ИНСТРУКЦИЯ ПО УстановкИ

Платформы речевая аналитика

ИНТЕЛЛЕКТУАЛЬНЫЕ РЕШЕНИЯ

МОСКВА 2021

Оглавление

[1 Введение 3](#_Toc91807016)

[2 Оборудование и программное обеспечение 3](#_Toc91807017)

[2.1 Перечень программного обеспечения целевой платформы 3](#_Toc91807018)

[2.2 ПОШАГОВАЯ ИНСТРУКЦИЯ ПО УСТАНОВКЕ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ ПЛАТФОРМЫ РЕЧЕВОЙ АНАЛИТИКИ 4](#_Toc91807019)

[3 описание функциональных характеристик экземпляра программного обеспечения 5](#_Toc91807020)

[3.1 Основная конфигурация ПРОГРАММНОГО обеспечения 5](#_Toc91807021)

[3.2 Программное обеспечение WEB 5](#_Toc91807022)

[3.4 Программное обеспечение App 5](#_Toc91807024)

[3.6 Программное обеспечение SA 5](#_Toc91807026)

[3.8 Программное обеспечение ASR PROXY 5](#_Toc91807028)

[3.10 Мониторинг 5](#_Toc91807030)

[3.12 Хранение образов контейнеров 5](#_Toc91807032)

[4 описание процесса сопровождения программного обеспечения платформы речекой аналитики 5](#_Toc91807033)

[4.1 данные о персонале, задействованном в процессе разработки 5](#_Toc91807034)

[4.2 фактический почтовый адрес, по которому осуществляется процесс разработки ПО 5](#_Toc91807035)

[5 описание процесса сопровождения программного обеспечения платформы речекой аналитики 5](#_Toc91807036)

[5.1 данные о возможных средствах коммуникации со службой поддержки 5](#_Toc91807037)

[5.2 данные о режиме работы службы поддержки 6](#_Toc91807038)

[5.3 фактический почтовый адрес, по которому осуществляется процесс сопровождения 6](#_Toc91807039)

[6 описание процесса сопровождения программного обеспечения платформы речекой аналитики 6](#_Toc91807041)

[6.1 Хранение образов контейнеров 6](#_Toc91807042)

# Введение

Данное руководство описывает основные сценарии по установке **Платформы Речевая аналитика** (далее веб-интерфейс или ПРА).

# Оборудование и программное обеспечение

ПРА использует программное обеспечение Docker для установки и управления компонентами Системы. В качестве основного управляющего оркестратора используется программное обеспечение Docker Swarm в режиме кластера.

Тип и версия операционной системы: Linux Ubuntu server 18.04 LTS или выше

Для обеспечения минимальной отказоустойчивости требуется:

Сервер WEB приложений в количестве 2 шт.

Сервер App приложений и Речевого анализа в количестве 2 шт.

Сервер управления в количестве 1 шт.

## Перечень программного обеспечения целевой платформы

***Prerequisite:***

Docker v.19.3x и выше ( ПО должно быть установлено на всех серверах).

***Устанавливаемое ПО:***

Программное обеспечение ПРА поставляется на инфраструктурную среду в виде архива с конфигурациями и образами контейнеров в следующем составе набора файлов:

Состав набора файлов программного обеспечения ПРА, необходимых для ручной установки

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| № | Описание Сервиса/модуля | Название файла | Сервер | Название сервера распределения |  |  |
| 1 | WEB сервис | va\_web | VM | va\_web |  |  |
| 2 | Сервис файловой загрузки данных | Va\_sftp | VM | Va\_sftp |  |  |
| 3 | Сервер приложений | va\_bo  (va\_bo\_app, va\_bo\_web) | VM | va\_bo |  |  |
| 4 | application для frontoffice | va\_fo | VM | va\_fo |  |  |
| 5 | Сервер СУБД | va\_db | VM | va\_db |  |  |
| 6 | Сервер управления, шина данных, мониторинг | va\_util | VM | va\_util |  |  |
| 7 | Сервер синтеза речи | va\_tts | VM | va\_tts |  |  |
| 8 | Модуль речевого анализа | va\_sa | VM | va\_sa |  |  |
| 9 | Сервер распознавания речи | va\_asr | VM | va\_asr |  |  |
| 10 | Балансировщик нагрузки | nginx\_proxy (for va\_asr) | VM | va\_asr |  |  |
| 11 | Сервер хранения контента | va\_stor (minio:latest) | VM | va\_stor |  |  |
| 12 | Сервер управления вызовами и сценариями виртуального оператора | va\_call | VM | va\_call | Развитие |  |
| 13 | Сервер Freeswitch | va\_fs | 12 | va\_call | Развитие |  |
| 14 | Модули виртуального оператора | va\_callcontrol | 12 | va\_call | Развитие |  |
| 15 | Модули виртуального оператора | va\_callmedia | 12 | va\_call | Развитие |  |
| 16 | Модули виртуального оператора | va\_callapi | 3,4 | va\_bo, va\_fo | Развитие |  |
| 17 | Модули виртуального оператора | redis | 3,4 | va\_bo, va\_fo | Развитие |  |

СТАРТОВАЯ СИСТЕМА (Пусковой этап) (требуется уточнение , не в сокращенном ли варианте реализации первая инсталляция производится. Если в сокращенном, то взять предыдущую таблицу и выбросить из нее лишнее, привести ниже)

(минимально необходимая и достаточная для реализации базового функционала)

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| № |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |

## ПОШАГОВАЯ ИНСТРУКЦИЯ ПО УСТАНОВКЕ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ ПЛАТФОРМЫ РЕЧЕВОЙ АНАЛИТИКИ

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **№** | **Название шага** | **Комментарии** | **Зона отв.** | **Длит-сть, ч** |
| 1 | Выделение инфраструктуры | Виртуальные или аппаратные сервера | Инженерная служба | экспертно |
| 2 | Организация доступов исполнителю, передача логинов и паролей |  | Инженерная служба | экспертно |
| 3 | Подготовка инфраструктуры | Установка Docker на сервера | Инженерная служба | экспертно |
| 4 | Копирование образов и файлов по ссылкам на выделенный Storage в контуре установки | Временное хранение набора необходимых файлов для ручной установки | Исполнитель | экспертно |
| 5 | Загрузка docker образов и файлов конфигураций компонентов программного обеспечения ПРА | На подготовленной инфраструктуре | Исполнитель | экспертно |
| 6 | Создание кластера swarm для каждой сетевой системы (DMZ, Swarm Network, DB Net, CallVoice, Мониторинг, Front/Back office) | Разворачивание проводится сразу в кластеризированный набор серверов с использованием настроки и управления сервисом средствами sworm | Исполнитель | экспертно |
| 7 | Добавление нод в кластер |  | Исполнитель | экспертно |
| 8 | Аудит корректности файлов конфигураций после загрузки, внесение изменений, уточнения  Проверка/корректировка монтирования файлов в контейнеры в конфигурации в соответсвии с правами и путями файлов в системе | volumes:  - /нужный для проверки путь до скопированных файлов/nats/nats-server.conf:/nats-server.conf | Исполнитель | экспертно |
| 9 | Развертывание компонентов приложений ПРА в качестве сервисов swarm | Используея docker-compose.yaml  docker stack deploy --compose-file docker-compose.yaml app | Исполнитель | экспертно |
| 10 | Проверка корректности развертывания сервисов ПРА  Проверка работы сервисов | Выявление недостатков в работе | Исполнитель | экспертно |
| 11 | Внесение изменений в файлы конфигураций и сервисы, отладка | Устранение недостатков в работе | Исполнитель | экспертно |
| 12 | Подготовка образов контейнеров для серверов в DMZ | Для va\_web01/ va\_web02/sftp | Исполнитель | экспертно |
| 13 | Организация связности межжду внутренним сегментом и DMZ | по факту проверок, с учетом реальных фактических адресов и имен | Инж.Сл. Заказчика | экспертно |
| 14 | Загрузка образов контейнеров на сервера в DMZ | Для va\_web01 / va\_web02 | Исполнитель | экспертно |

Действия 5-11 выполняются для каждого сервера.

# описание функциональных характеристик экземпляра программного обеспечения

## Основная конфигурация ПРОГРАММНОГО обеспечения

Основная конфигурационная структура описана в формате docker compose YAML, версии 3.4.

## Программное обеспечение WEB

Доступ к контейнерам происходит путем проксирования запросов от WEB сервера NGINX к контейнерам с WEB приложениями.

## Конфигурационный файл

*# Zone check for SSL: prod with SSL, test without SSL, rtk – iframe rtk*zone: rtk – режим работы шифрования (для продуктивной среды РТК с iframe - rtk)port: 8080 – порт запуска web приложений  
  
*#Manual Files upload "Отображение раздела Загрузить файлы", 0 – не отображать, 1 - отображать*manual\_upload: 0

*# Настройки доступа к хранению сессий пользователей*  
redis\_ip: redisredis\_port: 6379  
redis\_login: login  
redis\_password: password  
redis\_db: 0

*# Настройки доступа к APP серверам приложений*api\_ip: nginx  
api\_port: 8675

*# Настройки доступа к контейнеру хранения аудио контента*minio\_host: minio:9000  
minio\_access\_key: hd78347gdncw09823hd2x5134  
minio\_secret\_key: xm34gaxkfgaw8m232x3rmh8f9f  
minio\_files\_bucket: speech-files  
minio\_temp\_files\_bucket: temp-files  
minio\_web\_files: web  
minio\_secure: 0  
  
default\_mode: 1 - режим работы языковой модели (всегда 1)  
  
external\_content: 1 – использование СХДП для получения аудио контента  
external\_content\_ip: IP адрес WEB сервера c которого отправляется запрос  
external\_content\_nginx: IP адрес WEB сервера c которого отправляется запрос

external\_content\_vats\_url: <http://188.254.33.70:8080/svetets/orchestrator/speech/> : путь к получению контента с СХДП  
  
converted\_content: mp3 – поиск альтернативного формата контента (mp3)  
  
vocab\_description\_length: 120 – одна строка 120 символов (Длина фильтрации по количеству слов в словарях)default\_xlsx\_sheet: Sheet1 – (Вкладка по умолчанию в excel файле со скриптом )  
  
scripts\_percent: 70 – Значение по умолчанию для соответствия скрипту

## Программное обеспечение App

Программное обеспечение APP является ключевым сервером приложений обеспечивающих обработки логики и взаимодействие между компонентами ПРА.

## конфигурационный файл

*# Main app port, Основной порт работы приложения*port: 8675  
  
*# Cluster Params настройки работы режима кластера, текущий режим получения контента не позволяется использовать кластерный режим, значение 0 по-умолчанию.*cluster\_mode: 0  
cluster\_nodes:  
 - app1:8675- app2:8675 *# MongoDB настройки подключения*mongo\_ip: mongo  
mongo\_port: 27017  
mongo\_db: rtk\_analytics  
mongo\_collection\_data: data  
mongo\_collection\_jobs: jobs  
mongo\_collection\_queue: job\_queue  
mongo\_collection\_load\_queue: load\_queue  
  
*# Postgres DB настройки подключения*postgres\_ip: postgres  
postgres\_port: 5432  
postgres\_db: rtk\_ra  
postgres\_name: postgres  
postgres\_password: postgres  
  
*# Redis DB настройки подключения*redis\_ip: redis  
redis\_port: 6379  
redis\_db: 1  
redis\_name:  
redis\_password:  
  
*# Create empty Postgres DB – сервисный блок настроек, использовать только на тестовой среде. Используется для переиниациализации структуры данных БД.*create\_db: 0  
*# Initialize Postgres DB Scheme with drop all tables*init\_db: 0  
*# Add default content to Postgres DB*default\_content: 0

*# PG DB Limits*pg\_max\_connections: 1000  
pg\_max\_idle\_connections: 10  
*# PG Connection Lifetime in seconds*pg\_connection\_lifetime: 10  
  
*# Logfile path*logfile\_path: log/api\_log.log  
  
*# upload\_folder\_path*upload\_folder\_path: upload  
  
*# S3 storage path*minio\_host: minio:9000  
minio\_access\_key: hd78347gdncw09823hd2x5134  
minio\_secret\_key: xm34gaxkfgaw8m232x3rmh8f9f  
minio\_files\_bucket: speech-files  
minio\_temp\_files\_bucket: temp-files  
minio\_web\_files: web  
  
*# RTK VATS Svetets API*vats\_api\_url: <http://10.243.192.29:8500/svetets/orchestrator/speech> - URL запроса к основному API Svetetsvats\_api\_auth\_url: <http://10.243.192.29:8500/svetets/orchestrator/speech> - URL запроса к API Svetets для авторизации пользователей  
  
*# Worker params*timeout: 1 *# 1 время ожидания между запросами для обработки очереди, секунд*wait\_progress\_timeout: 60 *# 3 время ожидания обработки задания в состоянии in\_progress, минут*

wait\_queue\_processing\_timeout: 5 # *– время ожидания для обработки ожидающих в очереди заданий*wait\_queue\_timeout: 3 *# 3 – время ожидания для резервирования средств, минут*wait\_charging\_timeout: 60 *# 60 - время ожидания для списания средств, минут*wait\_attempts: 3 *# количество попыток по каждому действию*queue\_batch\_size: 10 *# количество одновременно обрабатываемых запросов*upload\_wait\_timeout: 720 *# 720 – время ожидания на повтор загрузки метаданных, минут*upload\_wait\_attempts: 30 *# количество попыток ожидания загрузки метаданных*local\_ip: 10.36.153.148 – IP адреса веб сервера для установки промежуточного Header X-Real-IP для получения аудио контента с СХДП.

nats\_host: nats – путь к nats серверу очередей   
nats\_port: 4222 – порт к nats серверу очередей  
nats\_version: 1 - # выбор версии 0 – версия без async очереди nats, 1 – версия с async очередью nats  
nats\_queue: sa – наименование канала в nats  
max\_parallel\_sa\_streams: 20 – максимальное количество параллельных заданий

## Программное обеспечение SA

Программное обеспечение SA является ключевым сервером анализа аудил и текстовых данных.

## Конфигурационный файл

app\_ip: # App IP заполняется только если нет доступа по hostname контейнера  
app\_port: 8980

app\_mode: 0 # 0 - режим отладки выключен, 1 - режим отладки включен  
node\_name: # Уникальное наименование ноды SA, если не заполнено используется hostname контейнера

*# Process limit* – количество одновременных потоков анализаprocess\_limit: 4

# Multiprocessing 0 - disable, 1 – enable, обработка каналов аудио записи параллельно  
mpc: 0  
   
*# Log file – путь к лог файлу*log\_file: log/speech\_api.log  
  
*#storage path внутренний путь хранения временных файлов*storage: storage  
minio\_host: minio:9000  
minio\_access\_key: hd78347gdncw09823hd2x5134  
minio\_secret\_key: xm34gaxkfgaw8m232x3rmh8f9f  
minio\_files\_bucket: speech-files  
minio\_temp\_files\_bucket: temp-files  
minio\_web\_files: web  
  
*# temp folders*temp\_folder: temp  
  
*# wave audio form path*audioform: /usr/bin/audiowaveform – генератор аудио волны для web приложений  
  
*# emotion\_detection model – модель определения эмоций*em\_model: classifier4.gmmvocab\_synon: synonyms\_only\_adjective.txt – указание на файл разметки словаря синонимов

# Set eng version 1 - enable, 0 – disable

eng\_asr: 0 – указание на включение англоязычную версию (1 – вкл, 0 - выкл), требуется наличие модели.

# Save export file format. wav/mp3

file\_format: mp3 – формат сохранения файлов

# engine version sync/async

engine: async – тип режима работы (требуется аналогичный с APP, async – версия с nats, sync – без nats)

# NATS Config

nats\_ip: nats

nats\_port: 4222 – порт к nats серверу очередей

nats\_queue: sa – наименование канала в nats

nats\_subqueue: queue1 – наименование очереди в канале в nats

# Redis Config

redis\_ip: redis – путь к redis серверу

redis\_port: 6379 – порт к redis серверу

redis\_login: – логин к redis серверу

redis\_password: – пароль к redis серверу

redis\_db: 5 – БД redis сервера

redis\_exp: 300 – время хранения информации о записи

# ASR Engine

asr\_engine: external – тип ASR (external – внешний, internal - внутренний)

asr\_ip: asr\_proxy – имя хоста с ASR

asr\_port: 2600 – порт хоста с ASR

asr\_ip\_eng: asrproxy – имя хоста с английской версией ASR

asr\_port\_eng: 2601– порт хоста с английской версией ASR

## Программное обеспечение ASR PROXY

Программное обеспечение ASR PROXY является сервером балансировки запросов перевода речи в текст.

## Конфигурационный файл

#user nobody;

worker\_processes 2;

events {

worker\_connections 1024;

}

http {

include mime.types;

default\_type application/octet-stream;

log\_format main '$remote\_addr - $remote\_user [$time\_local] "$request" '

'$status $body\_bytes\_sent "$http\_referer" '

'"$http\_user\_agent" "$http\_x\_forwarded\_for"';

sendfile on;

keepalive\_timeout 65;

server {

listen 7810;

server\_name localhost;

location / {

root html;

index index.html index.htm;

}

error\_page 500 502 503 504 /50x.html;

location = /50x.html {

root html;

}

}

map $http\_upgrade $connection\_upgrade {

default upgrade;

'' close;

}

upstream asrserver {

server asr1:2700 max\_conns=6; - # указание по лимиту подключений на asr1

server asr2:2700 max\_conns=6; - # указание по лимиту подключений на asr2

server asr3:2700 max\_conns=15; - # указание по лимиту подключений на asr3

server asr4:2700 max\_conns=15; - # указание по лимиту подключений на asr4

}

server {

listen 2600;

server\_name 0.0.0.0;

location / {

proxy\_pass http://asrserver;

proxy\_http\_version 1.1;

proxy\_set\_header Upgrade $http\_upgrade;

proxy\_set\_header Connection $connection\_upgrade;

proxy\_read\_timeout 950s;

}

}

}

## Мониторинг

Каждый компонент ПРА имеет встроенные функции мониторинга доступности.

Управление состояние производится встроенными средствами Docker swarm.

## Управление нагрузкой

Для управления масштабирование ПРА при превышении нагрузи на какой либо компонент, предусмотрена возможность проверки состояние размера очередей с запросов в APP компонент:

**Очередь на загрузку метаданных и аудиофайлов в систему из ВАТС.**

**Запрос:** GET <http://app/monitoring/load_length> , где app – URL app приложения

**Ответ:** цифровое значение длины очереди.  
**Очередь на обработку загруженных в систему аудиоданных.**

**Запрос:** GET <http://app/monitoring/sa_length> , где app – URL app приложения

**Ответ:** цифровое значение длины очереди

**Очередь на обработку загруженных в систему аудиоданных в минутах.**

**Запрос:** GET <http://app/monitoring/sa_length_minute/> , где app – URL app приложения

**Ответ:** цифровое значение длины очереди в минутах.

**Очередь на обработку загруженных в систему аудиоданных старше 12 часов.**

**Запрос:** GET <http://app/monitoring/sa_length_12/> , где app – URL app приложения

**Ответ:** цифровое значение длины очереди более 12 часов.

**Очередь на обработку загруженных в систему аудиоданных в минутах старше 12 часов.**

**Запрос:** GET <http://app/monitoring/sa_length_minute_12/> , где app – URL app приложения

**Ответ:** цифровое значение длины очереди более 12 часов в минутах.

**Текущая производительность системы в записях.**

**Запрос:** GET <http://app/monitoring/performance/> , где app – URL app приложения

**Ответ:** цифровое значение количества одновременно обрабатываемых записей.

**Текущая производительность системы в минутах.**

**Запрос:** GET <http://app/monitoring/performance_minute/> , где app – URL app приложения

**Ответ:** цифровое значение количества одновременно обрабатываемых записей в минутах.

## Хранение образов контейнеров

Образы контейнеров хранятся в виде .tar архивов.

# описание процесса сопровождения программного обеспечения платформы речекой аналитики

## данные о персонале, задействованном в процессе разработки

Процесс разработки программного обеспечения Платформы Речевой Аналитики многоуровнево распределен от стадии производства через стадию внедрения до стадии промышленной эксплуатации и последующего сопровождения, и поддержки:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **№** | **Наименование** | **Функционал** | **Кол-во специалистов** | **Квалификация** |
| 1 | Производственный блок | Создание и отладка кодов ПРА | 3 | Разработчики фронт, разработчики бэк |
| 2 | Служба внедрения | Установка и запуск программного обеспечения | 2 | Квалифицированный |
| 3 | Служба тестирования | Тестирование программного обеспечения | 3 | Квалифицированный |
| 4 | Служба поддержки | Сопровождение программного обеспечения, контакт центр | 2 | Квалифицированный |
| 5 | Руководитель проекта | Управление проектами | 1 | Квалифицированный |
| 6 | Системный администратор | Поддержание работы инфраструктуры | 1 | Квалифицированный |

## фактический почтовый адрес, по которому осуществляется процесс разработки ПО

Процесс разработки ПО платформы речевой аналитики организован по фактическому юридическому адресу компании-разработчика.

# описание процесса сопровождения программного обеспечения платформы речекой аналитики

## данные о возможных средствах коммуникации со службой поддержки

Средствами коммуникаций со службой поддержки программного обеспечения предусмотрена возможность взаимодействия посредством e-mail: support@intelligent-decisions.net , amerkulov@intelligent-decisions.net, а также посредством телефонной связи по номеру: +7 926 420 0117.

## данные о режиме работы службы поддержки

Служба поддержки программного обеспечения функционирует в режиме 5х8:

в будние дни с ПН по ПТ с 10:00 до 18:00 часов по Московскому времени.

## фактический почтовый адрес, по которому осуществляется процесс сопровождения

121353, г Москва, шоссе Можайское, дом 45 КОРПУС 1, КВАРТИРА 105

## возможные неисправности и какими силами они будут устраняться

После успешного и корректного старта платформы на продуктивной среде целевой инфраструктуры возможные неисправности могут собой представлять ситуации, связанные с отказом аппаратной части серверной инфраструктуры или отказы сетевого уровня, устранением проблем на которых находится в зоне ответственности системного администратора. В его обязанности, также, входим формирование актуальных резервных копий предустановленного ПО и конфигураций для выполнения оперативного восстановления работоспособности платформы и доступа к ее интерфейсу.

## характеристики персонала поддержки

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **№** | **Наименование** | **Функционал** | **Кол-во специалистов** | **Квалификация** |
| 1 | Служба поддержки | Сопровождение программного обеспечения, контакт центр | 2 | Квалифицированный |
| 2 | Системный администратор | Поддержание работы инфраструктуры | 1 | Квалифицированный |
| 3 | Производственный блок | Создание и отладка кодов ПРА | 2 | Разработчики фронт, разработчики бэк |

По-умолчанию путь к образам /opt/app/Images/

## Наименование образов

WEB: rtk\_web\_42.tar

APP: rtk\_api\_43.tar

SA: rtk\_sa\_44.tar

ASR: rtk\_asr\_40.tar

NGINX: nginx.tar

NATS: mats.tar

MONGO: mongo.tar

POSTGRES: postgres.tar

REDIS: redis.tar

MINIO: minio\_rtk.tar

# описание процесса Разработки

Интеграция ПРА с интеграционным API Svetets реализована с использованием прокси сервера nginx установленных и расположенных на каждом Application сервере для обработки входящих и исходящих запросов.

## Силы и средства

Процесс разработки программного обеспечения Платформы Речевой Аналитики многоуровнево распределен от стадии производства через стадию внедрения до стадии промышленной эксплуатации и последующего сопровождения, и поддержки:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **№** | **Наименование** | **Функционал** | **Кол-во специалистов** | **Квалификация** |
| 1 | Производственный блок | Создание и отладка кодов ПРА | 3 | Разработчики фронт, разработчики бэк |
| 2 | Служба внедрения | Установка и запуск программного обеспечения | 2 | Квалифицированный |
| 3 | Служба тестирования | Тестирование программного обеспечения | 3 | Квалифицированный |
| 4 | Служба поддержки | Сопровождение программного обеспечения | 2 | Квалифицированный |
| 5 | Руководитель проекта | Управление проектами | 1 | Квалифицированный |

Разработка программного обеспечения ведется на технических средствах типа IBM совместимых ПК.

## РЕсурсы

Процесс разработки программного обеспечения Платформы Речевой Аналитики выстроен на базе серверной архитектуры. Структура архитектурных решений программного обеспечения позволяет разворачивать платформу и вести ее разработку как на одном локальном сервере, так и на множестве распределенных в сети нод. Это позволяет для разработки и отладки программных кодов использовать один локальный сервер, расположенный в офисе компании.

## Дорожная карта развития

В 2022 году планируется расширение функционала: Виртуальный оператор Контакт Центра с возможностями применения автоматизированных исходящих обзвонов и приема вызовов.

## Конфигурация Nginx APP

Путь к файлу /etc/nginx/conf.d/app.conf

upstream api\_upstream {

server localhost:8675 max\_fails=3 fail\_timeout=5s weight=3;

server 10.243.56.231:8676 max\_fails=3 fail\_timeout=5s weight=3;

}

upstream api\_vats {

server 10.243.56.100:8080 max\_fails=3 fail\_timeout=5s weight=3;

server 10.243.56.101:8080 max\_fails=3 fail\_timeout=5s weight=3;

}

# проксирование внешних запросов от API VATS

server {

listen 80 default\_server;

server\_name \_;

location / {

proxy\_pass http://api\_upstream;

}

location /api1/v1 {

proxy\_pass http://api\_upstream;

}

location /minio {

proxy\_pass http://localhost:9000;

}

location /web {

proxy\_pass http://localhost:9000;

}

location /speech-files {

proxy\_pass http://localhost:9000;

}

}

# балансировка исходящих запросов к API Svetets VATS

server {

listen 8500;

server\_name \_;

access\_log /var/log/nginx/vats\_api\_access.log main;

charset utf-8;

location / {

proxy\_pass <http://188.254.33.70:8080>; # URL сервера API Svetets VATS

# proxy\_pass <http://10.243.56.100:8080>; # URL сервера API Svetets VATS

proxy\_set\_header X-Real-IP 10.243.56.230; # IP адрес Application сервера

}

location /svetets/orchestrator/speech {

proxy\_pass <http://188.254.33.70:8080>;

# proxy\_pass <http://api_vats>; # URL сервера API Svetets VATS с балансировкой

# proxy\_pass http://10.243.56.100:8080;

proxy\_set\_header X-Real-IP 10.243.56.230;

}

}

## Конфигурация Nginx WEB

Путь к файлу /etc/nginx/conf.d/web.conf

upstream web\_upstream {

server localhost:8000 max\_fails=3 fail\_timeout=3s weight=3; # внутренний балансировщик

server 10.243.56.228:8080 max\_fails=3 fail\_timeout=3s weight=5; # web контейнер №1

server 10.243.56.229:8081 max\_fails=3 fail\_timeout=3s weight=5; # web контейнер №2

}

server {

listen 80 default\_server;

listen [::]:80 default\_server;

server\_name \_;

location / {

expires 0;

add\_header Cache-Control "no-store";

proxy\_pass http://web\_upstream;

proxy\_set\_header Host $host;

proxy\_set\_header X-Forwarded-For $proxy\_add\_x\_forwarded\_for;

client\_max\_body\_size 102m;

client\_body\_buffer\_size 128k;

# client\_body\_temp\_path /var/nginx/client\_body\_temp;

}

location /api/v1/ {

proxy\_pass http://10.243.56.230/;

}

location /web/ {

proxy\_pass http://10.243.56.230/;

}

location /speech-files/ {

proxy\_pass http://10.243.56.230/;

}

}

server {

listen 8600;

server\_name \_;

location / {

proxy\_pass http://localhost:8000/;

}

}

server {

listen 9005;

server\_name \_;

ignore\_invalid\_headers off;

client\_max\_body\_size 0;

proxy\_buffering off;

location / {

proxy\_set\_header X-Real-IP $remote\_addr;

proxy\_set\_header X-Forwarded-For $proxy\_add\_x\_forwarded\_for;

proxy\_set\_header X-Forwarded-Proto $scheme;

proxy\_set\_header Host $http\_host;

proxy\_connect\_timeout 300;

# Default is HTTP/1, keepalive is only enabled in HTTP/1.1

proxy\_http\_version 1.1;

proxy\_set\_header Connection "";

chunked\_transfer\_encoding off;

proxy\_pass http://10.243.56.230:9005;

}

}