# Система фиксации технических сбоев для терминалов

Руководство по установке и первоначальной настройке программного продукта

# Оглавление

# 1. Список терминов и сокращений

| Термин / Аббревиатура | Значение |
|---|---|
| ОС | Операционная система. |
| ffmpeg | Набор свободных библиотек с открытым исходным кодом, которые позволяют записывать, конвертировать и передавать цифровые аудио- и видеозаписи в различных форматах. |
| Redis (Remote Dictionary Service) | Сервер баз данных типа «ключ – значение» с открытым исходным кодом. Может использоваться как для формирования хранилищ данных, так и для реализации кэшей, потоковой обработки данных, брокеров сообщений. |
| S3 (Simple Storage Service) | Облачный сервис для хранения цифровых данных большого объема. |
| Система фиксации технических сбоев для терминалов (Система) | Программный продукт, предназначенный для предоставления отчетов о технических сбоях на терминалах в виде архива логов, снимков экрана (скриншотов), а также видеозаписей экрана терминала. |

# 2. О документе

Документ «Руководство по установке и первоначальной настройке программного продукта» содержит:

- порядок действий при установке Системы фиксации технических сбоев для терминалов (далее – Системы);
- порядок настройки рабочей среды для ввода Системы в эксплуатацию;
- перечень минимальных требований к аппаратной части;
- перечень сторонних программных продуктов, необходимых для корректной работы Системы;
- перечень сторонних программных продуктов, необходимых для модернизации Системы.

# 3. Введение

Терминальная сеть – это способ построения компьютерной сети, при котором вся информация, необходимая для работы, сосредоточена в терминальном сервере. Информация, находящаяся на сервере, доступна с большого количества устройств малой мощности – терминалов. Как правило, управление терминалами осуществляется централизованно.

При организации терминальной сети очень важно обеспечить возможность получать отчеты об ошибках и технических сбоях, возникающих в работе терминалов, для дальнейшего анализа и устранения причин часто встречающихся ошибок.

Программный продукт Система фиксации технических сбоев для терминалов предназначен для сохранения и отправки по запросу отчетов о сбоях в работе терминала в форме текстовых логов, снимков экрана (скриншотов) и видеозаписей экрана терминала.

# 4. Системные требования

В разделе «Системные требования» приведены минимальные системные требования к оборудованию, предназначенному для установки Системы фиксации технических сбоев для терминалов, а также требования к стороннему программному обеспечению.

## 4.1. Минимальные аппаратные требования

Для обеспечения стабильного функционирования Системы аппаратная часть должна обладать следующими характеристиками:

- Количество логических ядер процессора: 4;
- Семейство процессоров: x86-64;
- Частота процессора: 3.6 ГГц;
- Объем установленной памяти: 4 Гб.

## 4.2. Минимальные требования к сторонним компонентам и/или системам

Для обеспечения корректной работы Системы должны быть предварительно установлены следующие программные компоненты:

- Операционная система *Alpine Linux 3.15.9* (Лицензия GNU GPL 2);
- *Docker 20.10.16* (Лицензия Apache);
- *Redis server 6.2.13* (Лицензия BSD).

Видеохостинг запускается на отдельной виртуальной машине, на которой должна быть установлена ОС Debian 11 (Лицензия GNU).

В качестве хранилища данных используется хранилище S3.

Запись видео осуществляется с помощью библиотек *ffmpeg*.

Для разработки и модернизации Системы должны применяться языки программирования:

- *Python 3.11* (Лицензия Python Software Foundation License);
- *Golang 1.21.3* (Лицензия BSD).

# 5. Установка и настройка системы

Система включает в себя компоненты (Рис. 1):

- *Task_manager* – приложение для взаимодействия с терминалом со стороны инфраструктуры Заказчика;
- *Stream_manager* – приложение для записи видео с экрана терминала;
- *video.storage* – сервис для публикации и хранения видеозаписей.

Приложения *Task_manager* и *Stream_manager* запускаются на виртуальной машине, на которой должна быть установлена ОС *Alpine Linux 3.15.9*.

Сервис *video.storage* запускается на отдельной виртуальной машине с установленной ОС *Debian 11*.

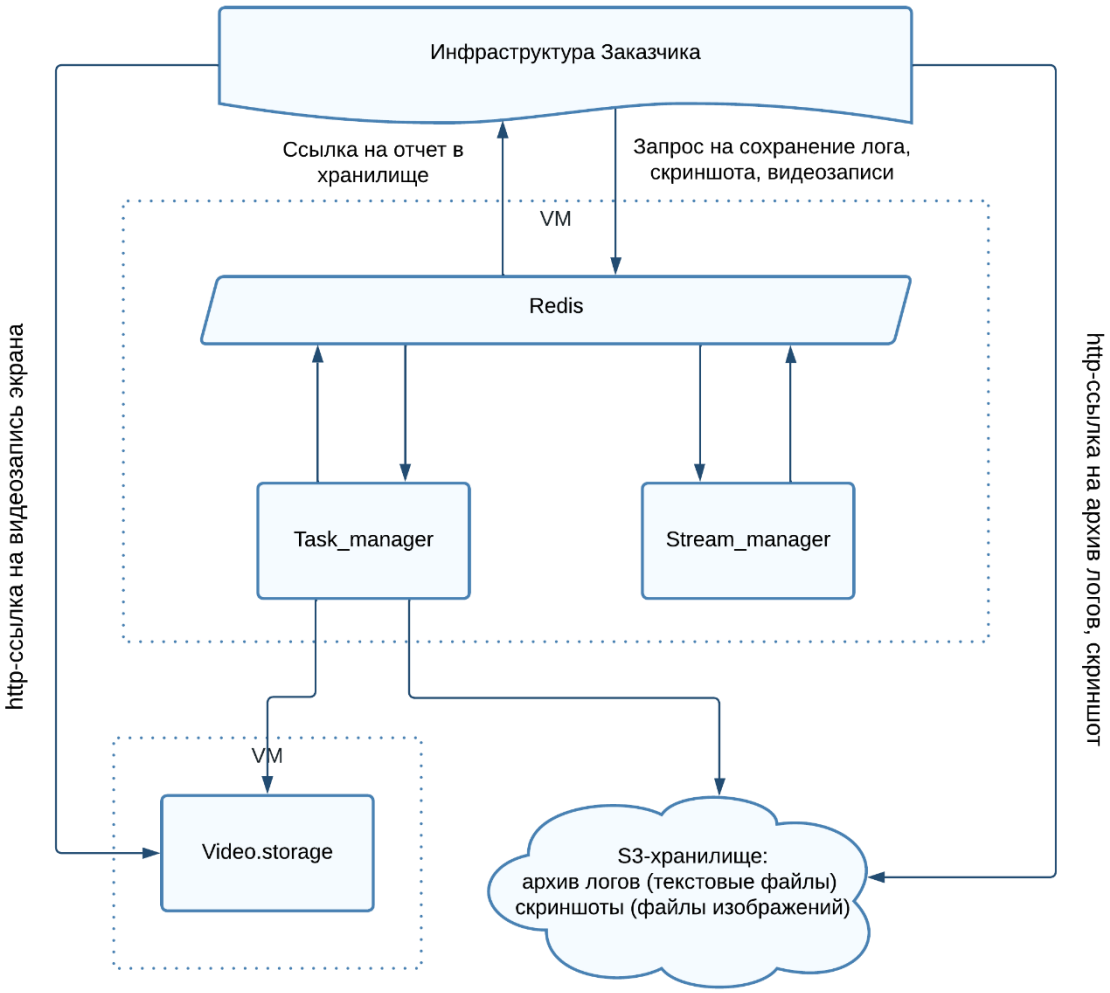


*Рис. 1 Компоненты Системы фиксации технических сбоев для терминалов*

Чтобы запустить программный продукт, необходимо:

1. Запустить виртуальную машину с приложениями task_manager и stream_manager (5.1);
2. Запустить сервис video.storage на отдельной виртуальной машине (5.2).

## 5.1. Запуск виртуальной машины с приложениями task_manager и stream_manager

Порядок действий для создания и запуска виртуальной машины:

1. скачать образ виртуальной машины:

```
wget http://terminal-ds-2.gbstd.ru/terminal.qcow2
```

2. запустить приложение virt-manager;
3. запустить мастер создания новой виртуальной машины: в окне приложения выбрать пункт меню File → New virtual machine (Рис. 2);

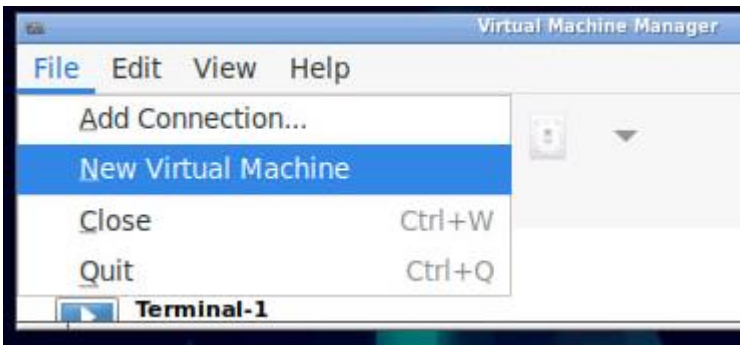


*Рис. 2 Создание новой виртуальной машины*

4. шаг 1 – выбрать ручной ввод настроек: в открывшемся окне выбрать пункт Manual Install и нажать на кнопку «Forward» (Рис. 3);

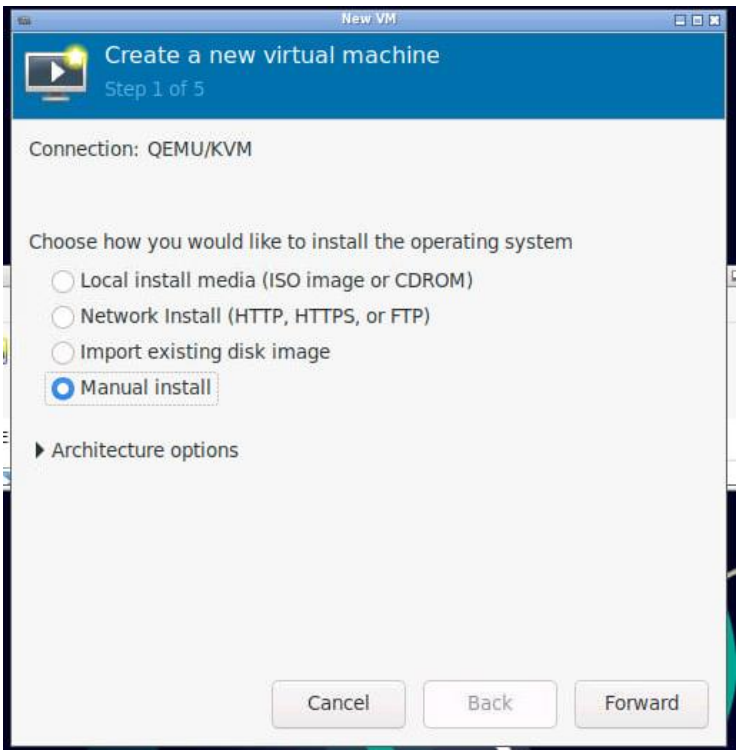


*Рис. 3 Выбор ручного ввода настроек*

5. шаг 2 – выбрать ОС: в поле «Choose the operating system you are installing» указать Generic Linux 2020 и нажать на кнопку «Forward» (Рис. 4);

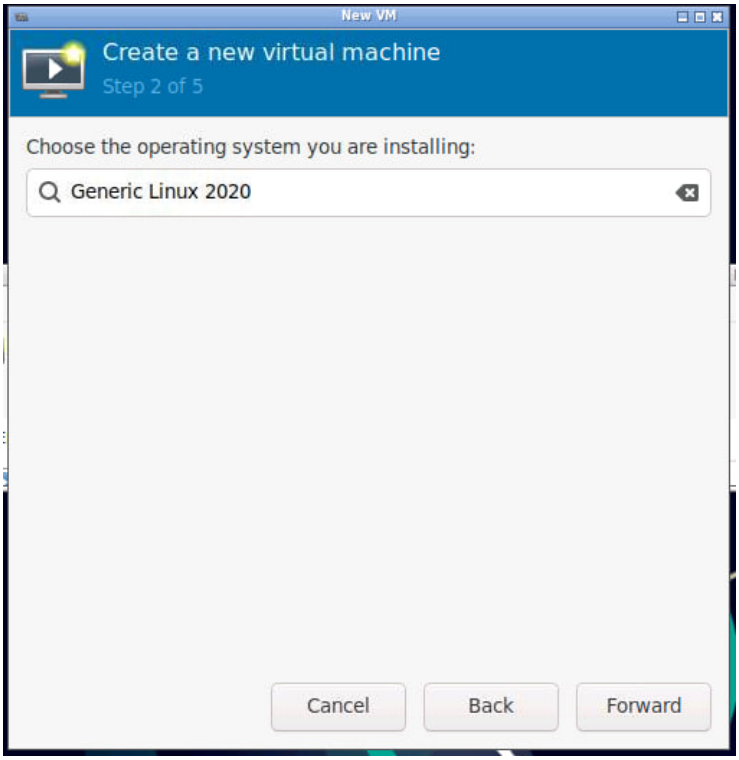


*Рис. 4 Выбор ОС*

6. шаг 3 – указать количество занимаемой памяти: в поле «Memory» установить значение 6096 и нажать на кнопку «Forward» (Рис. 5);

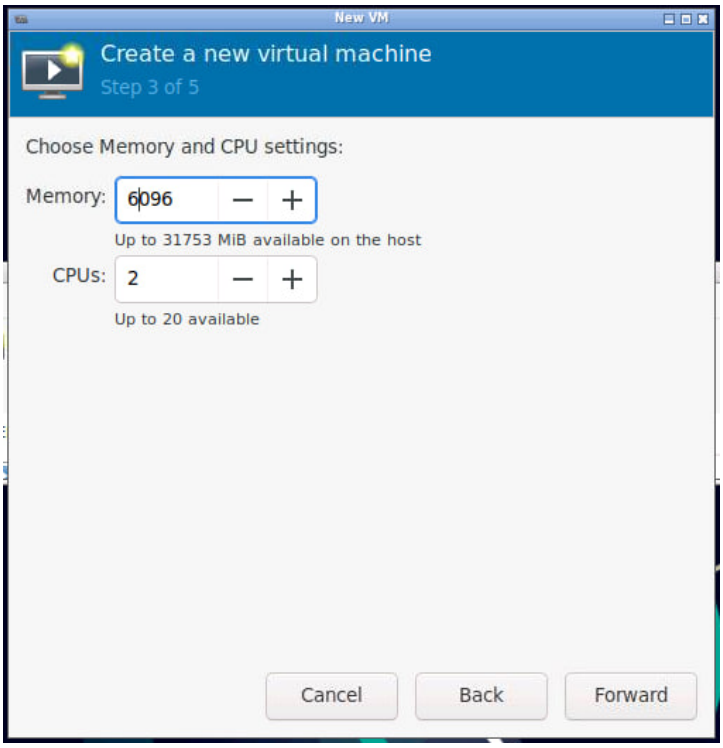


*Рис. 5 Количество занимаемой памяти*

7. шаг 4 – указать путь к скачанному образу:
   a. выбрать пункт «Select or create custom storage» и нажать на кнопку «Manage» (Рис. 6);
   b. в открывшемся окне нажать на кнопку «Browse Local» и указать путь к скачанному образу (Рис. 7). Окно закроется, в поле «Select or create custom storage» появится путь к выбранному образу (Рис. 8);
   c. нажать на кнопку «Forward»;

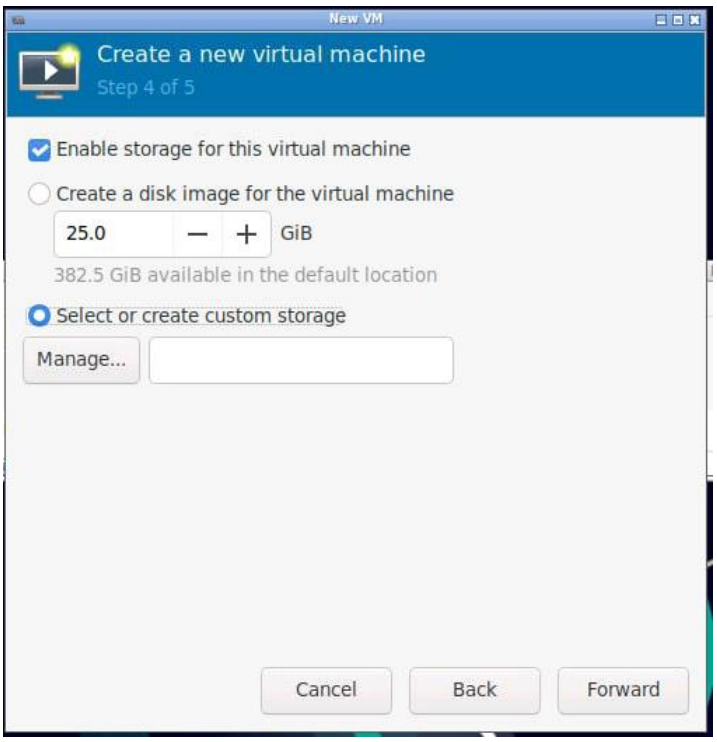


*Рис. 6 Выбор образа*

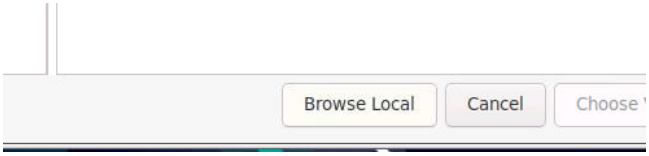


*Рис. 7 Выбор пути к файлу образа*

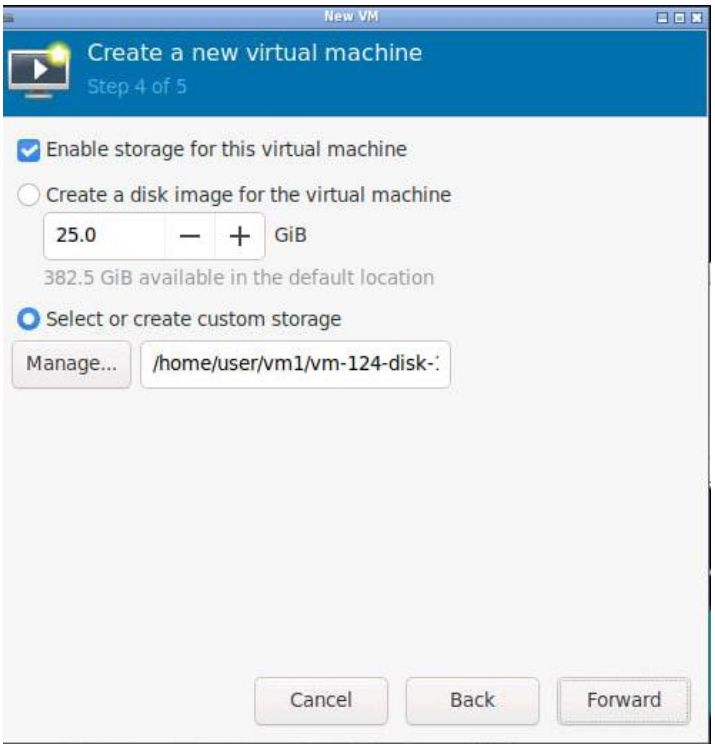


*Рис. 8 Путь к выбранному образу*

8. шаг 5 – ввести имя для новой виртуальной машины: ввести выбранное имя в поле «Name» (например, «Terminal-123») и нажать на кнопку «Finish» (Рис. 9);

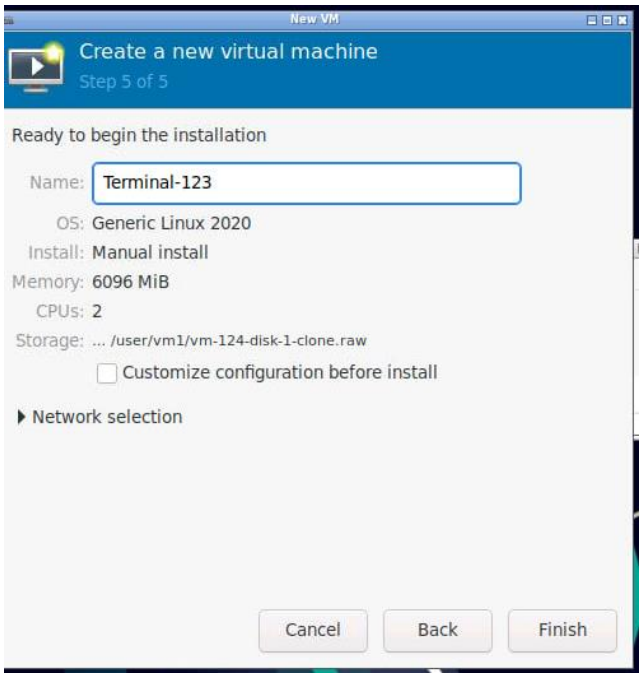


*Рис. 9 Проверка настроек и ввод имени виртуальной машины*

9. в настройках «Video Virtio» виртуальной машины указать модель «VGA» (Рис. 10);

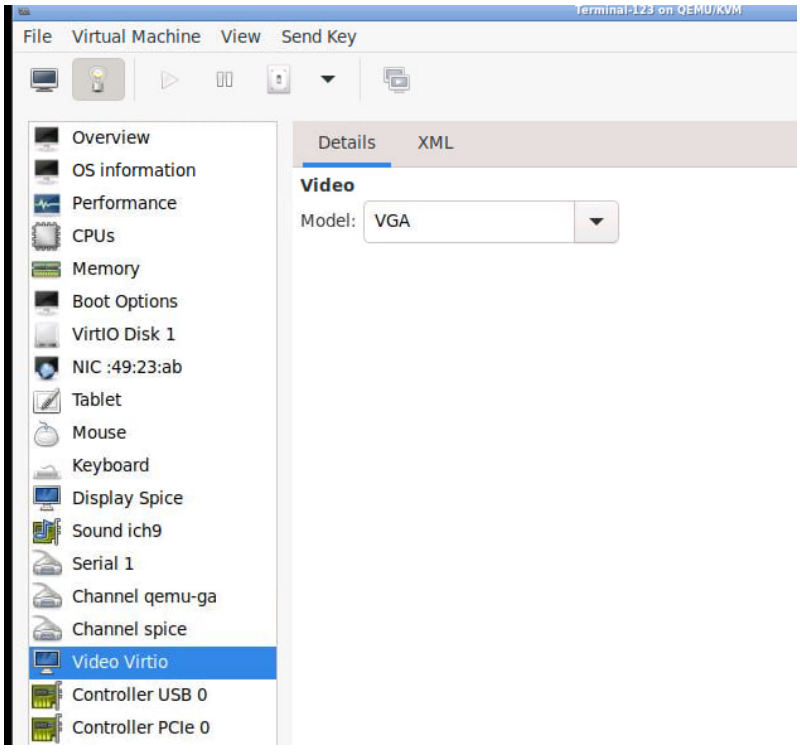


*Рис. 10 Выбор настроек Video*

10. в настройках «NIC» виртуальной машины указать модель интерфейса «e1000e» (Рис. 11);

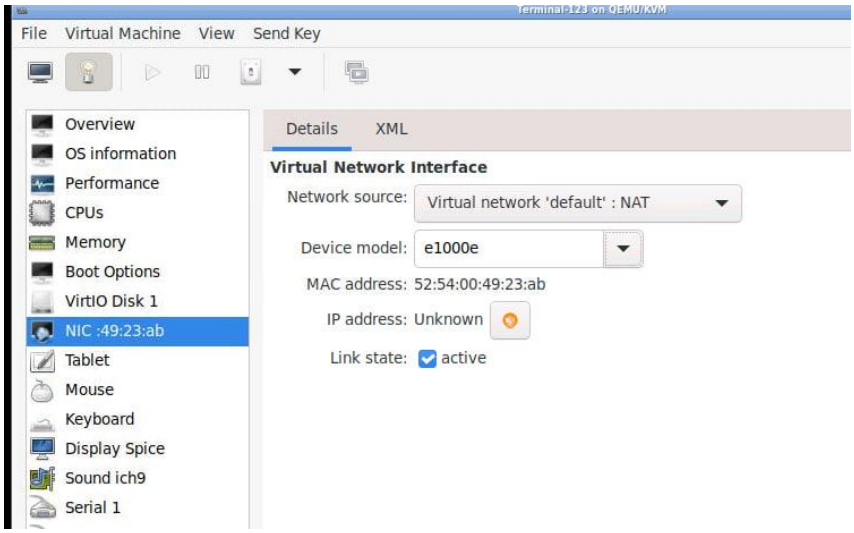


*Рис. 11 Выбор настроек NIC*

11. запустить виртуальную машину, нажав на кнопку ▷ (Рис. 12).

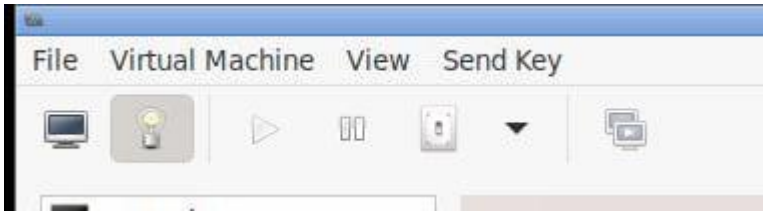


*Рис. 12 Запуск виртуальной машины*

## 5.2. Запуск сервиса video.storage

Сервис video.storage запускается на отдельной виртуальной машине, на которой должна быть установлена ОС Debian 11.

Для того, чтобы запустить сервис, необходимо:

1. установить необходимые пакеты:
   1.1. выполнить команды

```
sudo apt-get update && sudo apt-get install ca-certificates curl
gnupg nano
```

1.2. установить docker и docker-compose последней версии согласно инструкции, расположенной по адресу:

https://docs.docker.com/engine/install/debian/

2. создать папку для проекта и перейти в нее, выполнив команду

```
mkdir -p /project/video-storage/ && cd /project/video-storage/
```

3. настроить конфигурацию nginx:

3.1. создать папку для конфигурации nginx, выполнив команду

```
mkdir -p /project/video-storage/nginx
```

3.2. создать файл конфигурации по пути /project/video-storage/nginx/nginx-proxy.conf со следующим содержимым (<NAME_SERVER> необходимо заменить на имя fqdn-сервера video.storage):

```
server {

    listen              443 ssl;
    server_name         NAME_SERVER;
    ssl_certificate
/etc/letsencrypt/live/NAME_SERVER/fullchain.pem;
    ssl_certificate_key
/etc/letsencrypt/live/NAME_SERVER/privkey.pem;

    client_max_body_size 150M;

    # access_log        /var/log/nginx/video-storage/access.log;
    # error_log         /var/log/nginx/video-storage/error.log;
    location / {
        proxy_pass            http://hls-server:4001/;
        client_max_body_size  4000m;
        proxy_redirect        off;
        proxy_set_header X-Real-IP $remote_addr;
        proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
        proxy_set_header Host $http_host;
        proxy_pass_header Set-Cookie;
```

```
        satisfy any;
    }

    location /ping {
        return 200 "pong";
    }

}
```

4. создать файл переменных окружения по пути /project/video-storage/environments со следующим содержимым:

```
SENTRY_DSN=
KUBERNETES_NAMESPACE=video
GIT_SHORT_COMMIT=
LOG_LEVEL=info
APP_VERSION=APP_VERSION
LISTEN_HOST=0.0.0.0
LISTEN_PORT=3000
DEV_MODE=false
AUTH_TOKEN=AUTH_TOKEN
HLS_FILES_ROOT_PATH=/video-hls/
UPLOAD_CHUNK_SIZE_KB=64
FQDN=FQDN_SERVER
BASE_URL=https://FQDN_SERVER
HLS_URL_PATH=stream
```

где указаны значения переменных окружения:

- APP_VERSION – (str) версия приложения;
- LISTEN_HOST – (str) http listen host;
- LISTEN_PORT – (int) http listen port;
- AUTH_TOKEN – (str) токен для аутентификации запросов;
- HLS_FILES_ROOT_PATH – (str) базовый каталог для хранения hls файлов (.ts);
- UPLOAD_CHUNK_SIZE_KB – (str) размер чанка в Кб для загрузки бинарного файла (по умолчанию 64);
- BASE_URL – (str) базовый URL (для генерации ссылки на стрим);

- HLS_URL_PATH – (str) часть URL, указывающая на nginx stream location (https://foo.bar/{HLS_URL_PATH}/hash/stream.m3u8);

5. создать файл docker-compose.yml по пути /project/video-storage/docker-compose.yml со следующим содержимым (необходимо заменить NAME_SERVER на fqdn, а DOMAIN_NAME на имя домена сервера):

```
version: '3'
services:
  video-storage:
    env_file: environments
    image: "video-storage:master-latest"
    restart: unless-stopped
    container_name: video-storage
    volumes:
      - ./video-hls:/video-hls

  hls-server:
    env_file: environments
    image: "hls-server:master-latest"
    container_name: hls-server
    restart: unless-stopped
    ports:
      - "4001:4001"
    volumes:
      - ./video-hls:/video-hls
    depends_on:
      - video-storage

  nginx-certbot:
    env_file: environments
    environment:
      CERTBOT_EMAIL: admin@DOMAIN_NAME
      ENVSUBST_VARS: "NAME_SERVER"
    image: staticfloat/nginx-certbot
    container_name: nginx-certbot
    restart: unless-stopped
    ports:
```

```
      - "80:80/tcp"
      - "443:443/tcp"
    volumes:
      - ./nginx:/etc/nginx/user.conf.d:ro
      - ./www:/usr/share/nginx/html
      - letsencrypt:/etc/letsencrypt

volumes:
  letsencrypt:
```

6. запустить video.storage, выполнив команду:

```
cd /project/video-storage/ && docker-compose up -d
```