

## ПО для автоматического обнаружения потенциально тонущих людей в бассейне с помощью "технического зрения"

### Инструкция по установке экземпляра программного обеспечения

#### Основные технические требования

Для работы системы (клиент и сервер, запущенные на разных машинах) необходимо чтобы машины, на которых запущены компоненты, были в одной локальной сети.

Ожидается что на клиент и сервер установлена операционная система Ubuntu 22.04 с поддержкой графического интерфейса пользователя.

#### Загрузка исходных кодов

Перед сборкой образа необходимо подготовить исходные коды приложения. Коды предоставляются в виде архива.

Необходимо распаковать архив в корневую папку пользователя: тогда исходные коды будут лежать в папке: `/home/<имя пользователя>/workspace/`

#### Общие сведения об установке

Предполагается, что клиент и сервер работают в виде контейнеров `docker` ([www.docker.com](http://www.docker.com)). Такой подход позволяет изолировать рабочую программную среду, легко и быстро выполнять развёртывание, настройку и архивирование состояния системы. Работа ПО возможно и без использования `docker`, но такой вариант требует ручного повторения действий, которые записаны в `dockerfile`'е - файле с последовательностью действия для сборки образа.

Для сборки образов должна быть обязательно соблюдена следующая иерархия папок:

- `< произвольная папка, далее workspace >`
  - `CVSPoolSafety` (исходные коды системы, загруженные файлы onnx-моделей нейронных сетей)
    - `tools`
      - `PoolPtzControl` (исходные коды утилиты управления светом)
  - `configs`

В папке `configs` должны лежать конфигурационные файлы приложения (`client_config.json` и/или `server_config.json`, имена должны быть строго такими).

Папка `configs` может отсутствовать, в этом случае при установке скрипт автоматически создаст ее и положит туда файлы из папки `CVSPoolSafety/configs`.

Далее необходимо открыть консоль в папке `.../CVSPoolSafety/installation`

При первой установке на хосте с чистой системой, на котором будет устанавливаться сервер, или сервер вместе с клиентом необходимо выполнить скрипт:

```
sh ./configure_host_server.sh
```

Если будет устанавливаться только клиент, то необходимо выполнить:

```
sh ./configure_host_client.sh
```

**Важно!** Предполагается, что скрипт выполняется либо на чистой системе, либо на системе, где ранее выполнялся запуск этого скрипта!

Этот скрипт установит необходимые зависимости, а также (в случае `configure_host_server.sh`) сконфигурирует локальный `postgres` сервер.

**Важно:** будет установлен `postgres` и создана БД для проекта. Скрипт проверяет наличие `postgres` в системе, и наличие БД, но не умеет разрешать ситуации когда что-то из этого есть, но права или структура БД не те что ожидаются.

Если сохранение уже существующих БД на хосте не важно, то можно перед запуском скрипта выполнить команду удаления уже установленного `postgres` с ключом `--purge`:

```
sudo apt remove --purge postgresql-14 postgresql-client-14 postgresql-contrib-14 postgresql-server-dev-14
```

## Запуск скриптов установки

Для установки и клиента и сервера необходимо выполнить:

```
sh ./install_all.sh
```

Для установки только сервера необходимо выполнить:

```
sh ./install_server.sh
```

Для установки только клиента необходимо выполнить:

```
sh ./install_client.sh
```

Для установки только журнала необходимо выполнить:

```
sh ./install_journal.sh
```

После выполнения выбранного скрипта сервер, клиент или они оба будут установлены и могут быть запущены как сервисы (или запускаются автоматически после перезагрузки хоста).

## Обновление

При необходимости обновления, следует обновить файлы в папке CVSPoolSafety и запустить скрипт `install_server.sh`, или `install_client.sh`, или `install_all.sh`. Существующие сервисы будут автоматически остановлены, контейнеры удалены. Ручное вмешательство может потребоваться только в случае, если приложения сервера или клиента запущены из консоли. В этом случае попытка удаления контейнера может потерпеть неудачу и будет необходимо вручную остановить соответствующий контейнер и повторно запустить скрипт установки.

## Установка ключа лицензии

Для запуска сервера необходимо установить корректный лицензионный ключ. Для этого требуется открыть файл `<workspace>/configs/server_config.json` в секции "license", значение поля "code" на актуальный ключ как показано на рисунке ниже.

```
32 |     "license" : {  
33 |         "code" : "D1750341-5EA24C4E-856715D3-3A5259C3"  
34 |     },
```

## Запуск

После перезагрузки компьютера система запускается автоматически.

Далее будет описано использование скриптов для запуска/остановка сервера и клиента:

Запуск сервера как сервиса:

```
sh ./start_server_service.sh
```

Останов сервиса сервера (временный, до перезагрузки):

```
sh ./stop_server_service.sh
```

Запуск сервера с открытой консолью:

```
sh ./start_server_console.sh
```

Подключение к уже запущенному контейнеру сервера отдельной консолью:

```
sh ./attach_server_container.sh
```



**Запуск клиента как сервиса:**

```
sh ./start_client_service.sh
```

**Останов сервиса клиента (временный, до перезагрузки):**

```
sh ./stop_client_service.sh
```

**Запуск клиента с открытой консолью:**

```
sh ./start_client_console.sh
```

**Подключение к уже запущенному контейнеру клиента отдельной консолью:**

```
sh ./attach_client_container.sh
```

**Запуск журнала как сервиса:**

```
sh ./start_journal_service.sh
```

**Останов журнала клиента (временный, до перезагрузки):**

```
sh ./stop_journal_service.sh
```

**Запуск журнала с открытой консолью:**

```
sh ./start_journal_console.sh
```

**Подключение к уже запущенному контейнеру журнала отдельной консолью:**

```
sh ./attach_journal_container.sh
```