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

**ИНФОРМАЦИОННАЯ СИСТЕМА ФОРМИРОВАНИЯ ДОКУМЕНТОВ ПО ШАБЛОНУ DOC.ONE**

**Руководство по настройке и установке**

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

на 15 листах

г. Москва, 2020

Содержание

[Содержание 2](#_Toc44075004)

[1. Сокращения, термины и определения 3](#_Toc44075005)

[2. Общие сведения 4](#_Toc44075006)

[3. Требования к оборудованию 6](#_Toc44075007)

[4. Архитектурное решение 7](#_Toc44075008)

[5. Инструкция по восстановлению из резервной копии 8](#_Toc44075009)

[6. Инструкция по обновлению 9](#_Toc44075010)

[7. Инструкция по развертыванию 10](#_Toc44075011)

[Список рисунков 14](#_Toc44075012)

[Список таблиц 15](#_Toc44075013)

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 документов по шаблону Doc.one. |
| СУБД | Система управления базами данных. |

1. Общие сведения

В Руководстве представлено пошаговая инструкция по развертыванию системы формирования документов по шаблону «Doc.one» (далее - Система).

Назначением Системы является автоматическое формирование типовых документов на основе динамических шаблонов.

Целями создания Системы являются:

* Приведение документации к типовому виду;
* Снижение трудозатрат специалистов при формировании документов (договоров, соглашений и пр.);
* Хранение шаблонов и готовых документов.

Система состоит из следующих функциональных модулей:

1. Раздел для хранения документов – «Documents»;
2. Конструктор шаблонов документов – «Templates»;
3. Конструктор типовых блоков – «Library»;
4. Раздел для управления доступом пользователей – «Users»;
5. Раздел для настройки параметров Системы – «Settings»;
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 – формирует документы или пакеты документов на основе динамических шаблонов, экспортирует документы.
* Менеджер – формирует шаблоны документов, создает и ведет библиотеку типовых блоков текста.
* Администратор – производит установку, настройку и мониторинг Системы.

Менеджер, помимо своих функций, может осуществлять функции пользователя в полном объеме. Администратор, помимо своих функций, может осуществлять функции менеджера и пользователя в полном объеме.

В таблице (Таблица 2) представлен доступ пользователей к различной функциональности Системы в зависимости от роли.

Таблица 2. Права доступа пользователей

| Функциональность системы | Роль пользователя | | |
| --- | --- | --- | --- |
| Администратор | Менеджер | Пользователь |
| «Templates» | есть | есть | нет |
| «Library» | есть | есть | нет |
| «Documents» | есть | есть | есть |
| «Users» | есть | нет | нет |
| «Settings» | есть | нет | нет |
| Профиль пользователя | есть | есть | есть |

Роль пользователю назначается администратором Системы при отправке приглашения. Роль пользователя может быть изменена администратором Системы.

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: 2 x 4 ТБ SATA 6 ГБ/с 7200 об/м (программный-RAID 1).
* Языки программирования: PHP, JavaScript.
* СУБД: MySQL 8.0.19.
* ОС: Ubuntu Linux Server 16.04 LTS.

**Описание используемых технологий:**

* Система разработана на PHP и JavaScript.
* Взаимодействие с другими системами осуществляется через API веб-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).

1. Архитектурное решение

Схематическое изображение архитектурного решения Системы представлено на рисунке Рисунок 1.

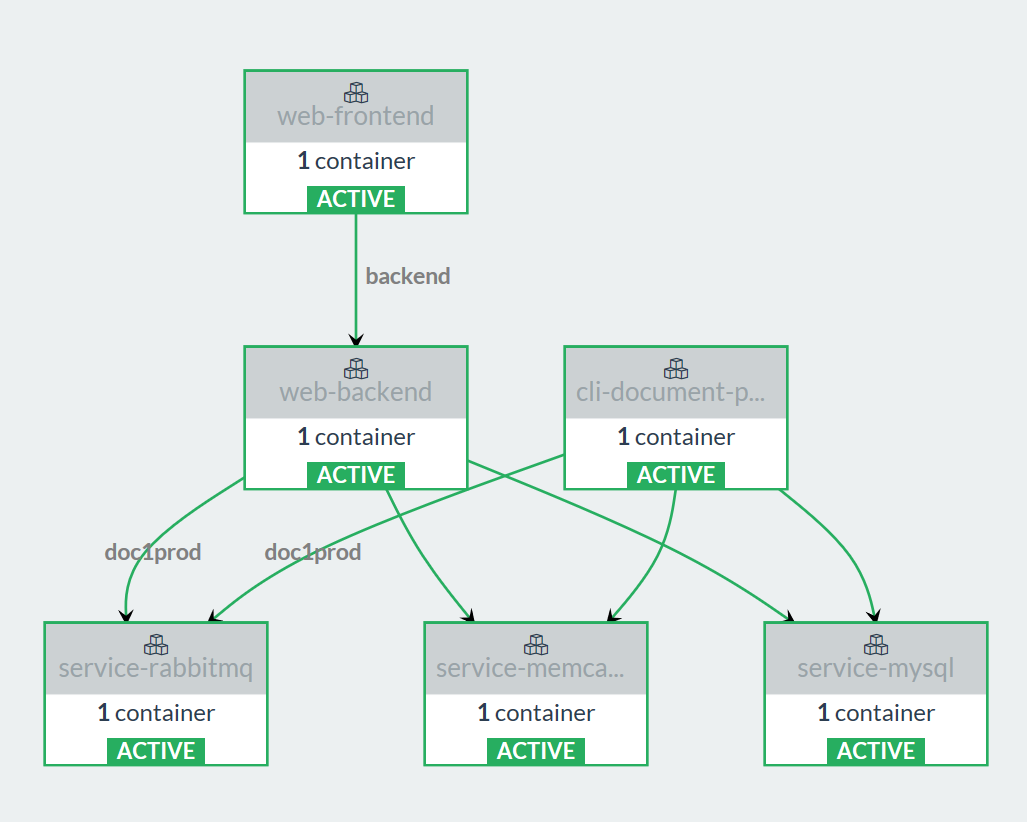


Рисунок 1. Схема архитектурного решения

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. RabbitMQ – менеджер очередей, используется для передачи сообщений в cli-воркеры.
6. Сli-document-package-download – php-воркер, слушает очередь из RabbitMQ и формирует пакеты документов в фоновом режиме.
7. Инструкция по восстановлению из резервной копии

Можно создавать и восстанавливать Систему из резервной копии с использованием стандартных средств 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]'

1. Инструкция по обновлению

Для обновления Системы необходимо выполнить следующую последовательность действий:

1. Настоятельно рекомендуется перед процедурой обновления создать резервную копию данных MySQL стандартными средствами данного продукта.
2. Создать каталог /data/docker/doc-one-[ваше наименование инстанса]/doc-one-general, установить на него права root 777.
3. Изменить в файле ~/doc\_one/docker-compose.yml, либо в .env номер версии, файлы которой получены и на которую выполняется обновление: TAG=master\_[версия].
4. Развертывание выполняется из образов:

registry.team-pravo.ru/doc-one/backend  
registry.team-pravo.ru/doc-one/frontend  
registry.team-pravo.ru/doc-one/docone-mysql  
memcached  
rabbitmq

Образы находятся в файлах, либо в реестре Docker Private Registry.

Если образы находятся в файлах, нужно загрузить образы:

docker load < [имя\_файла]

Если образы находятся в реестре Docker Private Registry, нужно:

подключиться к реестру:

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

загрузить образы:

docker pull [имя]

1. Из каталога проекта ~/doc\_one начать развертывание и запуск контейнеров:

docker-compose up -d

1. Выполнить изменение структуры базы данных и отключение мониторинга:

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

docker exec -ti doc\_one\_cli-document-package-download\_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:**

* doc\_one/install.txt – инструкция по развертыванию;
* doc\_one/doc\_one\_backend\_master\_[версия]  – образ backend приложения doc.one;
* doc\_one/doc\_one\_frontend\_master\_[версия] – образ frontend приложения doc.one;
* doc\_one/memcached:1.5.12 – образ Memcached;
* doc\_one/docone-mysql:master\_[версия] – образ MySQL;
* doc\_one/docker-compose.yml – файл конфигурации запуска контейнеров Docker;
* doc\_one/.env – файл переменных;
* 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/.env – файл переменных;
* ssl\_proxy\_ffd/ssl/certs/local\_doc\_one.pem – публичный ключ сертификата;
* ssl\_proxy\_ffd/ssl/private/local\_doc\_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 ]/doc-one-images

/data/docker/doc-one-[ ваше наименование инстанса ]/doc-one-general

1. После создания каталогов

/data/docker/doc-one-[ ваше наименование инстанса ]/doc-one-images

и

/data/docker/doc-one-[ ваше наименование инстанса ]/doc-one-general

необходимо установить на них права root 777.

1. 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/doc-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/doc-one-[ ваше наименование инстанса ]/doc-one-images:/var/lib/doc-one/images  
drwxrwxrwx root root images (777)  
  
/data/docker/doc-one-[ ваше наименование инстанса ]/doc-one-general:/var/lib/doc-one/general  
drwxrwxrwx root root images (777)

В контейнере service-rabbitmq

/data/docker/doc-one-[ ваше наименование инстанса ]/rabbitmq:/var/lib/rabbitmq  
drwxr-xr-x  5 docker root rabbitmq (755)  
uid=999(rabbitmq) gid=999(rabbitmq) groups=999(rabbitmq)

|  |
| --- |
|  |
| Внимание! Если UID 999 использовать невозможно из-за занятности, можно установить права 777, это разрешит доступ, но все файлы, создаваемые из контейнеров, всё равно будут созданы от UID 999. |
|  |

1. Создать каталоги проектов (root 755):

doc.one – например ~/doc\_one;

ssl-proxy – например ~/ssl\_proxy\_ffd.

1. Скопировать:

vhost.conf > /data/docker/nginx/conf/vhost.conf  
ssl/certs/local\_doc\_one.pem > /data/docker/nginx/ssl/certs/local\_doc\_one.pem  
ssl/private/local\_doc\_one.pem > /data/docker/nginx/ssl/private/local\_doc\_one.pem

doc\_one/docker-compose.yml > ~/doc\_one/docker-compose.yml  
.env > ~/doc\_one/.env

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

1. Изменить в файле ~/doc\_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 опубликован doc.one. Далее нужно изменить в файле /data/docker/nginx/conf/vhost.conf доменное имя на необходимое. 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. Развертывание выполняется из образов:

proxy-ssl  
registry.team-pravo.ru/doc-one/backend  
registry.team-pravo.ru/doc-one/frontend  
registry.team-pravo.ru/doc-one/docone-mysql  
memcached  
rabbitmq  
Образы находятся в файлах установки, либо в реестре Docker Private Registry.

Образы находятся в файлах установки, либо в реестре Docker Private Registry.

Если образы в файлах, нужно загрузить образы:

docker load < [имя\_файла]

Если образы в реестре Docker Private Registry, необходимо:

подключиться к реестру:

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

загрузить образы:

docker pull [имя\_образа]

1. Из каталога проекта ~/doc\_one начать развертывание и запуск контейнеров:  
   docker-compose up -d

Должен произойти успешный запуск контейнеров:

Starting doc\_one\_web-frontend\_1 ... done  
Starting doc\_one\_web-backend\_1 ... done  
Starting doc\_one\_service-mysql\_1 ... done  
Starting doc\_one\_service-memcached\_1 ... done

|  |
| --- |
|  |
| Внимание! После разворачивания Системы требуется из каталога проекта ~/ssl\_proxy\_ffd начать развертывание и запуск контейнеров:  docker-compose up -d  Должен произойти успешный запуск контейнера:  Starting ssl\_proxy\_ffd\_proxy-ssl\_1 ... done. |
|  |

1. Выполнить создание структуры базы данных и пользователя, и отключение мониторинга:

docker exec -ti doc\_one\_web-backend\_1 /bin/bash  
php artisan init:local:instance  
php instances migrate  
php artisan init:local:owner  
echo 'pinba.enabled=Off' > /usr/local/etc/php/conf.d/docker-php-ext-pinba.ini  
exit

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

1. Связать Систему с адресом хоста в файле hosts:

локальным ip:

127.0.0.1 api.doc.one doc.one [ ваше наименование инстанса ].doc.one -> (instance.doc.one)

либо внешним ip:

ip api.doc.one doc.one [ ваше наименование инстанса ].doc.one -> (instance.doc.one)

1. Выполнить вход **https://instance.doc.one**:

Login superadmin@superadmin.com;

Passwоrd secret.

Список рисунков

[Рисунок 1. Схема архитектурного решения 7](#_Toc44075014)

Список таблиц

[Таблица 1. Сокращения, термины и определения 3](#_Toc44075015)

[Таблица 2. Права доступа пользователей 4](#_Toc44075016)