

РУКОВОДСТВО АДМИНИСТРАТОРА LINUX

ИНСТРУКЦИЯ ПО УСТАНОВКЕ
И СИСТЕМНЫЕ ТРЕБОВАНИЯ

ВЕРСИЯ 1.0

Содержание

1. Глоссарий	3
2. Введение	4
3. Системные требования для установки CheckU	5
4. Установка CheckU	6
4.1. Установка с использованием скрипта.....	6
4.1.1. Настройка файлов конфигурации.....	6
4.1.2. Использование скрипта.....	10
4.2. Установка вручную.....	13
4.2.1. Установка CheckU на сервер.....	13
4.2.2. Настройка nginx.....	13
4.2.3. Настройка сервиса.....	15

1. Глоссарий

Список терминов и сокращений, принятых в документе.

Термин	Описание
ИБ	Информационная безопасность
ПО	Программное обеспечение
СУБД	Система управления базами данных

2. Введение

Данное руководство предназначено для администратора решения контроля соответствиям требований ИБ (далее — CheckU), с навыками администрирования семейства операционных систем Linux и базовыми знаниями о сетях TCP/IP.

Руководство содержит: системные требования для установки CheckU и инструкции по установке.

3. Системные требования для установки CheckU

Серверная часть

Операционная система	<ul style="list-style-type: none">– Windows Server 2022, Windows 10 и более поздние версии– Ubuntu/Ubuntu Server 22.04 LTS, Debian 12, Astra Linux 1.7, RedOS 8 и более поздние версии
Процессор	2 ядра, 1.5 GHz+
Оперативная память	8 ГБ
Накопитель	1 ГБ+ SSD/HDD (без наполнения данными)
Runtime	ASP.NET Core Runtime 8.0 или выше (.NET 8)
СУБД	PostgreSQL 15-18 или JatoBa 5-8
Дополнительные компоненты	nginx/caddy

Клиентская часть

– **Браузер:** Chrome, Edge, Yandex Browser и другие браузеры на основе Chromium актуальной версии

4. Установка CheckU

Установка CheckU выполняется с помощью скрипта: [Установка с использованием скрипта \(6\)](#). Это рекомендуемый способ установки.

Кроме того, возможна установка в ручном режиме: [Установка вручную \(13\)](#).

4.1. Установка с использованием скрипта

Скрипт `deploy.sh` позволяет произвести установку, настройку и запуск необходимых компонентов. Однако это не отменяет подготовку: скрипт не устанавливает `nginx`, `.NET` и `СУБД`.

Скрипт создает необходимые конфигурации для `systemd` и `nginx` (если используется другой обратный прокси, его нужно будет настраивать отдельно), создает иерархию директорий

по указанному корневому пути и переносит все необходимые файлы для запуска, а также устанавливает необходимые разрешения.

Порядок установки с помощью скрипта:

1. Установка программных компонентов, см. [Системные требования для установки CheckU \(5\)](#)
2. [Настройка файлов конфигурации \(6\)](#)
3. [Использование скрипта \(10\)](#)

4.1.1. Настройка файлов конфигурации

CheckU имеет два файла конфигурации запуска:

1. [Настройка конфигурационного файла `appsettings.json` \(6\)](#)
2. [Настройка конфигурационного файла `appsettings_secret.json` \(8\)](#)

Оба файла конфигурации запуска должны находиться в корневом каталоге `backend`.

4.1.1.1. Настройка конфигурационного файла `appsettings.json`

Пример файла

```
{
  "Logging": {
    "LogLevel": {
      "Default": "Information",
```

```
"Microsoft.AspNetCore": "Warning"
}
},
"Kestrel": {
  "Endpoints": {
    "Http": {
      "Url": "http://[::]:5213"
    }
  }
},
"cors": {
  "addresses" : [
    "http://localhost:8080"
  ]
},
"Serilog": {
  "Using": [ "Serilog.Sinks.File"
], "MinimumLevel": {
  "Default": "Information",
  "Override": {
    "Microsoft": "Error",
    "System": "Error"
  }
},
"WriteTo": [
  {
    "Name": "File",
    "Args": {
      "path": "logs/log-.txt",
      "rollingInterval": "Day"
    }
  }
]
},
"Monitoring": {
  "UseMonitoring":
false,
  "Endpoint": "http://localhost:4317/",
```

```
"Name": "CheckU_metrics",
  "UsePrometheus": true
},
"AllowedHosts": "*",
"IsSqlite": false
}
```

Настройка файла

1. В секции `cors:addresses` укажите адреса и порты для внешних клиентов. Пример:

```
"cors": {
  "addresses" : [
    "http://localhost:8080",
    "http://10.20.30.40:4123"
  ]
}
```

2. В секции `Monitoring` укажите, будут ли использоваться метрики и телеметрия. Значение `"UseMonitoring":true` говорит о включенном сборе телеметрии, трейсинга и логирования во внешний коллектор. Если значение `true`, то в поле `Endpoint` укажите адрес, куда будет производиться сбор данных о внутреннем состоянии системы. А в поле `Name` укажите имя экземпляра. Важно, чтобы имена экземпляра не совпадали с именами других запущенных экземпляров.

4.1.1.2. Настройка конфигурационного файла

`appsettings_secret.json`

Пример файла

```
{
  "ConnectionStrings" : {
    "DefaultConnection" : "Server=localhost;Port=5432;Username=postgres;Password=postgres;Database=CheckU"
  },
  "JwtSettings": {
    "SecretKey": "SuperSecretKeyWith128BitsAndMoreThen256",
    "Issuer": "SuperIssuer",
    "Audience": "SuperAudience",
```

```
"ExpireTime": 15,
  "RefreshExpireTime" : 1
},
"DefaultAdminSettings": {
  "Name": "admin",
  "Password": "AdminP@ssw0rd"
},
"CatalogServices": {
  "Default": "ActiveDirectory",
  "ActiveDirectory": {
    "BaseDn": "OU=users,OU=exampse,DC=example,DC=ru",
    "SearchFilter": "(sAMAccountName={0})"
  },
  "FreeIpa": {
    "BaseDn": "dc=example,dc=com",
    "SearchFilter": "(uid={0})"
  }
},
"LdapConnection": {
  "Host": "10.20.19.17",
  "Port": 389,
  "UseSsl": true,
  "ProtocolVersion" : 3,
  "ServiceUser" : "user",
  "ServicePassword" : "password",
  "Timeout" : 10
},
"DocumentSettings": {
  "SavePath" : "./Documents",
  "TempPath" : "./Temp"
}
}
```

Настройка файла

1. Укажите строку подключения к базе данных в секции:

```
"ConnectionStrings" : {  
    "DefaultConnection" : [Your Connection String]  
}
```

2. В секции `JwtSettings` укажите:

- `SecretKey` – секретный ключ; должен быть не менее 128 бит, т. е. подойдет GUID
- `Issuer` – издателя
- `Audience` – получателя
- `ExpireTime` – время жизни токена авторизации (в минутах)
- `RefreshExpireTime` – время жизни токена обновления (в днях)

3. В секции `DefaultAdminSettings` укажите данные первоначального пользователя с правами администратора.

4. Если используется служба каталогов, то укажите настройки в секции `CatalogService`, соответствующие службе каталогов в доменной сети, а также данные подключения к службе каталогов в секции `LdapSettings`.

4.1.2. Использование скрипта

Содержимое директории `Deploy`

– `deploy.sh` – скрипт установки и запуска CheckU на сервере с SELinux и Debian based дистрибутивами

– `nginx.template.conf` – шаблон конфигурации nginx для клиента (frontend).

Используется `deploy.sh` для настройки конфигурации nginx. Должен быть расположен в той же директории, что и `deploy.sh`

– `systemd.template.service` – шаблон конфигурации для запуска бэкенда в виде systemd сервиса. Используется `deploy.sh` для настройки конфигурации systemd сервиса. Должен быть расположен в той же директории, что и `deploy.sh`

Запуск скрипта

1. Рядом со скриптом `deploy.sh` создайте папки `backend` и `frontend`.

2. Поместите артефакты бэкенда (файлы после выполнения команды `dotnet publish -c Release`) в папку `backend`.

3. Поместите артефакты клиента (содержимое папки dist после выполнения команды `yarn run build`) в папку frontend. Если в папке dist создается файл `.env`, удалите его. Он может быть невидим на linux-системах, поэтому проверьте с помощью `ls -la` в терминале.

4. Должна получиться следующая иерархия:

```
~/checku
├─ backend/
│  └─ DatapkLaptopServer.Startup
│  └─ appsettings.json
│  └─ appsettings_secret.json
│  └─ ... (другие backend артефакты)
├─ frontend/
│  └─ index.html
│  └─ assets/
│  └─ ... (другие frontend артефакты)
├─ nginx.template.conf
├─ systemd.template.service
└─ deploy.sh
```

5. Скопируйте все на целевой сервер.

6. Запустите скрипт с `sudo` правами, например:

```
sudo /tmp/checku/deploy.sh
```

Скрипт задаст вопросы касательно расположения артефактов CheckU.

Пример работы со скриптом

```
bash
administrator@checkudev-7:~$ sudo bash /tmp/checku/prod_deploy.sh
Detected Debian-like system (ubuntu)
CheckU root dir path: /var/checku
#означает, что корневая папка /var/checku

CheckU backend dir (relative to root): backend
#означает, что бэкенд будет располагаться в /var/checku/backend
```

```
CheckU frontend dir (relative to root): frontend
#означает, что клиент будет располагаться в /var/checku/frontend
```

Скрипт спросит, где находится бэкенд, т.к. клиент и фронтенд не привязаны друг к другу. А также администратор может изменить порт бэкенда. Рекомендуется оставлять значения по умолчанию, поэтому при запросе `Enter backend address` введите:

```
Enter backend address (e.g. http://localhost:5213):
http://localhost:5213
```

Скрипт скопирует артефакты в целевые директории, создаст `checku.conf` в `/etc/nginx/conf.d` и `checku.service` в `/etc/systemd/system/checku.service`, а также перезапустит `nginx` и запустит сервис `checku`. В случае, если все сделано правильно, вывод в терминал должен быть таким, учитывая введенные параметры:

```
administrator@checkudev-7:~$ sudo bash /tmp/checku/prod_deploy.sh
Detected Debian-like system (ubuntu)
CheckU root dir path: /var/checku
CheckU backend dir (relative to root): backend
CheckU frontend dir (relative to root): frontend
=== Starting deployment ===
=== Cleaning backend target ===
=== Cleaning frontend target ===
=== Copying backend files ===
=== Copying frontend files ===
=== Setting permissions ===
=== Creating nginx config ===
Enter backend address (e.g. http://localhost:5213): http://localhost:5213
Nginx config saved to /etc/nginx/conf.d/checku.conf
Nginx reloaded
=== Creating systemd config ===
Systemd unit saved to /etc/systemd/system/checku.service
Created symlink /etc/systemd/system/multi-user.target.wants/checku.service
→ /etc/systemd/system/checku.service.
=== Deployment completed successfully ===
```

Для полноценного запуска на сервере должны быть установлены .NET 8 SDK, .NET 8 Runtime, nginx и Postgres (15/16/17/18)/Jatoba (5/6/7/8). СУБД может быть запущена в docker

на том же или другом сервере. А также в папке с артефактами backend должны находиться конфигурационные файлы `appsettings.json` и `appsettings_secret.json`.

О настройке см. [Настройка файлов конфигурации \(6\)](#).

4.2. Установка вручную

Порядок установки вручную:

1. Установка программных компонентов, см. [Системные требования для установки CheckU \(5\)](#)
2. [Настройка файлов конфигурации \(6\)](#)
3. [Установка CheckU на сервер \(13\)](#)
4. [Настройка nginx \(13\)](#)
5. [Настройка сервиса \(15\)](#)

4.2.1. Установка CheckU на сервер

1. Скопируйте артефакты бэкенда на сервер. Например, в `/var/checku/backend`.
2. Создайте в директории бэкенда необходимую иерархию, например:

```
~/backend
├── Documents
├── NpaDocuments
└── ProofDocuments
```

3. Скопируйте артефакты фронтенда на сервер. Например, в `/var/checku/frontend`.

Фронтенд имеет следующую иерархию:

```
~/frontend
├── assets
├── fonts
├── images
└── index.html
```

4.2.2. Настройка nginx

1. Создайте файл `/etc/nginx/conf.d/checku.conf` с содержимым:

```
listen 8080;

# The server listens on port 8080

# Default server block for any hostname
```

```
server_name _;

# Root directory for static frontend files
root /var/checku/frontend;

index index.html;

# Default file to server
client_max_body_size 50M;

# Maximum allowed request body size

location /api/ {
# Forward all /api requests to backend service
    proxy_pass http://localhost:5213;
    proxy_http_version 1.1;
    # Use HTTP/1.1 for proxying

    proxy_set_header Host $host;
    # Preserve original Host header
    proxy_set_header X-Real-IP $remote_addr; # Pass real client IP
    # Forward chain of client IPs
    proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
}

location / {
    # Serve static files; fallback to index.html for SPA routing
    try_files $uri /index.html;
}
}
```

где:

- `listen` – прослушиваемый порт, через который будет доступен фронтенд
- `root` – путь к папке со статическими файлами фронтенда
- `proxy_pass` – адрес бэкенда



Примечание: рекомендуется запускать бэкенд на том же сервере, где расположены статические файлы фронтенда.

2. Перезапустите службу `nginx` с помощью команды:

```
sudo systemctl restart nginx
```

4.2.3. Настройка сервиса

1. Создайте файл `/etc/systemd/system/checku.service` с содержимым:

```
[Unit]
# Description shown in systemctl listings
Description=CheckU Server

# Ensure the service starts only after networking is available
After=network.target

[Service]
# Directory from which the service will run
WorkingDirectory=/var/checku/backend

# Command used to start the application; binds to port 5213 on all interfaces
# (IPv6-compatible)
ExecStart=/var/checku/backend/DatapKlaptopServer.Startup --urls http://[::]:5213

# Restart the service automatically if it stops
Restart=always

# Wait 5 seconds before attempting a restart
RestartSec=5

# Environment variable for ASP.NET Core runtime
Environment=ASPNETCORE_ENVIRONMENT=Production

[Install]
# Start automatically when the system reaches multi-user mode
WantedBy=multi-user.target
```

2. Перезапустите службы с помощью команды:

```
sudo systemctl daemon-reload
```

3. Включите службу CheckU:

```
sudo systemctl enable checku
```

4. Запустите службу CheckU:

```
sudo systemctl start checku
```

Либо:

```
sudo systemctl start checku
```