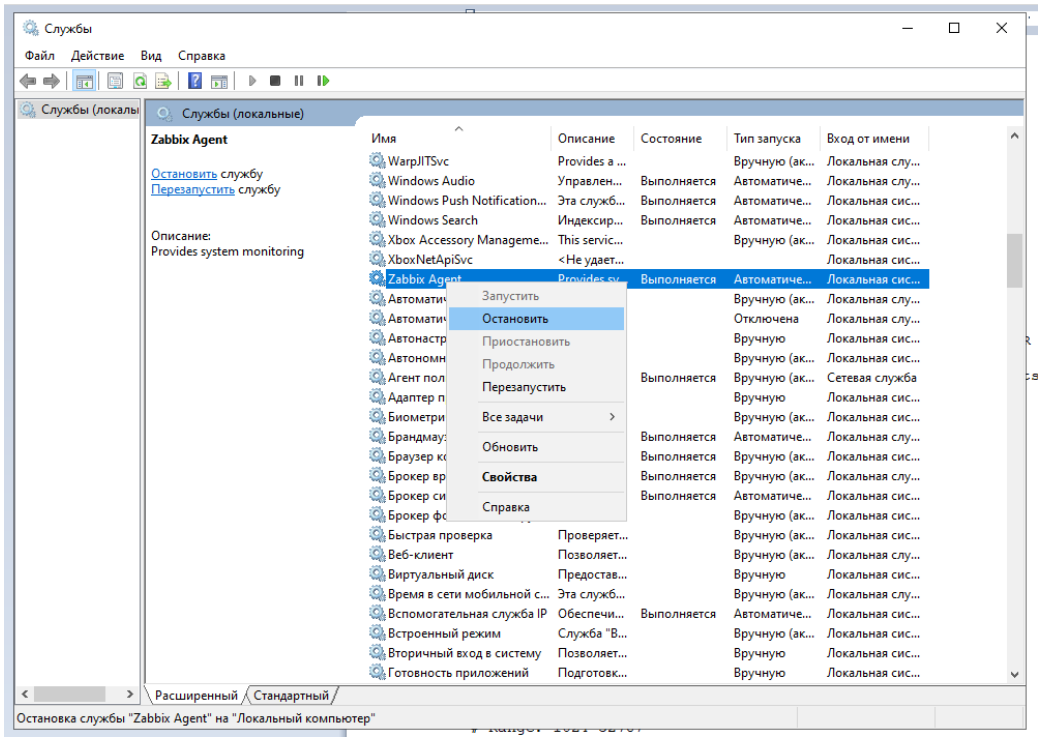

Рис.49 Окно выбора параметров установки.

7. Перейдите в папку с установленной программой «Zabbix agent» откройте файл конфигурации. Путь по умолчанию: «*C:\Program Files\Zabbix\zabbix_agentd.conf*».
8. В файле конфигурации раскомментируйте следующие значения:
   a. «EnableRemoteCommands» и установите значение «1»
   b. «Timeout» и установите значение «30»

```
### Option: Timeout
#     Spend no more than Timeout seconds on processing.
#
# Mandatory: no
# Range: 1-30
# Default:
# Timeout=3

Timeout=30
```

```
### Option: EnableRemoteCommands - Deprecated, use
AllowKey=system.run[*] or DenyKey=system.run[*] instead
#     Internal alias for AllowKey/DenyKey parameters depending
on value:
#     0 - DenyKey=system.run[*]
#     1 - AllowKey=system.run[*]
#
# Mandatory: no

EnableRemoteCommands=1
```

Рис.50 Окно выбора параметров установки.

9. Сохраните изменения в файле конфигурации, перейдите в «Службы», нажмите на строке «Zabbix agent» правой кнопкой, выберите запустить.

9.3 Установка Zabbix-agent на Ubuntu

1. Запустите терминал. Введите команду для установки пакета:

sudo apt install zabbix-agent

```
root@shamilubuntuserver: /home/shamil
root@shamilubuntuserver:/home/shamil# apt install zabbix-agent
```

Рис.51 Пример ввода команды установки Zabbix-agent

2. Введите следующую команду для редактирования файла конфигурации Zabbix-agent:

sudo nano /etc/zabbix/zabbix_agentd.conf

```
root@shamilubuntuserver: /home/shamil
root@shamilubuntuserver:/home/shamil# sudo nano /etc/zabbix/zabbix_agentd.conf
```

Рис.52 Пример ввода команды, открытие файла конфигурации для редактирования.

3. Пропишите в файле конфигурации IP адрес сервера, на котором развернута система напротив атрибута параметра «*Server*». Укажите в строке «*Hostname*» уникальное имя узла.

```
### Option: Server
#       List of comma delimited IP addresses, optionally in CIDR notation, or DNS names of Zabbix servers and Zabbix proxies.
#       Incoming connections will be accepted only from the hosts listed here.
#       If IPv6 support is enabled then '127.0.0.1', '::127.0.0.1', '::ffff:127.0.0.1' are treated equally
#       and '::/0' will allow any IPv4 or IPv6 address.
#       '0.0.0.0/0' can be used to allow any IPv4 address.
#       Example: Server=127.0.0.1,192.168.1.0/24,::1,2001:db8::/32,zabbix.example.com
#
# Mandatory: yes, if StartAgents is not explicitly set to 0
# Default:
# Server=

Server=10.10.167.236
```

Рис.53 Пример редактирования файла конфигурации

4. Сохраните файл конфигурации, перезапустите сервис Zabbix-agent выполнив команду:

sudo systemctl restart zabbix-agent

```
root@shamilubuntuserver: /home/shamil
root@shamilubuntuserver:/home/shamil# systemctl restart zabbix-agent
```

Рис.54 Пример ввода команды перезагрузки сервиса.

10 Приложения

10.1 Файл конфигурации

Ниже в таблице представлено описание параметров в файле конфигурации системы appsettings.json. Файл находится: /srv/itim/publish/appsettings.json Для применений изменений в файле конифгурации, необходимо перезапускать системы командой:

systemctl restart itim

Имя параметра	**Значение по умолчанию**	**Описание**
Zabbix:Url	http://{zabbix-web-ip:port}/api_jsonrpc.php	Адрес до API сервера Zabbix.
Zabbix:Login	Admin	Имя пользователя от учетной записи Zabbix-сервер, используемого при выполнении запросов API.
Zabbix:Password	zabbix	Пароль учетной записи Zabbix-сервер, используемого при выполнении запросов API.
Ansible:Url	http://{semaphore-ip:port}/api	Адрес до API сервера Ansible-semaphore.
Ansible:Login	Admin	Имя пользователя от учетной записи Ansible-semaphore, используемого при выполнении запросов API.
Ansible:Password	semaphorepassword	Пароль учетной записи Ansible-semaphore, используемого при выполнении запросов API.
Git:Url	http://{gitlab-ip:port}/	Адрес до API сервера Git.
Git:Login	root	Имя пользователя от учетной записи Git-Lab, используемого при выполнении запросов API.
Git:Password	gitlabPassword	Пароль учетной записи Git-Lab, используемого при выполнении запросов API.
Users:LDAPHost	Заполнить исходя из Вашего окружения. См. пункт. № 4	Адрес LDAP сервера
Users:LDAPPort		Порт LDAP сервера
Users:LDAPAccount		Учетная запись LDAP
Users:LDAPPassword		Пароль LDAP
Users:LDAPDomain		Домен LDAP

ConnectionStrings:Default	Server={адрес-MySQL};user={пользователь-MySQL};password={пароль-пользователя-MySQL};	Строка подключения к БД MySQL сервера. Содержит адрес сервера, имя пользователя, пароль.
Cors:Origin	http://{itim-web-ip:port}	Адрес docker контейнера с ITIM-Web. Используется в конфигурации политики запросов CORS.