

7(4712) 26-99-00
www.dcatk.pro
info@dcatk.pro

АТК

**Инструкция по установке экземпляра программного обеспечения,
предоставленного для проведения экспертной проверки**

Tronic Messenger — это корпоративная система мгновенных сообщений с интегрированными AI-сервисами, включающая в себя:

Основные компоненты:

- OpenFire — кастомное XMPP-ядро с поддержкой OAuth 2.0 для работы с SSO, с дополнительной поддержкой кастомных stanz для управление сообщениями и реакциями
- Asterisk — VoIP-сервер с поддержкой PostgreSQL Real-time Database
- PostgreSQL — основная база данных для хранения конфигураций и данных
- Messenger Client — веб-интерфейс для пользователей
- Asterisk Dynamic PJSIP Users — сервис динамического управления пользователями VoIP

AI-сервисы:

- AI Quick Answers — сервис быстрых ответов на базе ИИ
- Bot AI — ИИ-бот для автоматизации диалогов
- SST (Speech-to-Text) — сервис распознавания речи с поддержкой Celery и Redis
- TTS (Text-to-Speech) — сервис синтеза речи
- OCR (Optical Character Recognition) — сервис оптического распознавания текста

Дополнительные компоненты для SST:

- Redis — брокер сообщений для очереди задач
- Celery Workers — обработчики фоновых задач для SST

2. Системные требования

- ОС: Ubuntu 22.04+ или CentOS 8+
- Память на диске: От 300 Гб (рекомендуется SSD для AI-сервисов)
- Оперативная память: От 64 Гб (AI-сервисы требуют значительных ресурсов)
- CPU: 16+ ядер (рекомендуется для обработки AI-задач)
- GPU: Рекомендуется для ускорения AI-операций (опционально)
- Сеть: Статический IP-адрес, пропускная способность от 1 Гбит/с
- Docker: версия 20.10+
- Docker Compose: версия 2.0+

3. Подготовка к установке

3.1. Установка Docker и Docker Compose

Обновление системы

```
sudo apt update && sudo apt upgrade -y
```

Установка Docker

```
curl -fsSL https://get.docker.com -o get-docker.sh
```

```
sudo sh get-docker.sh
```

Добавление пользователя в группу docker

```
sudo usermod -aG docker $USER
```

```
# Установка Docker Compose
sudo curl -L "https://github.com/docker/compose/releases/latest/download/docker-compose-$(uname -s)-$(uname -m)" -o /usr/local/bin/docker-compose
sudo chmod +x /usr/local/bin/docker-compose
# Перезагрузка для применения изменений
sudo reboot
```

3.2. Запуск сервисов

```
cd /opt
git clone https://git.svgun.ru/cloud-infrastructure/tronic-messenger.git
# Переход в рабочую директорию
cd /opt/tronic-messenger
```

Создайте следующие файлы конфигурации:

3.2.1. Файл `postgres.env`

```
POSTGRES_DB=tronic_messenger
POSTGRES_USER=tronic_user
POSTGRES_PASSWORD=your_secure_password_here
POSTGRES_HOST=db
POSTGRES_PORT=5432
```

3.2.2. Файл `keycloak.env`

```
KEYCLOAK_URL=http://your-keycloak-server:8080
KEYCLOAK_REALM=tronic-messenger
KEYCLOAK_CLIENT_ID=messenger-client
KEYCLOAK_AUDIENCE=account
```

3.2.4. Файл `prod.env` (для AI-сервисов)

```
# Основные настройки
AI_MODEL=gemma3
AI_URL=https://llm.domain
AI_API_KEY=sk-ZC6juEqANx1uGSmAhqSBRg
API_KEY=sk-u3GNfqJvpPUbszuFBhaDrA
# Настройки бота
BOT_ID=llama_bot@openfire.domain
BOT_PASSWORD=
```

```
ENABLED_SITES=["http://localhost","https://localhost"]
USE_GPU=False
#optional
KEYCLOAK_ISSUER=https://keycloak.tes.com/realms/main
KEYCLOAK_AUDIENCE=account
MODEL=tts_models/en/vctk/vits
SPEAKER_ID=p227
DEFAULT_LANG=en
TRAINER_TELEMETRY=0
```

3.2.5. Файл `api/.env` (для SST сервиса)

```
# Базовые URL
BASE_URL=https://sst.domain
WEB_URL=https:// sst.domain
# Настройки PostgreSQL
POSTGRES_HOST=database
POSTGRES_PORT=5432
POSTGRES_USER=user
POSTGRES_PASSWORD=password
POSTGRES_DB=alpha
# Настройки Redis
REDIS_HOST=redis
REDIS_PORT=6379
# Конфигурация Whisper
MODEL=tiny
# Аутентификация Ollama
OLLAMA_USER_AUTH=ollama
OLLAMA_PASSWORD_AUTH=ollama_secret
# Настройки Keycloak
KEYCLOAK_REALM=my_realm
KEYCLOAK_URL=https://auth.example.com
KEYCLOAK_CLIENT_ID=app_client
KEYCLOAK_CLIENT_SECRET=very_secure_secret
# Дополнительные настройки
```

```
ENABLED_SITES=https://element.tronic.sndi.my
DELETE_FILES=True
MAX_DOWNLOAD_SIZE=200000000
```

3.2.6. Файл config.js (для веб-клиента)

```
var CONFIG = {
  VITE_BASE_URL: '/api/xmpp/plugins/restapi/v1',
  VITE_XMPP_URL: 'wss://openfire.my.domain/ws',
  OPENFIRE_URL: 'openfire.my.domain',
  VITE_SSO_URL: 'https://ssopwa.my.domain',
  STT_URL: '/stt_url',
  ASTERISK_URL: 'stun.svgun.ru',
  TTS_URL: '/api/tts',
  OCR_API_URL: '/ocr_url',
  VITE_NOTIFY_DOMAIN: 'https://notification.my.domain',
  QUICK_ANSWERS: 'https://element.my.domain/ai/answer',
  ONLYOFFICE_SERVER_URL: 'https://onlyoffice.my.domain',
  FILE_STORE_API: 'https://api-filestorage.my.domain',
  TRANSLATE_URL:
'https://translate.sndi.my/api/libretranslate/translate',
  VITE_KEYCLOAK_DOMAIN: 'https://keycloak.my.domain',
  VITE_KEYCLOAK_GROUP: 'master',
  VITE_KEYCLOAK_CLIENT_ID: 'ssopwa',
  VITE_KEYCLOAK_REDIRECT_DOMAIN: 'https://ssopwa.my.domain',
  VITE_SERVICE_ELEMENT: 'https://element.my.domain',
  VITE_SERVICE_SETTINGS: 'https://settings.my.domain',
  VITE_SERVICE_PWA: 'https://pwa.my.domain',
  VITE_SERVICE_STORAGE_DASHBOARD: 'https://pwa.my.domain',
  VITE_NAME_STORAGE_DASHBOARD: 'Storage dashboard',
}
```

3.3 Запуск

```
# Загрузка образов
```

```
docker-compose pull
```

```
# Запуск сервисов
```

```
docker-compose up -d
```

4.3. Проверка состояния сервисов

```
# Проверка статуса контейнеров
```

```
docker-compose ps
```

```
# Просмотр логов
```

```
docker-compose logs -f
```

```
# Проверка логов конкретного сервиса
```

```
docker-compose logs -f openfire
```

5. Настройка компонентов

5.1. Настройка OpenFire

1. Откройте веб-интерфейс OpenFire: `http://your-domain.com:9090`
2. Пройдите процедуру первоначальной настройки
3. Настройте подключение к PostgreSQL

5.2. Настройка Asterisk

1. Убедитесь, что конфигурационные файлы находятся в `./etc/asterisk/`
2. Проверьте подключение к базе данных через ODBC
3. Настройте Real-time конфигурацию

6. Доступ к сервисам

После успешного запуска сервисы будут доступны по следующим адресам:

Основные сервисы:

- **Messenger Client:** `http://your-domain.com:8080`
- **OpenFire Admin:** `http://your-domain.com:9090`
- **Adminer (DB Admin):** `http://your-domain.com:9991`
- **Asterisk Dynamic Users API:** `http://your-domain.com:9992`
- **PostgreSQL:** `your-domain.com:5433`

AI-сервисы:

- **AI Quick Answers API:** `http://your-domain.com:7779`
- **TTS (Text-to-Speech) API:** `http://your-domain.com:7778`
- **SST (Speech-to-Text) API:** `http://your-domain.com:8888`
- **OCR (Optical Character Recognition) API:** `http://your-domain.com:8787`

7. Проверка работоспособности

8. Устранение неполадок

8.1. Проблемы с базой данных

Если PostgreSQL не запускается:

```
# Проверка прав доступа
ls -la data/postgres/
chmod 750 data/postgres/.pgdata
```

```
# Очистка PID файлов
rm -f data/postgres/.pgdata/postmaster.pid
```

8.2. Проблемы с OpenFire

Если OpenFire не подключается к базе данных:

- Проверьте настройки подключения в веб-интерфейсе
- Убедитесь, что пользователь `openfire_user` создан в PostgreSQL
- Проверьте сетевое подключение между контейнерами

8.3. Проблемы с AI-сервисами

SST (Speech-to-Text) сервис:

Если SST не обрабатывает файлы:

Проверка Redis соединения

```
docker-compose exec api redis-cli -h redis ping
```

Проверка Celery воркеров

```
docker-compose exec celery_tasks celery -A celery_tasks inspect stats
```

Проверка логов обработки

```
docker-compose logs -f celery_tasks
```

TTS (Text-to-Speech) сервис:

Если TTS не генерирует аудио:

- Проверьте настройки модели в `prod.env`
- Убедитесь в доступности AI API
- Проверьте ограничения по размеру текста

OCR сервис:

Если OCR не распознает текст:

- Проверьте качество изображения
- Убедитесь в правильности языковых настроек
- Проверьте поддерживаемые форматы изображений

11. Контакты и поддержка

При возникновении проблем обращайтесь к документации или в службу технической поддержки.

Среднее время развертывания системы: 3-4 часа