

ИНСТРУКЦИЯ ПО УСТАНОВКЕ И ИСПОЛЬЗОВАНИЮ METASFERA TRACKING

Дата: 23.05.2022 г.

Версия: 1.3

**г. Москва
2022 г.**

Лист изменений

№ п/п	Дата	Версия	Краткое описание изменений
1.	01.11.2021	1.0	Первая версия документа
2.	05.04.2022	1.1.	Доработка документа по установке Metasfera Tracking
3.	20.04.2022	1.2	Доработка документа по установке backend Tracking
4.	23.05.2022	1.3	Доработка документа по установке backend Tracking на Windows

ГЛОССАРИЙ

Обозначение	Расшифровка
ОС	Операционная система.
ИС	Информационная система.
vCPU	Виртуальный процессор, содержащий одно ядро и занимающий один сокет.
ОП	Операционная система.
Web интерфейс	Совокупность веб-страниц, предоставляющая пользовательский интерфейс для взаимодействия с сервисом или устройством посредством протокола HTTP и веб-браузера.
Oracle VM VirtualBox	Программный продукт виртуализации для операционных систем.

СОДЕРЖАНИЕ

1. УСТАНОВКА METASFERA XD	4
2. ИМПОРТИРОВАНИЕ ПОДГОТОВЛЕННОЙ ВИРТУАЛЬНОЙ МАШИНЫ В СРЕДУ ВИРТУАЛИЗАЦИИ VIRTUALBOX	4
2.1. Импорт виртуальной машины	4
3. ПЕРВОЕ ВКЛЮЧЕНИЕ И ВХОД В ОПЕРАЦИОННУЮ СИСТЕМУ	8
3.1. Включение виртуальной машины	8
3.2. Аутентификация в операционной системе	9
4. УСТАНОВКА НЕОБХОДИМЫХ ПАКЕТОВ	9
4.1 Подготовка к установке	9
4.2 Установка PostgreSQL	10
4.3 Настройка PostgreSQL	10
4.4 Установка баз данных	10
4.5 Установка PostGIS	10
4.6 Установка Python	11
4.7 Установка Redis	11
4.8 Сборка и запуск Tracker	11
5. ЗАПУСК БРАУЗЕРА	12
6. НАСТРОЙКА METASFERA XD	14
6.1. Настройка веб-части	14
6.2. Запуск/остановка веб-части	18
6.3. Настройки серверной части	18
6.4. Запуск\остановка серверной части	22
6.5. Запуск\остановка системы управления базой данных (СУБД)	22
6.6. Конфигурация СУБД postgresql	22
7. УСТАНОВКА СЕРВИСОВ ПОД WINDOWS	23
7.1. Установка сервиса Tracking	23
7.2. Установка сервиса Vega Client	24

1. УСТАНОВКА METASFERA XD

Для инсталляции **Oracle VM VirtualBox** необходимо:

- Зайти на официальный сайт <https://www.virtualbox.org/wiki/Downloads>;
- Скачать установщик, подходящий под вашу операционную систему. Тестирование проводилось под ОС Windows 10.
- Произвести инсталляцию Oracle VM VirtualBox

2. ИМПОРТИРОВАНИЕ ПОДГОТОВЛЕННОЙ ВИРТУАЛЬНОЙ МАШИНЫ В СРЕДУ ВИРТУАЛИЗАЦИИ VIRTUALBOX

2.1. Импорт виртуальной машины

После инсталляции Oracle VM VirtualBox на новой Windows или на другом компьютере необходимо открыть Oracle VM VirtualBox и в меню «*Файл*» выбираем «*Импорт конфигураций*». (Рисунок 1)

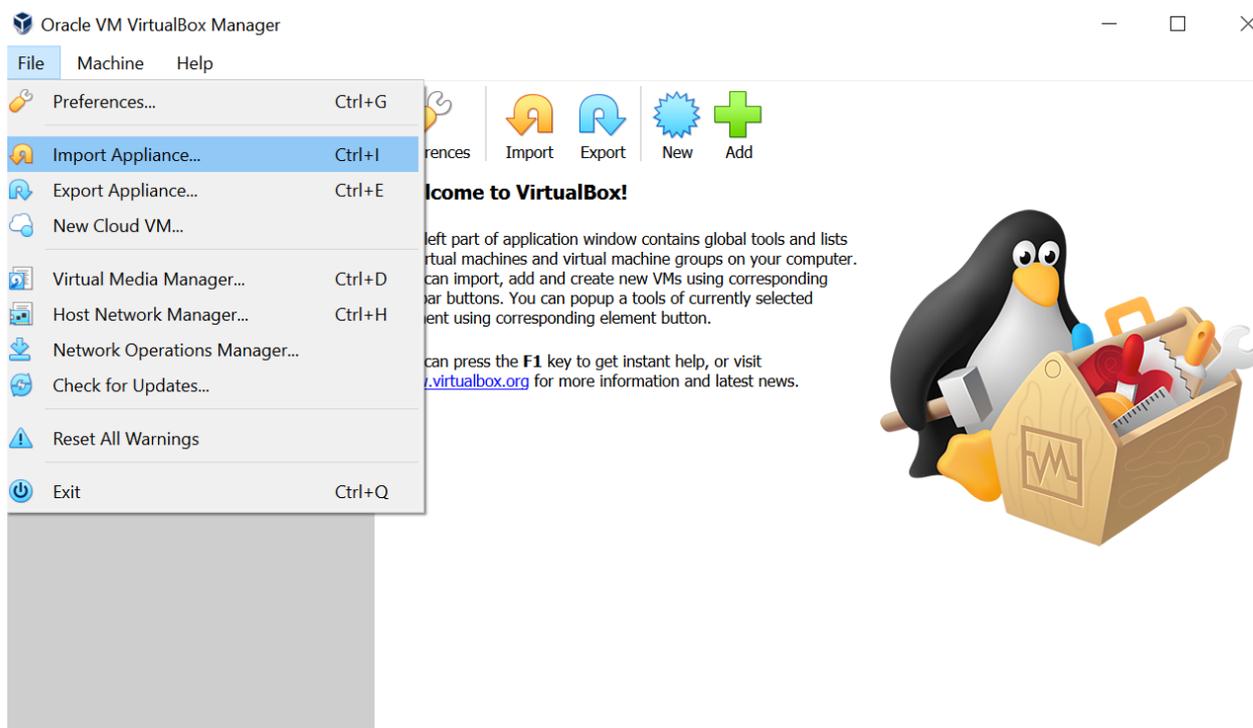


Рисунок 1. Импорт виртуальной машины

В следующем окне необходимо указать путь к файлу «.ova» с экспортированной виртуальной машиной. Необходимо нажать «*Next*». (Рисунок 2)

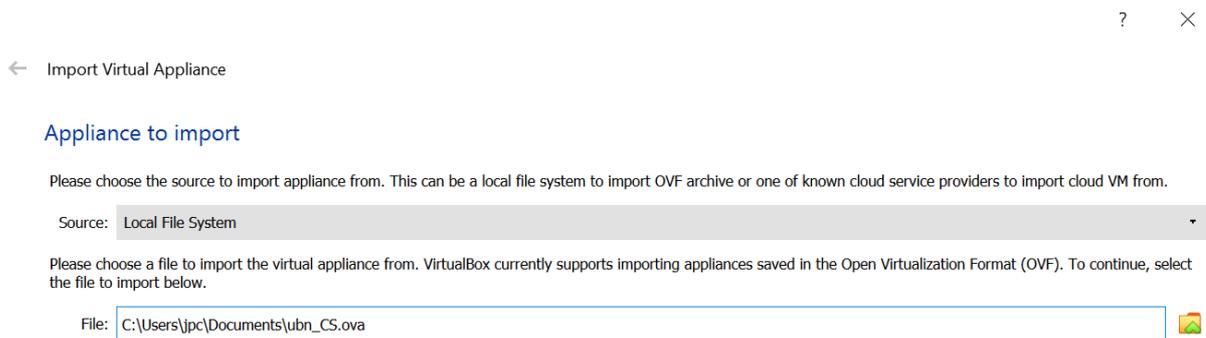


Рисунок 2. Путь к файлу «.ova»

Образ виртуальной машины необходимо взять с:

<https://disk.yandex.ru/d/pclEmgIwwM3nnA>

Login: user (запрашивается отдельно)

Password: 221201

Далее, при необходимости, можно изменить некоторые параметры – например, сменить имя виртуальной машины, уменьшить или увеличить оперативную память. (Рисунок 3)

Минимальные требования к виртуальной машине (для запуска):

1. Не менее 8 Гбайт оперативной памяти;
2. Не менее 40 Гбайт раздел жесткого диска;
3. Не менее 2 vCPU

Поддерживаемые ОС:

1. РЕД ОС
2. Red Hat Enterprise Linux (RHEL) (32-bit or 64-bit)
3. Novel SUSE Linux Enterprise Server (SLES) (32-bit or 64-bit)
4. Microsoft Windows Server (32-bit or 64-bit)
5. Apple Mac OS
6. Solaris SPARC
7. Fedora
8. Debian Linux
9. HP-UX
10. FreeBSD
11. CentOS
12. Ubuntu

Поддерживаемые веб-браузеры:

1. Mozilla Firefox

2. Microsoft Internet Explorer
3. Apple Safari
4. Google Chrome

* при восстановлении виртуальной машины ресурсы берутся те же, что и были в оригинале, т.е. назначать их руками нет необходимости.

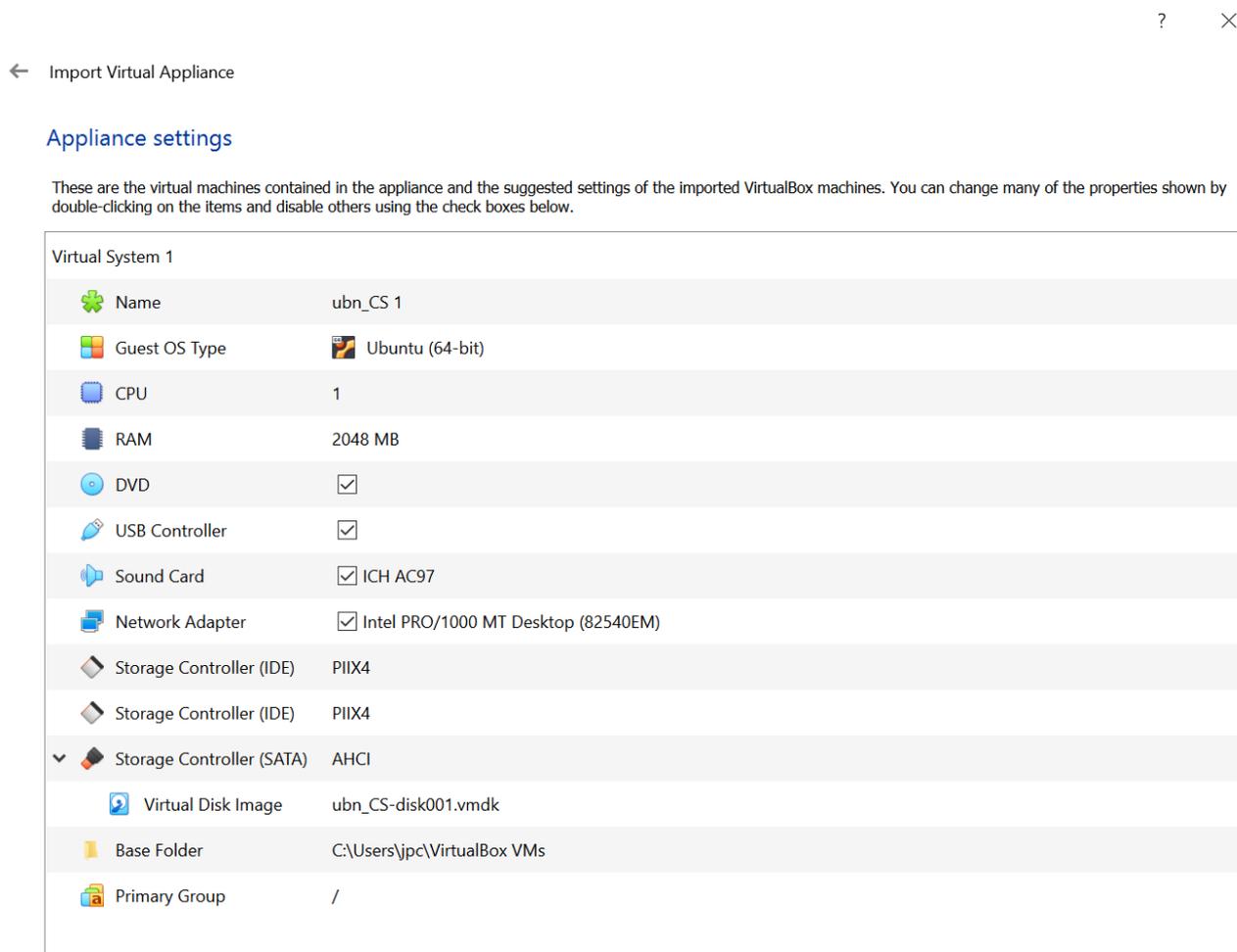


Рисунок 3 Свойства виртуальной машины

В самом конце окна будет указан путь, куда после импорта будет помещён файл жёсткого диска «.vdi». По умолчанию это системный диск, однако целевая папка может быть изменена. Необходимо нажать кнопку «Импорт», после чего начнется импорт виртуальной машины. (Рисунок 4)

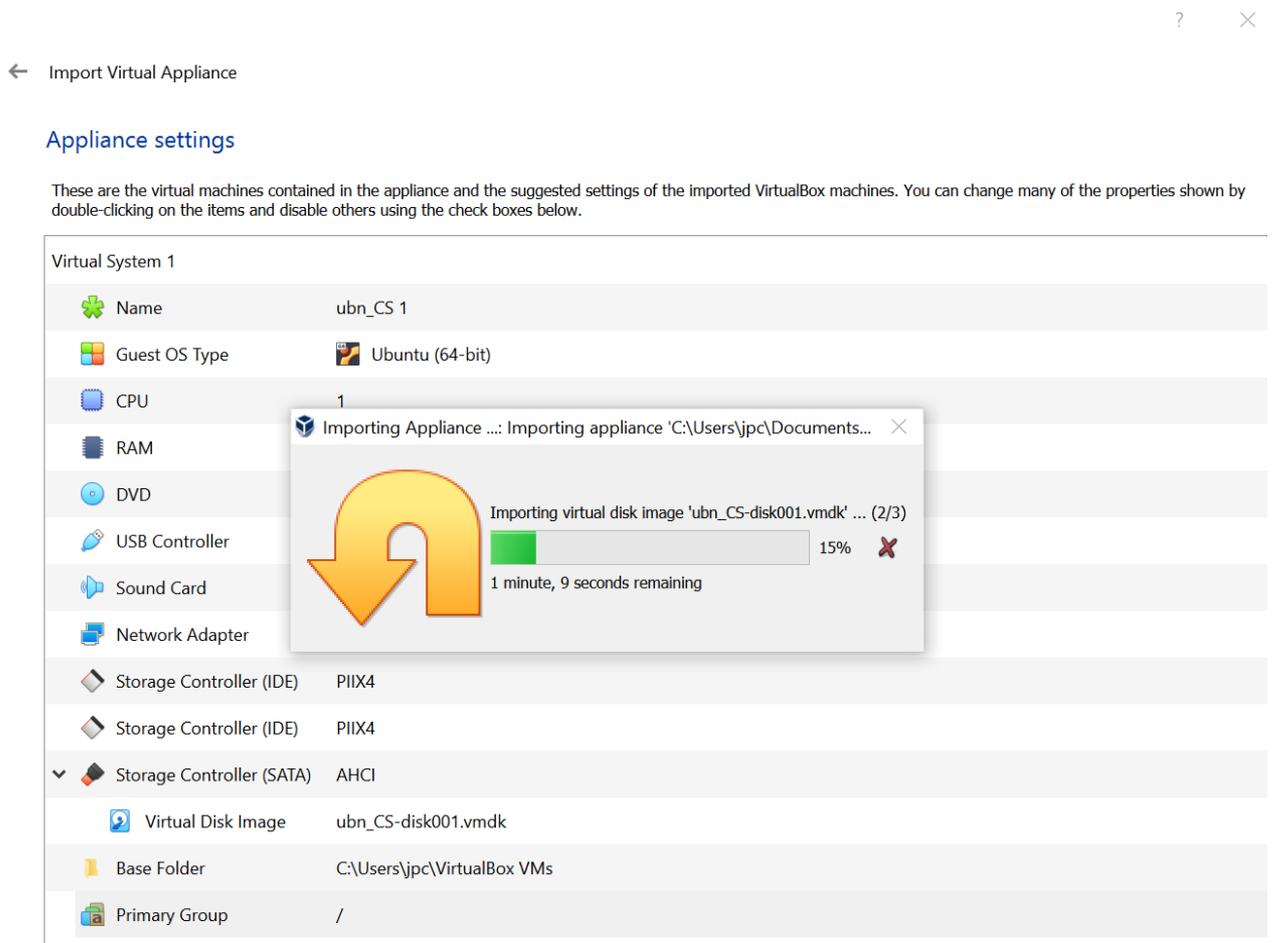


Рисунок 4 Импорт виртуальной машины

Дожидаемся завершения процесса импорта. (Рисунок 5)

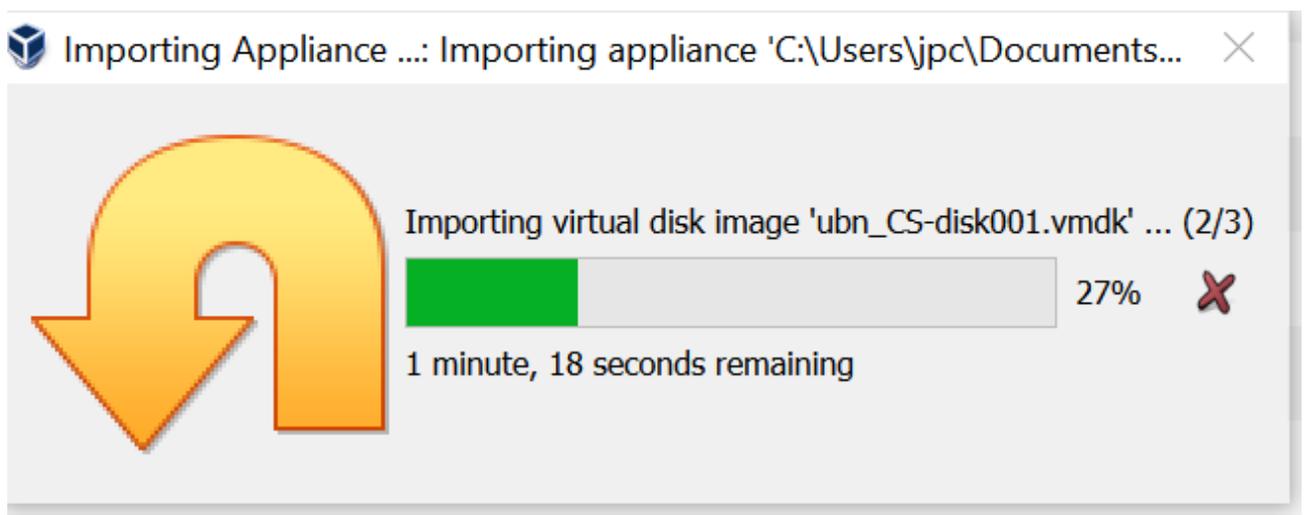


Рисунок 5 Окно импорта Виртуальной машины

Теперь импортированная виртуальная машина находится в списке машин Oracle VM VirtualBox . (Рисунок 6)

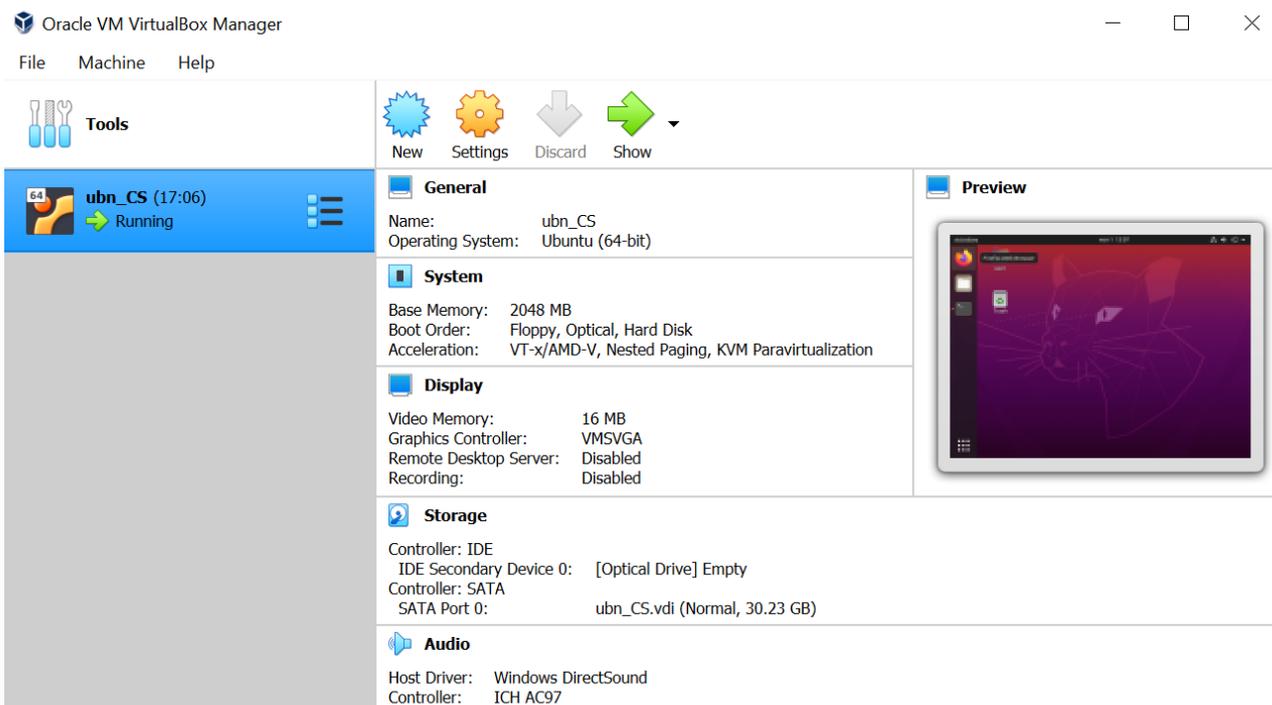


Рисунок 6 Импортёрванная виртуальная машина

3. ПЕРВОЕ ВКЛЮЧЕНИЕ И ВХОД В ОПЕРАЦИОННУЮ СИСТЕМУ

3.1. Включение виртуальной машины

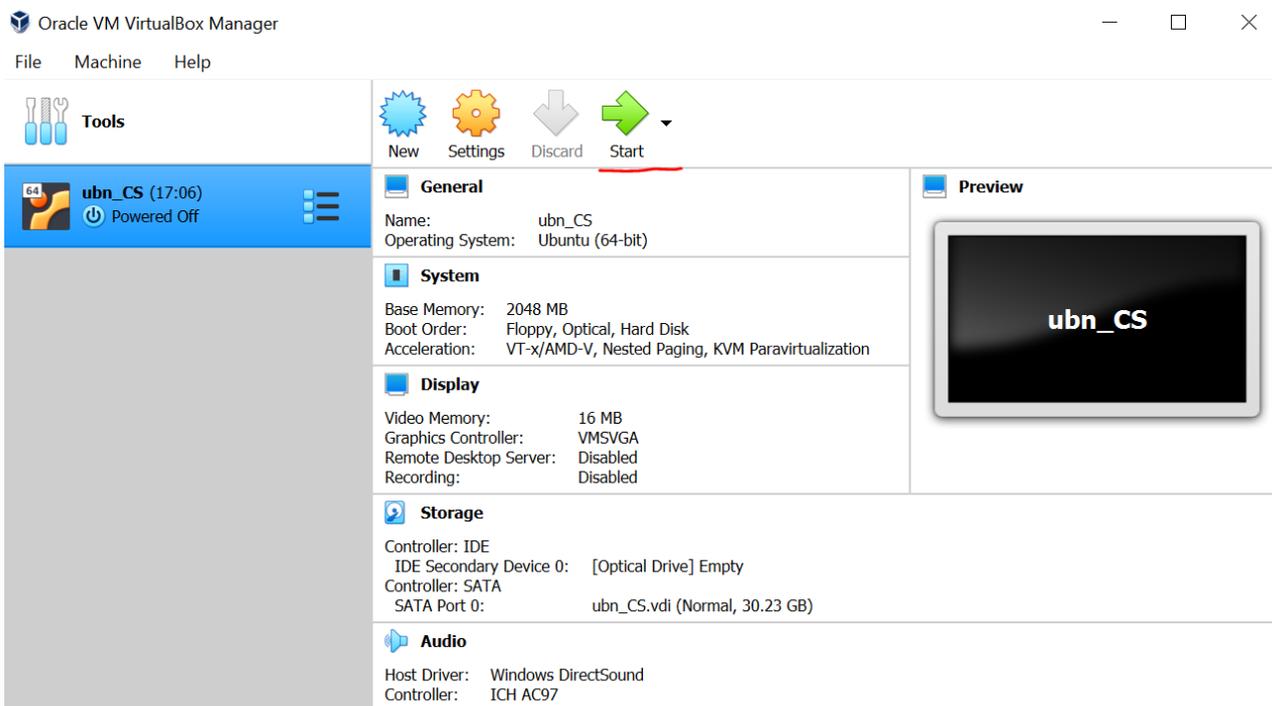


Рисунок 7 Включение виртуальной машины

Необходимо зайти в главное окно Oracle VM VirtualBox и запустить виртуальную машину, нажав на кнопку «Start». (Рисунок 7) Если до этого у вас не было установленных виртуальных машин, то в списке

слева она будет одна. Иначе выберите виртуальную машину с именем ubn_CS.

3.2. Аутентификация в операционной системе

После того, как операционная система запустится необходимо выбрать пользователя user и ввести пароль. (Рисунок 8)

Пользователь: user

Пароль: 221201

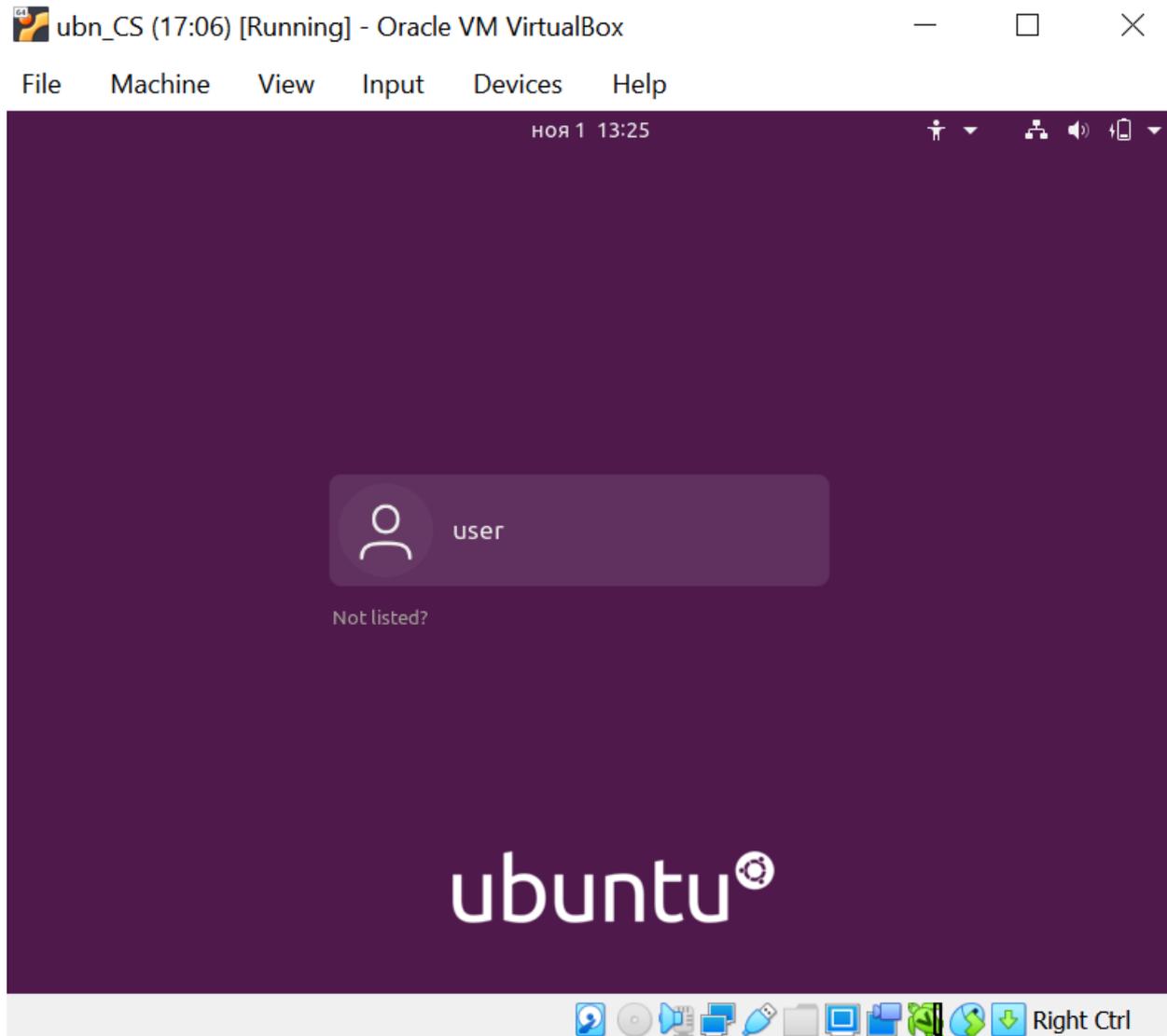


Рисунок 8 Вход в операционную систему

4. УСТАНОВКА НЕОБХОДИМЫХ ПАКЕТОВ

4.1 Подготовка к установке

Откройте терминал и выполните (обновит менеджер пакетов): **apt -y update && apt -y upgrade**

Установка общих вспомогательных пакетов: **apt -y install vim bash-completion wget gnupg lsb-core firewall-applet ffmpeg libsm6 libxext6**

4.2 Установка PostgreSQL

Выполните следующую последовательность команд:

- Получение и установка ключа для загрузки пакета: **wget --quiet -O - https://www.postgresql.org/media/keys/ACCC4CF8.asc | apt-key add -**
- Добавление нового репозитория: **echo "deb http://apt.postgresql.org/pub/repos/apt/ `lsb_release -cs`-pgdg main" | tee /etc/apt/sources.list.d/pgdg.list**
- Обновление пакетного менеджера, чтобы изменения выше применились: **apt -y update**
- Установка PostgreSQL: **apt -y install postgresql-13 postgresql-client-13**
- Подключение службы:
 - **systemctl enable postgresql**

4.3 Настройка PostgreSQL

Войти в систему под стандартным пользователем PostgreSQL: **su - postgres**

Установить ему новый пароль: **psql -c "alter user postgres with password 'postgres'"**

Создать пользователя и БД:

- **createuser osm**
- **psql -c "alter user osm with password 'osm'"**
- **exit**

Настройки PostgreSQL:

Командой **vi /etc/postgresql/13/main/postgresql.conf** настроить параметр **listen_addresses = '*'**

Командой **vi /etc/postgresql/13/main/pg_hba.conf** настроить параметры:

local	all	all		md5
host	all	all	0.0.0.0/0	md5
host	all	all	:::1/128	md5

Настройки firewall:

- **systemctl start firewalld**
- **firewall-cmd --permanent --zone=public --add-port=5432/tcp**
- **firewall-cmd --reload**

Запуск службы:

- **systemctl daemon-reload**
- **systemctl restart postgresql**

4.4 Установка баз данных

su - postgres

psql -c "create database 'db_track'"

pg_restore -h localhost -p 5432 -U postgres -d db_track -v "/opt/dbs/db_track.db"
exit

4.5 Установка PostGIS

apt -y install postgis postgresql-13-postgis-3

4.6 Установка Python

Установка python 3.9: `apt -y install python3.9`

Установка пакетного менеджера pip: `apt -y install python3-pip`

4.7 Установка Redis

Подключение репозитория:

- `curl -fsSL https://packages.redis.io/gpg | gpg --dearmor -o /usr/share/key-rings/redis-archive-keyring.gpg`
- `echo "deb [signed-by=/usr/share/keyrings/redis-archive-keyring.gpg] https://packages.redis.io/deb $(lsb_release -cs) main" | tee /etc/apt/sources.list.d/redis.list`
- `apt update`

Установка Redis:

`apt -y install redis`

Запуск Redis:

`redis-server --daemonize yes`

4.8 Сборка и запуск Tracker

Загрузить архив `backend_tracker.zip` в папку `/opt`

`cd /opt`

`unzip Tracking back.zip -d backend_tracker`

Создание сервиса:

`vi /etc/systemd/system/backend_tracker.service`

```
[Unit]
Description=backend_tracker
After=network.target

[Service]
User=root
WorkingDirectory=/opt/backend_tracker
LimitNOFILE=4096
ExecStart=tracker_compiled
Restart=on-failure
RestartSec=5s

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

Настройка firewall:

- `firewall-cmd --permanent --zone=public --add-port=8881/tcp`
- `firewall-cmd --reload`

Инициализация и запуск сервиса:

- `systemctl daemon-reload`
- `systemctl enable backend_tracker`
- `systemctl start backend_tracker`

5. ЗАПУСК БРАУЗЕРА

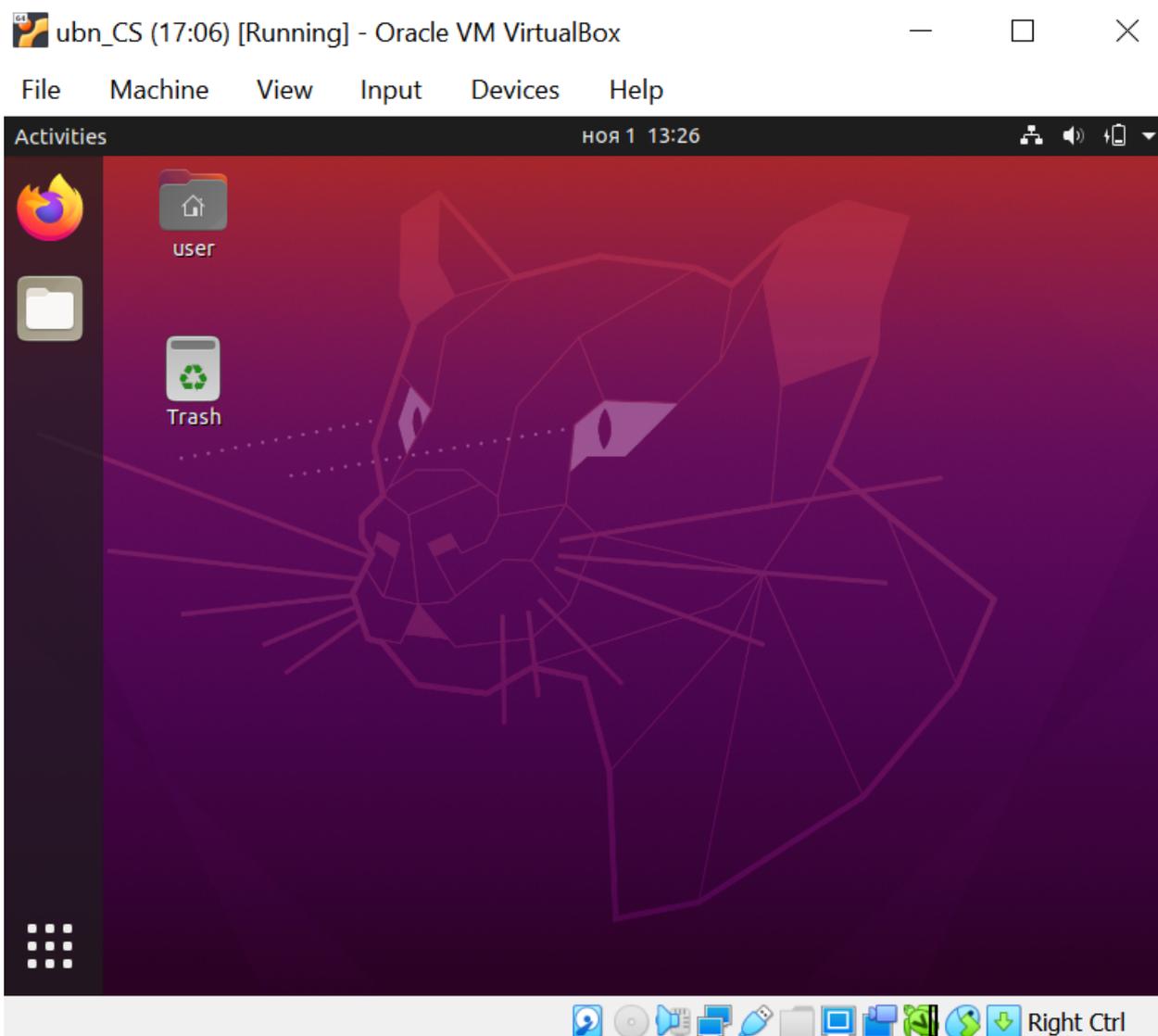


Рисунок 9 Desktop виртуальной машины

Необходимо нажать на иконку браузера «Mozilla Firefox» (в данной операционной системе установлен по умолчанию), после чего запустится данный браузер. (Рисунок 9)

По умолчанию открывается окно с логином в систему. (Рисунок 10)

Если оно не открылось, выбираем из закладки.

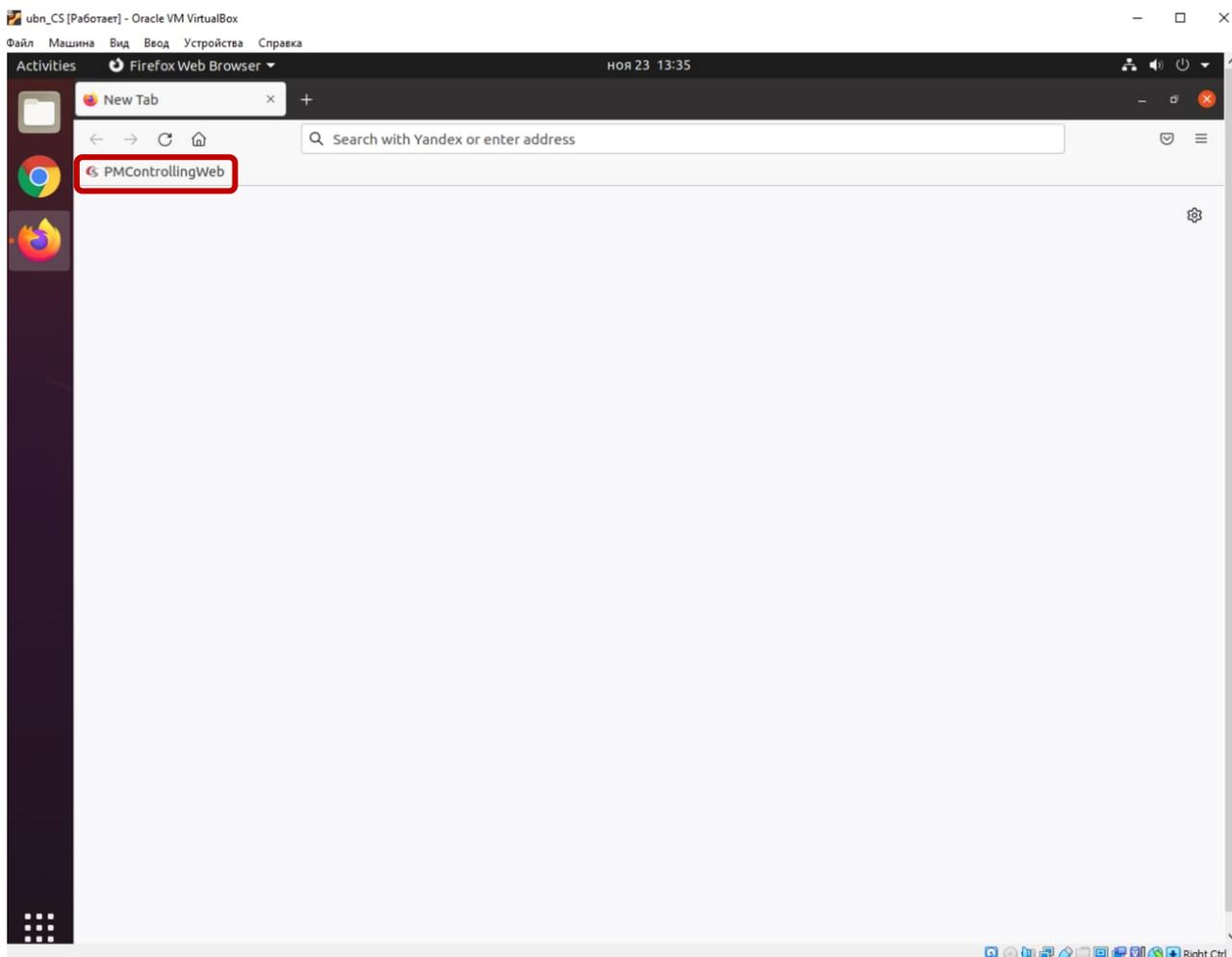


Рисунок 10 Окно с логином в систему

После чего откроется страница логина. (Рисунок 11) . На странице логина необходимо ввести логин и пароль от системы.

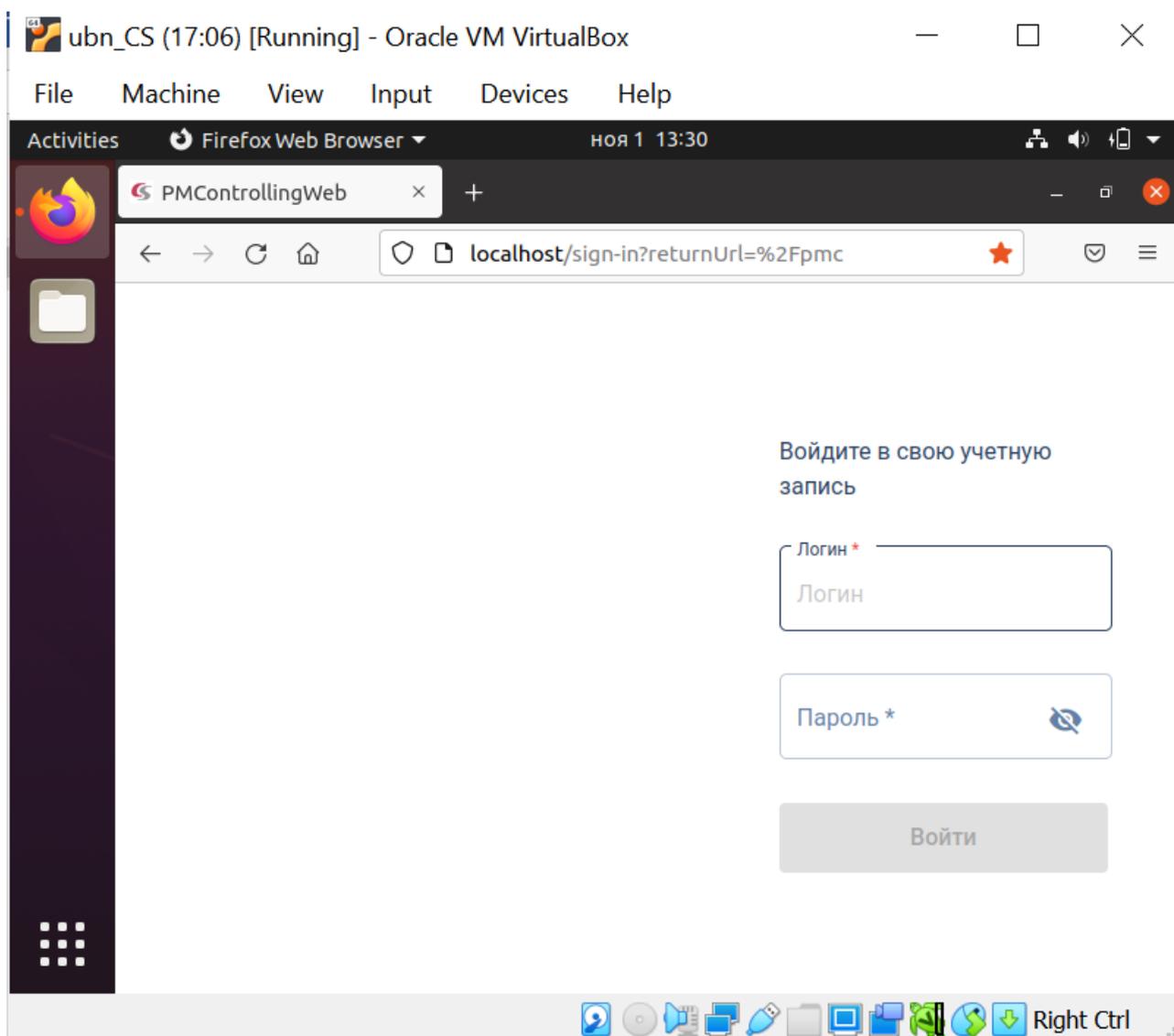


Рисунок 11 Страница входа в УЗ

6. НАСТРОЙКА METASFERA TRACKING

6.1. Настройка веб-части

К конфигурации web интерфейса системы относятся параметры необходимые для обновления версии и отображения системы.

Настройка порта соединения, корневой директории сайта и способа отображения находятся в конфигурационном файле веб сервера.

Конфигурационный файл находится:

```
/etc/nginx/nginx.conf
```

И выглядит следующим образом

```
user www-data;  
worker_processes auto;  
pid /run/nginx.pid;  
include /etc/nginx/modules-enabled/*.conf;
```

```
events {
```

```
    worker_connections 768;
    # multi_accept on;
}

http {

    ##
    # Basic Settings
    ##

    sendfile on;
    tcp_nopush on;
    tcp_nodelay on;
    keepalive_timeout 65;
    types_hash_max_size 2048;
    # server_tokens off;
    # server_names_hash_bucket_size 64;
    # server_name_in_redirect off;
    include /etc/nginx/mime.types;
    default_type application/octet-stream;

    ##
    # SSL Settings
    ##

    ##
    # Logging Settings
    ##

    access_log /var/log/nginx/access.log;
    error_log /var/log/nginx/error.log;

    ##
    # Gzip Settings
    ##
    gzip on;
    # gzip_vary on;
    # gzip_proxied any;
    # gzip_comp_level 6;
    # gzip_buffers 16 8k;
    # gzip_http_version 1.1;
    # gzip_types text/plain text/css application/json application/javascript text/xml
application/xml application/xml+rss text/javascript;

    ##
    # Virtual Host Configs
```

```
##

include /etc/nginx/conf.d/*.conf;
include /etc/nginx/sites-enabled/*;
server {
    listen    80;
    server_name 192.168.10.11;
    underscores_in_headers on;
    root /etc/nginx/html;
    #charset koi8-r;
    #access_log logs/host.access.log main;
    location / {
        try_files $uri $uri/ /index.html =404;
    }

    location /api/ {
        proxy_pass http://192.168.10.11:4300;
        proxy_http_version 1.1;
        proxy_set_header Upgrade $http_upgrade;
        proxy_set_header Connection 'upgrade';
        proxy_set_header Host $host;
        proxy_cache_bypass $http_upgrade;
        client_max_body_size 100M;
        proxy_read_timeout 300;
        proxy_connect_timeout 300;
        proxy_send_timeout 300;
        send_timeout 300;
    }
    location /socket.io/ {
        proxy_pass http://192.168.10.11:4300;
        proxy_http_version 1.1;
        proxy_set_header Upgrade $http_upgrade;

        proxy_set_header Connection "upgrade";
        proxy_set_header Host $host;
    }
}

#mail {
#    # See sample authentication script at:
#    # http://wiki.nginx.org/ImapAuthenticateWithApachePhpScript
#
#    # auth_http localhost/auth.php;
```

```
# # pop3_capabilities "TOP" "USER";
# # imap_capabilities "IMAP4rev1" "UIDPLUS";
#
# server {
#     listen    localhost:110;
#     protocol  pop3;
#     proxy    on;
# }
#
# server {
#     listen    localhost:143;
#     protocol  imap;
#     proxy    on;
# }
#}
```

Ниже приведем отдельные его части с пояснением того, что они значат.

1. server {
2. listen 80; - указывает какой порт готов принимать входящее соединения
3. server_name 192.168.10.11; - указывает адрес на котором работает веб сервер
4. underscores_in_headers on;
5. root /etc/nginx/html; - указывает корневую директорию.
- 2.
1. location / { - инструкция указывающая на корневой каталог с ответом на ошибочные запросы
2. try_files \$uri \$uri/ /index.html =404;
3. }
- 3.
1. location /api/ { -блок инструкций по указанию работы с серверной частью части веб интерфейса
2. proxy_pass http://192.168.10.11:4300; - адрес серверной части
3. proxy_http_version 1.1; - версия запросов используемая в ПО
4. proxy_set_header Upgrade \$http_upgrade; - системные настройки части переадресации запросов
5. proxy_set_header Connection 'upgrade'; ; - системные настройки части переадресации запросов
6. proxy_set_header Host \$host; - системные настройки части переадресации запросов
7. proxy_cache_bypass \$http_upgrade; - системные настройки части переадресации запросов

8. `client_max_body_size 100M`; - максимальный допустимый размер файлов для передачи через веб интерфейс
9. `proxy_read_timeout 300`; - таймаут ожидания после которого веб приложение выдаст ошибку
10. `proxy_connect_timeout 300`; - таймаут ожидания после которого веб приложение выдаст ошибку в варианте 2
11. `proxy_send_timeout 300`; - таймаут ожидания после которого веб приложение выдаст ошибку в варианте 3
12. `send_timeout 300`; - таймаут ожидания после которого веб приложение выдаст ошибку в варианте 4
13. }

6.2. Запуск/остановка веб-части

Для запуска веб интерфейса необходимо выполнить команду
`systemctl start nginx`

Для остановки веб интерфейса необходимо выполнить команду
`systemctl stop nginx`

6.3. Настройки серверной части

Главный файл конфигурации серверной части располагается по следующему пути `/opt/pmcontrolling.api/PMC-api/config.xml`

Данная конфигурация выглядит следующим образом

```
"NODE_ENV_": "PROD",
"NODE_ENV": "DEV",
"secret": "secret_here",
"token_expires": "7d",
"time_to_check_expire": 720,
"length": 128,
"digest": "sha256",
"module": 50,
"main_type_db_connection": "postgres",
"is_convert_xml": false,
"log": {
  "LOG_LEVEL": "debug",
  "LOG_OUT_MODE": "short"
},
"api": {
  "title": "REST API",
  "description": "REST services",
  "tag": "PMC",
  "version": "5.0.0",
  "host": "localhost",
  "httpPort": 4300,
```

```
"httpsPort": 4301,  
"sslKey": "pmc.key",  
"sslCert": "pmc.crt",  
"ssl": false,  
"https": false  
},  
"db_connection": [  
  {  
    "type": "mssql",  
    "database": "db_name",  
    "server": "server_name",  
    "user": "user",  
    "password": "password",  
    "port": 1433,  
    "requestTimeout": 60000,  
    "connectionTimeout": 60000,  
    "options": {  
      "appName": "WE",  
      "dateFormat": "dmy",  
      "datefirst": 1,  
      "language": "русский",  
      "encrypt": false,  
      "cancelTimeout": 5000,  
      "enableArithAbort": true  
    }  
  },  
  {  
    "type": "postgres",  
    "database": "db_name",  
    "server": "server_name",  
    "user": "user_name",  
    "password": "password",  
    "port": 5432,  
    "connectionTimeout": 60000,  
    "options": {  
      "appName": "WE",  
      "dateFormat": "dmy",  
      "datefirst": 1,  
      "language": "русский",  
      "encrypt": false  
    }  
  }  
],  
"upload": {
```

```
"uploadPath": "/opt/uploads/",
"fileSize": 104857600,
"streamPort": 9999
},
"xmlConvert": {
"compact": true,
"ignoreComment": true,
"nativeType": true,
"spaces": 2
},
"convert": {
"fromFolderPath": "./In",
"inFolderPath": "./Out",
"options": "-options alerts:no"
},
"auth": {
"isLdap": false,
"ldap": {
"url": "ldap://adress",
"url_reserve1": " ldap://adress:389",
"url_reserve2": " ldap://adress:389",
"baseDN": "dc=cs,dc=local",
"baseDN_reserve1": "dc=exampl,dc=local",
"baseDN_reserve2": "dc= exampl,dc=local",
"postfix": "@ exampl.local",
"postfix_reserve1": "@ exampl.local",
"postfix_reserve2": "@ exampl.local"
}
},
"login_image": {
"analytics": "assets/images/png/bg_img.png",
"gsp": "assets/images/jpg/bg_img.jpg"
},
"reporting": {
"reportViewerPath": "/opt/Work/Applications/Report",
"defaultreportPath": "/opt/ReportFiles/Additional Folder",
"username": "username",
"password": "password"
}
}
```

Далее приведем блоки конфигурационного файла с пояснениями:

1. Блок первый

1. "NODE_ENV": "DEV", - обозначение среды на которой запущен сервер
2. "secret": "secret_here", - задание пароля

3. "token_expires": "7h", - время жизни ключа авторизации
 4. "time_to_check_expire": 720, - время за которое при бездействии пропадет ключ
 5. "length": 128, - длина ключа авторизации
 6. "digest": "sha256", - подпись ключа авторизации
 7. "module": 50,
 8. "main_type_db_connection": "postgres", - переключение способа подключения к БД
 9. "is_convert_xml": false,
 10. "log": {
 11. "LOG_LEVEL": "debug", - уровень ведения логирования приложения
 12. "LOG_OUT_MODE": "short" – тип завершения работы с системой
2. Два блока описывающие два разных типа подключения к базе данных
1. },
 2. "db_connection": [- 3. {
 - 4. "type": "mssql", - тип используемой БД
 - 5. "database": "db_name", - имя бд
 - 6. "server": "server_name", - имя сервера бд
 - 7. "user": "user", - имя пользователя
 - 8. "password": "password", - пароль
 - 9. "port": 1433, - порт
 - 10. "requestTimeout": 60000, - время ожидания ответа
 - 11. "connectionTimeout": 60000, - - время ожидания ответа
 - 12. "options": {
 - 13. "appName": "WE", - имя приложения
 - 14. "dateFormat": "dmy", - формат времени
 - 15. "datefirst": 1,
 - 16. "language": "русский",
 - 17. "encrypt": false,
 - 18. "cancelTimeout": 5000,
 - 19. "enableArithAbort": true
 - 20. }
 - 21. },
 - 22. {
 - 23. "type": "postgres", - тип используемой БД
 - 24. "database": "db_name", - имя бд
 - 25. "server": "server_name", - имя сервера бд
 - 26. "user": "user_name", - имя пользователя
 - 27. "password": "password", - пароль
 - 28. "port": 5432, - порт
 - 29. "connectionTimeout": 60000, - - время ожидания ответа
 - 30. "options": {

31. "appName": "WE", - имя приложения
 32. "dateFormat": "dmy",
 33. "datefirst": 1,
 34. "language": "русский",
 35. "encrypt": false
 36. }
 37. }
 38.],
3. Блок описания корневой директории файлов системы
 1. "upload": {
 2. "uploadPath": "/opt/uploads/", - путь
 3. "fileSize": 104857600, - размер файла
 4. "streamPort": 9999 - порт
 5. },
 4.
 1. "login_image": {
 2. "analytics": "assets/images/png/bg_img.png", - путь к главному изображению
 3. "gsp": "assets/images/jpg/bg_img.jpg" - путь к главному изображению
 4. },
 5. "reporting": { - блок отчетов
 6. "reportViewerPath": "/opt/Work/Applications/Report", - путь к отчетам
 7. "defaultreportPath": "/opt/ReportFiles/Additional Folder", - путь к отчетам дополнительный
 8. "username": "username", - имя пользователя
 9. "password": "password" – пароль

6.4. Запуск\остановка серверной части

Для того, чтобы запустить серверную часть, необходимо выполнить две команды в терминале

1. cd /opt/pmcontrolling.api/pmc-api
2. npm run start:prod

Для того, чтобы остановить серверную часть, необходимо выполнить две команды в терминале

3. cd /opt/pmcontrolling.api/pmc-api
4. npm run stop

6.5. Запуск\остановка системы управления базой данных (СУБД)

Для того, чтобы запустить СУБД, необходимо выполнить команду
systemctl start postgresql

Для того, чтобы остановить СУБД, необходимо выполнить команду
systemctl stop postgresql

6.6. Конфигурация СУБД postgresql

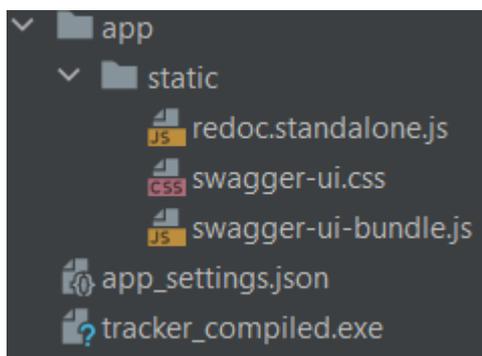
Основной конфигурационный файл — postgresql.conf

Если необходимо, при запуске сервера, значения параметров можно передать команде postgres в аргументе командной строки -c. Например:
postgres -c log_connections=yes -c log_destination='syslog'

7. УСТАНОВКА СЕРВИСОВ ПОД WINDOWS

7.1. Установка сервиса Tracking

В директорию, отведенную под запуск Tracking, поместить файлы со следующей структурой:



Файл **app_settings.json** содержит конфигурации по подключению к БД и микро сервису авторизации;

Директории **app/static** содержит файлы необходимые для отображения Swagger-UI (интерфейс визуализирующий доступные endpoints приложения);

Исполняемый файл **tracker_compiled.exe**, при запуске приложение будет доступно на порту 4581.

Для того, чтобы запустить приложение как службу Windows используется **nssm.exe**

Пример запуска:

```
C:\nssm\win64\nssm.exe install GISTRACKER "C:\gis_tracker\tracker_compiled.exe" AppDirectory "C:\gis_tracker\"
```

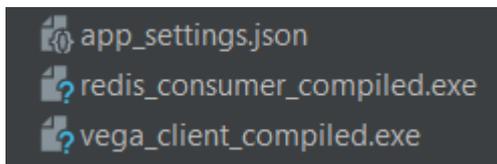
Общий вид команды:

```
[Путь до nssm.exe]/nssm.exe install [Имя Сервиса] "[Путь до tracker_compiled.exe]/tracker_compiled.exe" AppDirectory "[Путь до tracker_compiled.exe]"
```

После выполнения команды сервис нужно будет запустить вручную в приложении Windows "Службы"

7.2. Установка сервиса Vega Client

В директорию, отведенную под запуск Vega Client поместить файлы со следующей структурой:



Файл **app_settings.json** содержит конфигурации по подключению к сервису Tracking (порт 4581), API Vega Client и Redis;

Файл **vega_client_compiled.exe** – приложение по взаимодействию с Vega API (получение данных), Tracking (получение данных для связи их с данными от Vega API) и Redis (для хранения обработанных данных);

Файл **redis_consumer_compiled.exe** – приложение по взаимодействию с Redis (получение ранее записанных обработанных данных) и Tracking (запись данных в БД).

Для того, чтобы запустить приложения как службу Windows используется nssm.exe

Примеры запуска:

```
C:\nssm\win64\nssm.exe install VEGA_CLIENT "C:\vega_client\  
vega_client_compiled.exe" AppDirectory "C:\vega_client\"
```

```
C:\nssm\win64\nssm.exe install REDIS_CONSUMER "C:\vega_client\  
redis_consumer_compiled.exe" AppDirectory "C:\vega_client\"
```

Общий вид команд:

```
[Путь до nssm.exe]/nssm.exe install [Имя Сервиса] "[Путь до  
vega_client_compiled.exe]/vega_client_compiled.exe" AppDirectory "[Путь до  
vega_client_compiled.exe]"
```

```
[Путь до nssm.exe]/nssm.exe install [Имя Сервиса] "[Путь до  
redis_consumer_compiled.exe]/redis_consumer_compiled.exe" AppDirectory  
"[Путь до redis_consumer_compiled.exe]"
```

После выполнения команды сервис нужно будет запустить вручную в приложении Windows "Службы"