|  |  |
| --- | --- |
|  |  |
| **Утверждаю**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ФИО«\_\_\_\_»\_\_\_\_\_\_\_\_\_\_\_\_2020  | **Утверждаю**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ФИО«\_\_\_\_»\_\_\_\_\_\_\_\_\_\_\_\_2020 |

**ИНФОРМАЦИОННАЯ СИСТЕМА ФОРМИРОВАНИЯ АВТОМАТИЧЕСКИХ ДИАЛОГОВ FORM.ONE**

**Руководство по настройке и установке**

Действует с «\_\_» \_\_\_\_\_\_\_ 2020 г.

на 15 листах

г. Москва, 2020

Содержание

[Содержание 2](#_Toc38110086)

[1. Сокращения, термины и определения 3](#_Toc38110087)

[2. Общие сведения 4](#_Toc38110088)

[3. Требования к оборудованию 5](#_Toc38110089)

[4. Архитектурное решение 6](#_Toc38110090)

[5. Инструкция по восстановлению из резервной копии 7](#_Toc38110091)

[6. Инструкция по обновлению 8](#_Toc38110092)

[7. Инструкция по развертыванию 9](#_Toc38110093)

[Список рисунков 14](#_Toc38110094)

[Список таблиц 15](#_Toc38110095)

1. Сокращения, термины и определения

Таблица 1. Сокращения, термины и определения

| Термин/Сокращение | Определение |
| --- | --- |
| API | (От англ. - Application Programming Interface) описание способов (набор классов, процедур, функций, структур или констант), которыми одна компьютерная программа может взаимодействовать с другой программой. |
| CPU | (От англ. Сentral Processing Unit) центральный процессор. |
| HDD | (От англ. Hard (Magnetic) Disk Drive) жесткий диск, устройство хранения информации произвольного доступа, основанное на принципе магнитной записи. |
| JavaScript | Язык программирования, поддерживает объектно-ориентированный, императивный и функциональный стили. |
| PHP | (От англ. - Personal Home Page Tools) скриптовый язык общего назначения, интенсивно применяемый для разработки веб-приложений. |
| RAM | (От англ. Random Access Memory) запоминающее устройство с произвольным доступом. |
| Инстанс | Виртуальный экземпляр программы. Позволяет использовать программу без установки на каждый компьютер пользователя. |
| ОС | Операционная система. |
| Система | Система формирования автоматических диалогов Form.one, предназначенная для создания и развертывания ботов ведения диалогов с респондентом. |
| СУБД | Система управления базами данных. |

1. Общие сведения

В Руководстве представлено пошаговая инструкция по развертыванию системы формирования автоматических диалогов Form.one (далее, Система).

Назначением Системы является автоматизация общения с клиентами, что ведет к снижению трудозатрат на поддержку и привлечение клиентов.

Система состоит из следующих функциональных модулей:

1. Конструктор ботов;
2. Конструктор шаблонов ботов;
3. Раздел для хранения диалогов;
4. Модуль управления доступом пользователей;
5. Модуль для настройки параметров Системы;
6. Интеграция с внешними Системами.

Работа пользователей с Системой осуществляется через веб-интерфейс.

Предоставление доступа к Системе обеспечивается через приглашение и установку индивидуального пароля.

В Системе реализовано разграничение прав пользователей путем назначения ролей. Доступны следующие роли:

* Пользователь – лицо, которое использует Систему, обладает стандартным набором прав: создание ботов и шаблонов, доступ к данным диалогов, настройка личного аккаунта Системы. Имеет доступ только к ботам, созданным им самим.
* Администратор – пользователь с расширенным набором прав. Кроме стандартной функциональности, администратор имеет возможность настраивать корпоративный аккаунт, просматривать информацию о тарифе, управлять доступом пользователей. Кроме того, администратор имеет доступ ко всем ботам инстанса.
* Владелец – пользователь инстанса, которому принадлежат права на использование Системы. Владелец обладает правами администратора, но имеет возможность оплачивать доступ к Системе, а также настраивать тариф.
* Респондент – авторизованный собеседник бота.
1. Требования к оборудованию

Система функционирует с использованием следующего технического и программного обеспечения серверов:

* CPU: 1 процессор Intel® Xeon® E5-1650 v3 Hexa-Core Haswell.
* RAM: 128Gb.
* HDD: 2x2Tb, SATA rev.3.0 (6 Gbit/s), Rotational speed 7200 rpm, (Software RAID 1).
* Языки программирования: PHP, JavaScript.
* СУБД: MySQL 5.7.
* Сервер обмена сообщений: RabbitMQ 3.4.3.
* ОС: Ubuntu Linux server 16.04 LTS.

**Описание используемых технологий:**

* Система разработана на PHP и JavaScript.
* Взаимодействие с другими системами осуществляется через API web-сервиса.
* Для хранения данных используется СУБД MySQL.
* Система работает на Ubuntu Linux server 16.04 LTS c использованием контейнеров Docker (docker-ce 17.06.2), и состоит из десяти контейнеров.
1. Архитектурное решение

Схематическое изображение архитектурного решения Системы представлено на рисунке Рисунок 1.



Рисунок 1. Схема архитектурного решения

1. Веб-сервер (nginx) – отдает статический контент и проксирует запросы на бэкенд.
2. Бэкенд (php-fpm) – приложение, обрабатывает данные, работает с кэширующим сервисом и базой данных.
3. Кэширующий сервер (memcached) – хранит сессии и результаты запросов от базы данных.
4. База данных (MySQL) – хранит данные приложения (шаблоны, документы, файлы и т.д.).
5. Redis – In memory нереляционная высокопроизводительная СУБД.
6. RabbitMQ – платформа системы обмена сообщениями между приложениями или скриптами.
7. Echo – websocket сервер для обмена данными в реальном времени.
8. Cron – программа периодического выполнения заданий.
9. Cli-bot – consumer, скрипт обработки сообщений из RabbitMQ (здесь заложена основная логика работы бота).
10. MongoDB – документоориентированная СУБД. Используется для хранения файлов (GridFS).
11. Инструкция по восстановлению из резервной копии

Можно создавать и восстанавливать Систему из резервной копии с использованием стандартных средств MySQL и MongoDB.

**MySQL**

Для создания копии нужно выполнить в контейнере MySQL команду:

mysqldump -uroot -p[пароль] --dump-date -x -v --log-error=[путь к логу] --all-databases --result-file=[путь / файл бекапа]

Для восстановления из резервной копии нужно выполнить в контейнере MySQL команду:

mysql -uroot -p[пароль]  -e 'source [путь / файл бекапа]'

**MongoDB**

Для создания копии нужно выполнить в контейнере MongoDB команду:

mongodump -o [путь\_к\_каталогу\_копии] 2>&1 | tee [путь / файл лога]

Для восстановления из резервной копии нужно выполнить в контейнере MongoDB команду:

mongorestore [путь\_к\_каталогу\_копии] 2>&1 | tee [путь / файл лога]

1. Инструкция по обновлению

Для обновления Системы необходимо выполнить следующую последовательность действий:

1. Настоятельно рекомендуется перед процедурой обновления создать резервную копию данных mysql, mongodb стандартными средствами данных продуктов.
2. Изменить в файле ~/form\_one/docker-compose.yml либо в .env
TAG=master\_[версия] на версию, образы которой были получены в обновлении.
3. Обновление выполняется из образов (могут присутствовать только те образы, которые обновлены), которые находятся в файлах, либо в реестре Docker Private Registry.

Если образы содержатся в файлах, необходимо загрузить образы:

docker load < [имя\_файла]

Если образы содержатся в реестре Docker Private Registry, требуется:

подключиться к реестру:

docker login -u {USER} -p "{PASSWD}" registry-host:port

загрузить образы:

docker pull [имя\_образа]

1. Из каталога проекта ~/form\_one выполнить развертывание и запуск контейнеров:

docker-compose up -d

Должно произойти обновление контейнеров.

1. Выполнить изменение структуры базы данных:

docker exec -ti form\_one\_web-backend\_1 /bin/bash

php instances migrate
exit

1. В контейнерах cli-bot, web-backend и cron отключить отправку данных мониторингу:

docker exec -ti form\_one\_web-backend\_1 /bin/bash
echo 'pinba.enabled=Off' > /usr/local/etc/php/conf.d/docker-php-ext-pinba.ini
exit

docker exec -ti form\_one\_cli-bot\_1 /bin/bash
echo 'pinba.enabled=Off' > /usr/local/etc/php/conf.d/docker-php-ext-pinba.ini
exit

docker exec -ti form\_one\_cron\_1 /bin/bash
echo 'pinba.enabled=Off' > /usr/local/etc/php/conf.d/docker-php-ext-pinba.ini
exit

1. Инструкция по развертыванию

Для развертывания Системы на сервере необходимо наличие интернет-соединения для установки пакетов docker. Нужен либо локальный репозиторий, либо скопированные на сервер установочные файлы пакетов.

Для запуска контейнеров необходим доступ к командной консоли ОС.

Для размещения файлов Системы должна быть выделена отдельная директория.

При выполнении данных условий запуск Системы занимает примерно три часа.

**Список поставляемых файлов:**

* form\_one/install.txt – инструкция по развертыванию;
* form\_one/form-one\_backend\_master\_[версия] – образ backend приложения form.one;
* form\_one/form-one\_frontend\_master\_[версия] – образ frontend приложения form.one;
* form\_one/docker-compose.yml – файл конфигурации запуска контейнеров Docker;
* form\_one/memcached:1.5.12 – образ Memcached;
* form\_one/mysql\_master\_[версия] – образ Mysql;
* form\_one/echo\_release – образ Echo;
* form\_one/rabbitmq:3.7.8-management – образ RabbitMQ;
* form\_one/redis:5.0.3 – образ Redis;
* form\_one/mongo:4.0.5-xenial – образ MongoDB;
* ssl\_proxy\_ffd/proxy-ssl:latest – образ Nginx (ssl прокси);
* ssl\_proxy\_ffd/vhost.conf – конфигурационный файл для Nginx (ssl прокси);
* ssl\_proxy\_ffd/docker-compose.yml – файл конфигурации запуска контейнеров Docker;
* ssl\_proxy\_ffd/ssl/certs/local\_form\_one.pem – публичный ключ сертификата;
* ssl\_proxy\_ffd/ssl/private/local\_form\_one.pem – секретный ключ сертификата.

Для развертывания Системы необходимо выполнить следующую последовательность действий:

1. Для записи и хранения пользовательских данных нужно создать каталог, к которому будет открыт доступ из контейнеров (root 755):

/data
/data/docker
/data/docker/nginx
/data/docker/nginx/conf
/data/docker/nginx/ssl
/data/docker/nginx/ssl/certs
/data/docker/nginx/ssl/private
/data/docker/form-one-[ ваше наименование инстанса ]

/data/docker/form-one-[ ваше наименование инстанса ]/files

1. Необходимо установить на него права root 777.
2. Для сопоставления UID пользователей, и GID групп между контейнерами и хостом.

useradd docker -u 999 -g 999

usermod -a -G root docker

id docker

uid=999(docker) gid=999(docker) groups=999(docker),0(root)

Пользовательская группа c GID = 999 создается при установке Docker.

В контейнере service-mysql

/data/docker/form-one-[ ваше наименование инстанса ]/mysql:/var/lib/mysql

drwxr-xr-x mysql mysql mysql (755)

uid=999(mysql) gid=999(mysql) groups=999(mysql)

В контейнере web-backend

/data/docker/form-one-[ ваше наименование инстанса ]/files:/var/lib/form-one/files
drwxrwxrwx root root files

В контейнере service-rabbitmq

/data/docker/form-one-[ ваше наименование инстанса ]/rabbitmq:/var/lib/rabbitmq

drwxr-xr-x rabbitmq root rabbitmq

uid=999(rabbitmq) gid=999(rabbitmq) groups=999(rabbitmq)

В контейнере service-mongo

/data/docker/form-one-[ ваше наименование инстанса ]/mongo:/data/db

drwxr-xr-x mongodb root db

uid=999(mongodb) gid=999(mongodb) groups=999(mongodb)

1. Создать каталоги проектов (root 755):

form.one
~/form\_one
ssl-proxy
~/ssl\_proxy\_ffd

1. Скопировать:

vhost.conf > /data/docker/nginx/conf/vhost.conf

ssl/certs/local\_form\_one.pem > /data/docker/nginx/ssl/certs/local\_form\_one.pem

ssl/private/local\_form\_one.pem > /data/docker/nginx/ssl/private/local\_form\_one.pem

form\_one/docker-compose.yml > ~/form\_one/docker-compose.yml

.env > ~/form\_one/.env

ssl\_proxy\_ffd/docker-compose.yml > ~/ssl\_proxy\_ffd/docker-compose.yml

.env > ~/ssl\_proxy\_ffd/.env

1. Изменить в файле ~/form\_one/docker-compose.yml, либо в .env для LOCAL\_INSTANCE [ ваше наименование инстанса ] на необходимое (на латинице в нижнем регистре). Наименование должно соответствовать наименованию вашего инстанса и будет равняться домену последнего уровня в URL, на котором будет опубликована Система.
2. Изменить в файле ~/ssl\_proxy\_ffd/docker-compose.yml доменное имя на необходимое.
3. Для использования кастомного почтового сервера необходимо внести его настройки в файле .env в переменные:

MAIL\_DRIVER=smtp

MAIL\_USERNAME=test@yandex.ru

MAIL\_PASSWORD=123321

MAIL\_HOST=smtp.yandex.ru

MAIL\_PORT=465

MAIL\_ENCRYPTION=ssl

MAIL\_FROM\_NAME=info

MAIL\_FROM\_ADDRESS=test@yandex.ru

|  |
| --- |
|  |
| Внимание! При отсутствии значения нужно ничего не указывать, но оставить переменную (например, MAIL\_ENCRYPTION=). |
|  |

1. Установить Docker CE: https://docs.docker.com/install/
2. Установить Docker Compose: https://docs.docker.com/compose/install/
3. 7. Развертывание выполняется из образов:

proxy-ssl latest;

registry.team-pravo.ru/form-one/backend master;

registry.team-pravo.ru/form-one/frontend master;

mysql master;

memcached 1.5.12;

registry.team-pravo.ru/file-one/echo release;

redis 5.0.3;

rabbitmq 3.7.8-management;

mongo 4.0.5-xenial.

1. Образы находятся в архивах, либо в реестре Docker Private Registry.

Если образы в архиве, нужно загрузить образы:

docker load < [имя\_файла]

Если образы в реестре Docker Private Registry, нужно подключиться к реестру:

docker login -u {USER} -p "{PASSWD}" registry-host:port

загрузить образы:

docker pull [имя\_образа]

1. Из каталога проекта ~/form\_one выполнить развертывание и запуск контейнеров:

docker-compose up -d

Должен произойти успешный запуск контейнеров:

Starting form\_one\_service-mongo\_1 ... done;

Starting form\_one\_web-frontend\_1 ... done;

Starting form\_one\_web-backend\_1 ... done;

Starting form\_one\_service-mysql\_1 ... done;

Starting form\_one\_service-memcached\_1 ... done;

Starting form\_one\_service-rabbitmq\_1 ... done;

Starting form\_one\_service-redis\_1 ... done;

Starting form\_one\_cli-bot\_1 ... done;

Starting form\_one\_cron\_1 ... done.

1. Из каталога проекта ~/ssl\_proxy\_ffd выполнить развертывание и запуск контейнеров:

docker-compose up -d

Должен произойти успешный запуск контейнера:

Starting ssl\_proxy\_ffd\_proxy-ssl\_1 ... done

1. Установить привилегии базы данных (пароль Word1234pass):

docker exec -ti form\_one\_service-mysql\_1 /bin/bash

mysql -uroot –p

grant all privileges on \*.\* to 'form\_one'@'%';

flush privileges;

exit;

exit

1. Выполнить создание структуры базы данных:

docker exec -ti form\_one\_web-backend\_1 /bin/bash

php artisan instance:init-local

php instances migrate

exit

1. В контейнерах cli-bot, web-backend и cron отключить отправку данных по мониторингу:

docker exec -ti form\_one\_web-backend\_1 /bin/bash

echo 'pinba.enabled=Off' > /usr/local/etc/php/conf.d/docker-php-ext-pinba.ini

exit

docker exec -ti form\_one\_cli-bot\_1 /bin/bash

echo 'pinba.enabled=Off' > /usr/local/etc/php/conf.d/docker-php-ext-pinba.ini

exit

docker exec -ti form\_one\_cron\_1 /bin/bash

echo 'pinba.enabled=Off' > /usr/local/etc/php/conf.d/docker-php-ext-pinba.ini

exit

1. Связать «form.one» с адресом хоста в файле hosts:

локальным:

127.0.0.1 api.form.one form.one [ ваше наименование инстанса ].form.one

внешним ip:

ip api.form.one form.one [ ваше наименование инстанса ].form.one

1. Выполнить вход [ ваше наименование инстанса ].form.one:

Login superadmin@superadmin.com;

Password secret.

Список рисунков

[Рисунок 1. Схема архитектурного решения 6](#_Toc38110096)

Список таблиц

[Таблица 1. Сокращения, термины и определения 3](#_Toc38110097)