



ООО «ЗИАКС»  
410028, Россия, г. Саратов, ул. Вольская, 28, пом. 2А  
+7 (495) 108-70-01  
[info@ziaх.ru](mailto:info@ziaх.ru); [ziaх.ru](http://ziaх.ru)

## Инструкция по установке ZIAХ ASR (Документация по развертыванию контейнеров MRCP/ASR)

## СОДЕРЖАНИЕ

<b>1. ПОДГОТОВКА СИСТЕМНОГО ОКРУЖЕНИЯ.....</b>	<b>3</b>
<b>2. МОДЕЛЬ РАСПОЗНАВАНИЯ.....</b>	<b>3</b>
<b>3. ФАЙЛ DOCKER-COMPOSE.YML.....</b>	<b>3</b>
<b>4. ЗАПУСК КОНТЕЙНЕРОВ.....</b>	<b>5</b>
<b>5. РЕСУРСЫ ДЛЯ ПРОВЕРКИ РАБОТОСПОСОБНОСТИ КОНТЕЙНЕРОВ.....</b>	<b>5</b>
<b>6. ПРОВЕРКА РАБОТОСПОСОБНОСТИ НА УРОВНЕ КОНТЕЙНЕРА СЕРВЕРА РАСПОЗНАВАНИЯ (ASR).....</b>	<b>7</b>
<b>7. ПРОВЕРКА РАБОТОСПОСОБНОСТИ НА УРОВНЕ КОНТЕЙНЕРА СЕРВЕРА UNIMRCP.....</b>	<b>7</b>

## 1. ПОДГОТОВКА СИСТЕМНОГО ОКРУЖЕНИЯ

Предполагается, что **docker engine** установлен и запущен. В противном случае, можно воспользоваться следующей документацией [Install Docker Engine](#).

Также предполагается, что **docker-compose** установлен на систему пользователя. В противном случае, можно воспользоваться следующей документацией [Install Docker Compose](#).

Развертывание контейнеров, модели и примеров будет производиться в директорию **/opt/ziax**.

Каждую серию команд, приведенную в документации, можно выполнить самостоятельно, так как они описаны в избыточной форме (действия могут повторяться несколько раз).

Следующие команды создают директорию **/opt/ziax** и задают на неё соответствующие права (текущего пользователя).

```
$ sudo mkdir -p /opt/ziax
$ sudo chown $(id -u $USER):$(id -g $USER) -R /opt/ziax
```

## 2. МОДЕЛЬ РАСПОЗНАВАНИЯ

Для работы контейнера **asr** потребуется модель распознавания, которая будет расположена в директории **/opt/ziax/models**.

Получить демонстрационную модель (для которой гарантировано работают приведённые примеры и **docker-compose.yml**) можно выполнив следующую серию команд, которые создают директорию **/opt/ziax/models**, загружают и распаковывают архив с моделью распознавания **tiny-model**.

```
$ mkdir -p /opt/ziax/models
$ cd /opt/ziax/models
$ wget https://hub.ziax.ru/models/tiny-model.tar.gz
$ tar -xzf tiny-model.tar.gz
```

## 3. ФАЙЛ DOCKER-COMPOSE.YML

Ниже представлен простейший вариант файла **docker-compose.yml**, который описывает два контейнера **asr** и **mrcp**, образы которых могут быть получены с ресурса **hub.ziax.ru:5000**.

Для контейнера **asr** в значение **volumes** задается расположение модели распознавания **/opt/ziax/models/tiny-model**, которое всегда должно ассоциироваться с **/opt/asr-models/model** в контейнере **asr**.

Для контейнера **mrcp** в значение **depends\_on** задано имя контейнера **asr**. Данная настройка задействует порядок запуска контейнеров (вначале запускается **asr**, затем **mrcp**) и гарантирует их корректную работу.

```
version: '3'

services:
  asr:
    image: hub.ziax.ru:5000/asr-grpc-server:latest
    container_name: asr
    hostname: asr
    restart: always
    volumes:
      - /opt/ziax/models/tiny-model:/opt/asr-models/model

  mrcp:
    image: hub.ziax.ru:5000/ziaxmrcp:latest
    container_name: mrcp
    hostname: mrcp
    restart: always
    entrypoint: ["/usr/local/unimrcp/bin/unimrcpserver", "-r", "/usr/local/unimrcp", "-o",
"2", "-w"]
    depends_on:
      - asr
```

Ниже представлен способ получения `docker-compose.yml` с ресурса [hub.ziax.ru](http://hub.ziax.ru), который размещается в директорию `/opt/ziax/docker` после её предварительного создания.

```
$ mkdir -p /opt/ziax/docker
$ cd /opt/ziax/docker
$ wget https://hub.ziax.ru/docker/docker-compose.yml
```

Также можно воспользоваться альтернативным вариантом создания файла `docker-compose.yml`

```
$ mkdir -p /opt/ziax/docker
$ cat << EOF > /opt/ziax/docker/docker-compose.yml
version: '3'

services:
  asr:
    image: hub.ziax.ru:5000/asr-grpc-server:latest
    container_name: asr
    hostname: asr
    restart: always
    volumes:
      - /opt/ziax/models/tiny-model:/opt/asr-models/model

  mrcp:
    image: hub.ziax.ru:5000/ziaxmrcp:latest
    container_name: mrcp
    hostname: mrcp
    restart: always
```

```
entrypoint: ["/usr/local/unimrcp/bin/unimrcpsrv", "-r", "/usr/local/unimrcp", "-o",  
"2", "-w"]  
depends_on:  
  - asr  
EOF
```

## 4. ЗАПУСК КОНТЕЙНЕРОВ

После создания или получения `docker-compose.yml` можно переходить к запуску контейнеров.

```
$ cd /opt/ziax/docker  
$ docker-compose up -d
```

Проверить успешность запуска можно, выполнив команду `docker ps`, которая отобразит список запущенных контейнеров.

```
$ docker ps  
CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES  
ecedf22e37f3 hub.ziax.ru:5000/ziaxmrcp:latest "/usr/local/unimrcp/..." 51 seconds  
ago Up 50 seconds mrcp  
39437746042c hub.ziax.ru:5000/asr-grpc-server:latest "python3.9 ./stt_ser..." 51  
seconds ago Up 50 seconds 5001/tcp asr
```

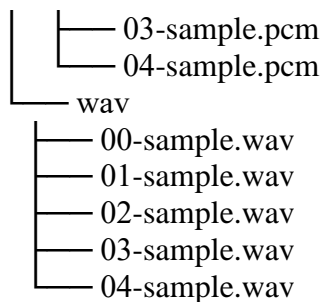
## 5. РЕСУРСЫ ДЛЯ ПРОВЕРКИ РАБОТОСПОСОБНОСТИ КОНТЕЙНЕРОВ

Данный и последующие разделы являются не обязательными, но могут быть выполнены для проверки базовой работоспособности контейнеров.

Данный архив `samples.tar.gz` содержит несколько `*.wav` файлов в формате `LINEAR16_PCM` с частотой дискретизации `8000 Hz` с помощью которых можно осуществить процесс распознавания на `gRPC`-сервере (`asr`). Также в архиве содержатся несколько `*.pcm` файлов (которые являются копией `*.wav`, но не содержатся `WAVE`-заголовков), с помощью которых можно осуществить процесс распознавания с `MRCP`-сервера.

Ниже представлена серия команд для получения примеров записей и результатов выполнения команды `tree`, отображающий структуру файлов.

```
$ cd /opt/ziax  
$ wget https://hub.ziax.ru/samples.tar.gz  
$ tar -xzvf samples.tar.gz  
$ tree samples  
samples  
├── pcm  
│   ├── 00-sample.pcm  
│   ├── 01-sample.pcm  
│   └── 02-sample.pcm
```



Архив [umc-scenarios.tar.gz](#) содержит несколько **\*.xml** файлов, которые описывают сценарий распознавания, и необходимы для корректной работы клиента **umc**.

Ниже представлена серия команд для получения примеров сценариев и результатов выполнения команды **tree**, отображающий структуру файлов.

```
$ cd /opt/ziax
$ wget https://hub.ziax.ru/umc-scenarios.tar.gz
$ tar -xzvf umc-scenarios.tar.gz
$ tree umc-scenarios
umc-scenarios
├── 00-recog.xml
├── 01-recog.xml
├── 02-recog.xml
├── 03-recog.xml
└── 04-recog.xml
```

Пример содержимого **umc**-сценария, который описывает его имя **00-recog** и файл, с которым оно проассоциировано **00-sample.pcm**.

```
<?xml version="1.0" encoding="UTF-8"?>
<umcscenario name="00-recog" class="Recognizer" profile="uni2">
  <resource-discovery enable="0"/>
  <define-grammar enable="0" content-type="application/srgs+xml" content-
location="grammar.xml"/>
  <recognize enable="1" audio-source="00-sample.pcm"/>
  <termination enable="1">
    <capabilities>
      <codec name="LPCM" rates="8000"/>
    </capabilities>
  </termination>
</umcscenario>
```

## 6. ПРОВЕРКА РАБОТОСПОСОБНОСТИ НА УРОВНЕ КОНТЕЙНЕРА СЕРВЕРА РАСПОЗНАВАНИЯ (ASR)

Перед началом выполнения тестов полученные примеры необходимо скопировать в контейнер **asr**.

```
$ docker cp /opt/ziax/samples/wav/. asr:/opt/asr-server/grpc
```

После копирования примеров необходимо зайти внутрь контейнера **asr** и запустить интересующий пример с помощью сценария **stt\_client.py**.

```
$ docker exec -it asr bash
$ ./stt_client.py --path 00-sample.wav
chunk:
alternative: передать показания счётчика
final: True
end_of_utterance: True
```

## 7. ПРОВЕРКА РАБОТОСПОСОБНОСТИ НА УРОВНЕ КОНТЕЙНЕРА СЕРВЕРА UniMRCP

Перед началом выполнения тестов полученные примеры и сценарии тестирования необходимо скопировать в контейнер **mrsp**.

```
$ docker cp /opt/ziax/samples/pcm/. mrsp:/opt/unimrmp/data
$ docker cp /opt/ziax/umc-scenarios/. mrsp:/opt/unimrmp/conf/umc-scenarios/
```

После копирования примеров и сценариев тестирования необходимо зайти внутрь контейнера **mrsp** и запустить UniMRCP клиента (**umc**) для их выполнения.

```
$ docker exec -it mrsp bash
$ cd /opt/unimrmp/bin
$ ./umc
run 00-recog
```

Где **00-recog** имя выполняемого сценария