

РУКОВОДСТВО АДМИНИСТРАТОРА
Программное обеспечение Система транскрибации
разговоров и подсчета характеристик

Листов 6

г. Москва, 2024г.

Оглавление

1. ТРЕБОВАНИЯ ДЛЯ УСТАНОВКИ И НАСТРОЙКИ ПО.....	4
2. ИНСТРУКЦИЯ ПО НАСТРОЙКЕ ПО	5

Общая информация о документе

Настоящий документ представляет собой руководство по установке и настройке информационной системы «Система транскрибации разговоров и подсчета характеристик».

В данном документе представлены сведения о необходимых объемах носителя поставки, системные требования для установки и настройки ИС, а также описан пошаговый процесс установки и настройки экземпляра.

1. ТРЕБОВАНИЯ ДЛЯ УСТАНОВКИ И НАСТРОЙКИ ПО

Системные требования, описанные в данном документе, предполагают, что окружение на сервере экземпляра уже настроено.

Необходимые системные требования:

1. Наличие GPU (модель объем памяти) с актуальными драйверами на целевой машине (v23.05)
2. Установлена CUDA Toolkit (v13)
3. Подключение к интернет для импорта вспомогательных образов из `dockerhub` или наличие в локарном хранилище `keycloak` версии `13.0.1`.
4. Установленный `docker` и `docker-compose`

2. ИНСТРУКЦИЯ ПО НАСТРОЙКЕ ПО

Экспорт версий

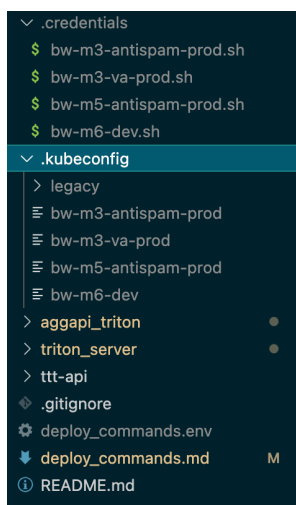
```
export AGGAPI_BRANCH_VER="develop:0.12.19"  
export TRITON_SERVER_BRANCH_VER="main:0.1.02"  
export TTT_SERVER_BRANCH_VER="main:0.0.1"
```

Загрузка образа из Nexus лаборатории и заливка в Harbor Beeworks

```
docker pull nexus-dcr.avlab.dev/ap-aggapapi/$AGGAPI_BRANCH_VER  
docker tag nexus-dcr.avlab.dev/ap-aggapapi/$AGGAPI_BRANCH_VER harbor.vimpelcom.ru/audio-analytics-big-data1-2/ap-aggapapi/$AGGAPI_BRANCH_VER  
docker push harbor.vimpelcom.ru/audio-analytics-big-data1-2/ap-aggapapi/$AGGAPI_BRANCH_VER  
  
docker pull nexus-dcr.avlab.dev/audio-triton-server/$TRITON_SERVER_BRANCH_VER  
docker tag nexus-dcr.avlab.dev/audio-triton-server/$TRITON_SERVER_BRANCH_VER harbor.vimpelcom.ru/audio-analytics-big-data1-2/audio-triton-server/$TRITON_SERVER_BRANCH_VER  
docker push harbor.vimpelcom.ru/audio-analytics-big-data1-2/audio-triton-server/$TRITON_SERVER_BRANCH_VER  
  
docker pull nexus-dcr.avlab.dev/ttt-api/$TTT_SERVER_BRANCH_VER  
docker tag nexus-dcr.avlab.dev/ttt-api/main:0.0.1 harbor.vimpelcom.ru/audio-analytics-big-data1-2/ttt-api/$TTT_SERVER_BRANCH_VER  
docker push harbor.vimpelcom.ru/audio-analytics-big-data1-2/ttt-api/$TTT_SERVER_BRANCH_VER
```

Инсталляция решения с использование helm чартов проекта deploy-solution

Создаем папки `.credentials/` и `.kubeconfig/`. Складываем туда креды и конфиги.



Запускаем скрипт для инициализации переменных окружения

```
sh ./credentials/bw-m6-dev.sh
```

Делаем релиз нужных микросервисов

```
helm upgrade --install -n $DEPLOY_NAMESPACE ap-aggapl-triton aggapl_triton/helm/ --kubeconfig $DEPLOY_KUBE_CONFIG
helm upgrade --install -n $DEPLOY_NAMESPACE audio-triton-server triton_server/helm/ --kubeconfig $DEPLOY_KUBE_CONFIG
helm upgrade --install -n $DEPLOY_NAMESPACE ap-ttt-api ttt-api/helm/ --kubeconfig $DEPLOY_KUBE_CONFIG
```

Инициализация секретов

Hurbor auth for docker images

```
echo "Hurbor auth for docker images"
kubectl --kubeconfig $DEPLOY_KUBE_CONFIG -n $DEPLOY_NAMESPACE create secret docker-registry harbor-authenticate --docker-username=$DOCKER_USERNAME --docker-password=$DOCKER_PASSWORD --docker-server=$DOCKER_SERVER
kubectl --kubeconfig $DEPLOY_KUBE_CONFIG patch serviceaccount default -p '{"imagePullSecrets": [{"name": "harbor-authenticate"}]}'

# Windows version
# kubectl --kubeconfig $DEPLOY_KUBE_CONFIG patch serviceaccount default -p '{"imagePullSecrets": [{"name": "harbor-authenticate"}]}'
```

Aggapl secrets

```
echo "Aggapl secrets"
kubectl --kubeconfig $DEPLOY_KUBE_CONFIG -n $DEPLOY_NAMESPACE create secret generic ap-aggapl-auth-login --from-literal=login=$AGGAPI_AUTH_LOGIN --from-literal=password=$AGGAPI_AUTH_PASSWORD
kubectl --kubeconfig $DEPLOY_KUBE_CONFIG -n $DEPLOY_NAMESPACE create secret generic ap-aggapl-auth-keycloak-login --from-literal=client-secret=$AGGAPI_KEYCLOAK_AUTH_PASSWORD --from-literal=client-id=$AGGAPI_KEYCLOAK_AUTH_LOGIN
kubectl --kubeconfig $DEPLOY_KUBE_CONFIG -n $DEPLOY_NAMESPACE create secret generic ap-aggapl-minio --from-literal=access-key=$S3_ACCESS_KEY --from-literal=secret-key=$S3_SECRET_KEY
kubectl --kubeconfig $DEPLOY_KUBE_CONFIG -n $DEPLOY_NAMESPACE create secret generic ap-aggapl-redis --from-literal=password=$AGGAPI_REDIS_PASSWORD --from-literal=username=$AGGAPI_REDIS_LOGIN
kubectl --kubeconfig $DEPLOY_KUBE_CONFIG -n $DEPLOY_NAMESPACE create secret generic ap-aggapl-rabbitmq --from-literal=login=$AGGAPI_RABBIT_LOGIN --from-literal=password=$AGGAPI_RABBIT_PASSWORD
```