КРАФТЛАБ

ООО «КрафтЛаб»

RVS. Руководство администратора.

1 ОБЗОР АРХИТЕКТУРЫ

Инсталляция RVS состоит из нескольких компонентов и сервисов (Рисунок 1.1). В следующих разделах рассмотрим каждый компонент и его роль в системе по отдельности.

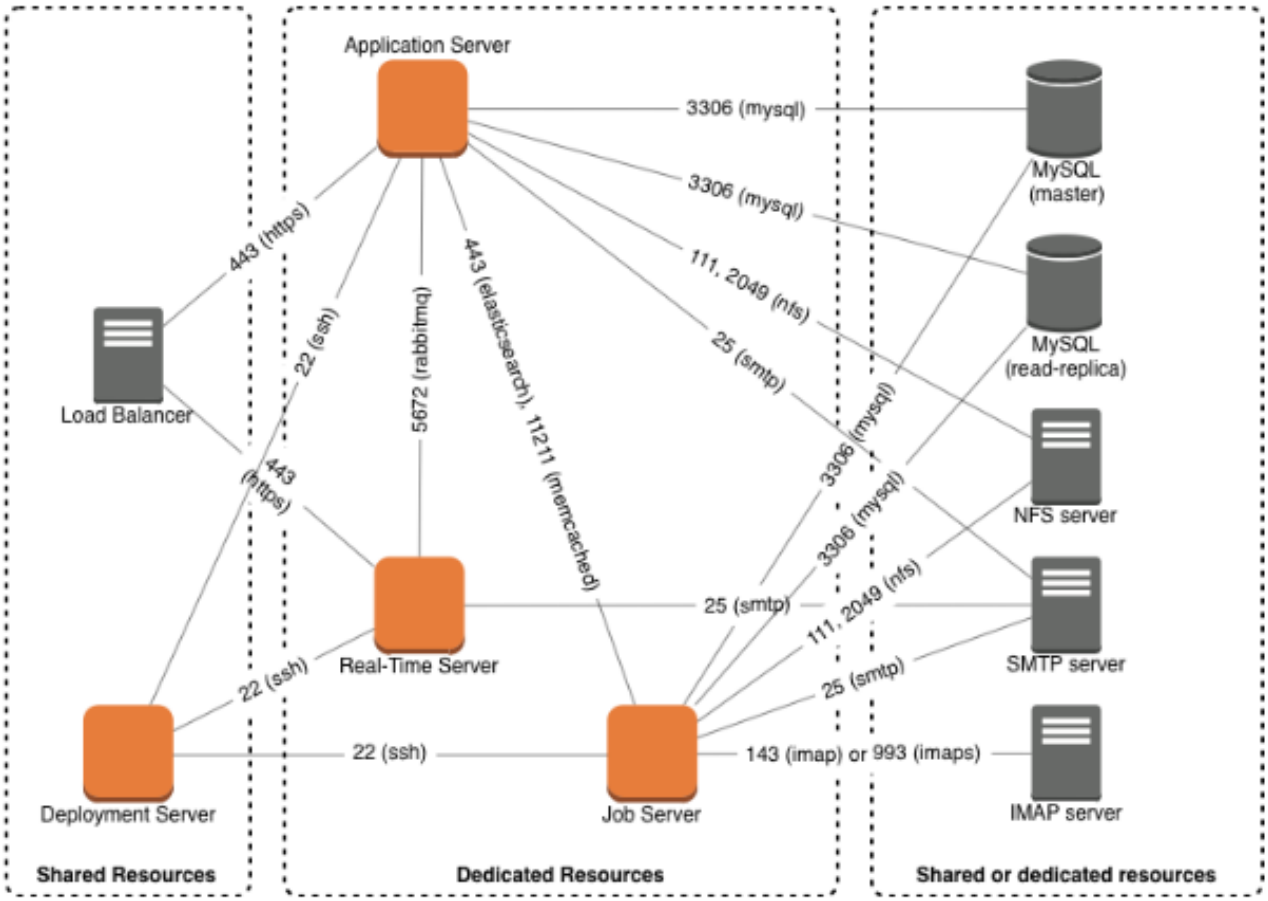

Рисунок 1.1 – Топология инсталляции RVS

1.1 Load Balancer – балансировщик нагрузки

Балансировщик нагрузки балансирует web-запросы к RVS между доступными application и real-time серверами. Также балансировщик нагрузки выполняет SSL-терминацию перед перенаправлением запроса к одному из серверов.

1.2 Application server – сервер приложения

Сервер приложения принимает основные web-запросы. Сервисы, указанные в таблице 1.1, работают на сервере приложения.

Таблица 1.1 – Сервисы, работающие на сервере приложения.

Имя сервиса	Комментарий
Docker[1]	Платформа контейнеризации.
r-service	Сервис приложения, который выполняет код RR Tech Service Management. Использует

[1] docker работает на самом хосте, все остальные сервисы работают внутри или как контейнеры

	nginx внутри как reverse-proxy для входящих web-запросов
postfix	Перенаправляет почту на SMTP сервер
s6	Управление процессами (process superviser)
monit	Сервис мониторинга

1.3 Real-time server – сервер реального времени

Сервер реального времени отвечает за следующие функции:

- отправку изменений для представления inbox клиентам;
- детекцию коллизий;
- отправку нотификаций.

Сервисы, указанные в таблице 1.2, работают на сервере отложенных задач.

Таблица 1.2 – Сервисы, работающие на сервере отложенных задач.

Имя сервиса	Комментарий
Docker[1]	Платформа контейнеризации.
faye	Сервис обработки соединений реального времени. Использует nginx внутри как reverse-proxy для входящих web-запросов. Данный сервис будет пытаться использовать websocket соединение при возможности, в ином случае будет использовать http как запасной вариант.
rabbitmq	Очередь сообщений, получает сообщения от сервера приложения.
sneakers	Сервис обработки сообщений. Получает сообщения из rabbitmq и передает их по соединению, установленному faye
redis	База данных для хранения состояния коллизий
postfix	Перенаправляет почту на SMTP сервер
s6	Управление процессами (process superviser)
monit	Сервис мониторинга

1.4 Сервер базы данных

Приложение RVS хранит информацию внутри одной MySQL схемы. База внутри MySQL сервера и её таблицы автоматически создаются при первой установке.

Второй, read-only, сервер базы данных может использовать для разгрузки требовательных запросов.

RVS может работать с серверами баз данных, которые шифруют информацию.

1.5 Сервер хранения файлов

Вложения, файлы логов для импорта, аватары пользователей и многие другие файлы требуют постоянного хранилища файлов. RVS поддерживает хранение

файлов в любом сетевом хранилище, которое можно использовать в вашем дистрибутиве ОС.

Сервера приложений и отложенных задач должны иметь общий доступ к этим файлом средствами примонтированных дисков.

1.6 Сервер SMTP (исходящей почты)

Сервер исходящей почты отвечает за отправку уведомлений:

- пользователям RR Tech Service Management;
- системным администраторам RR Tech Service Management.

1.7 Сервер IMAP (входящей почты)

Приложение RVS может обрабатывать входящие письма для того, чтобы создавать новые запросы или добавлять комментарии к существующим сущностям, таким как запросы и задачи. Чтобы данный функционал работал, сервер отложенных задач должен иметь возможность обращаться к почтовому ящику средствами IMAP(S).

1.8 Сервер Deployment (развертывания)

Сервер развертывания используется во время первичной установки и установки обновлений. Он использует Ansible для автоматизации процессов развертывания. Сервер развертывания может быть общим для нескольких сред (dev, qa, prod и т.д.).

2 РАЗВЕРТЫВАНИЕ

2.1 Требования перед установкой

Перед установкой приложения RVS должно быть соблюдено несколько требований.

2.1.1 Доменные имена

RVS требует доменное имя, под которым будут доступны все пространства и сервисы. Как пример, для компании example таким доменным именем может быть r-service.example.com.

Каждое пространство в системе имеет свой поддомен. К примеру, пространства hr и aho будут доступны по следующим адресам:

- https://hr.r-service.example.com
- https://aho.r-service.example.com

Помимо поддоменов каждого пространства, система также использует несколько поддоменов по умолчанию (Таблица 2.1).

Таблица 2.1 – Поддомены системы.

Поддомен	Комментарий
api	REST API Endpoint
graphql	GraphQL API Endpoint
oauth	OAuth Endpoint
realtime	Endpoint сервера реального времени
assets0	Список поддоменов, по адресам которых возможно получить ресурсы приложения
assets1	

assets2	(статические файлы – CSS, JavaScript, изображения и шрифты). Несколько поддоменов позволяют браузерам получать несколько ресурсов одновременно.
assets3	
io	Endpoint сервиса коротких ссылок

Если требуется, то стандартные поддомены возможно изменить. Так как приложение использует большое количество поддоменов, то рекомендуется использовать wildcard SSL сертификаты. Данный сертификат должен быть установлен на балансировщике нагрузки.

При развертывании нескольких сред (к примеру, QA и Production) каждой среде необходим отдельный домен (Таблица 2.2).

Таблица 2.2 – Примеры доменов при развертывании нескольких сред.

Среда	Домен
Development	`r-service-dev.example.com`
Test	`r-service-test.example.com`
Acceptance	`r-service-qa.example.com`
Production	`r-service.example.com`

2.1.2 Балансировщик нагрузки

Следующие HTTP-заголовки должны быть в каждом web-запросе, который перенаправляет балансировщик нагрузки:

- Host: убедитесь, что он включает в себя исходный домен, так как он используется для того, что определить пространство, к которому обращается пользоваться;
- X-Forwarded-For: исходный IP адрес;
- X-Forwarded-Proto: должно быть равно https.

Чтобы убедиться, что сервера приложений готовы принимать и обрабатывать запросы, балансировщик нагрузки может отправлять следующие запросы:

- /_.gif: проверяет, что nginx внутри контейнера доступен, статус ответа должен быть в диапазоне от 200 до 299;
- /teapot: проверяет, что сервер приложения (unicorn) доступен. Отвечает “I am a teapot”. Используйте данную опцию только тогда, когда ваш балансировщик умеет проверять тело ответа, не только статус ответа.

Чтобы убедиться, что сервера реального времени готовы принимать и обрабатывать запросы, балансировщик нагрузки может отправлять следующие запросы:

- /teapot: проверяет, что сервер приложения (faye) доступен. Отвечает “I am a teapot”. Используйте данную опцию только тогда, когда ваш балансировщик умеет проверять не только тело ответа, но и статус ответа.

2.1.3 Требования к ресурсам

Данный блок рассматривает требования к ресурсам для разных сред. Данные требования – лишь стартовая точка. Когда среда начинает расти, её ресурсы должны быть отмасштабированы соответственно. Благодаря постоянному мониторингу

нагрузки вы должны суметь определить, что и в каких объемах необходимо отмасштабировать.

Рекомендуется развернуть две среды – QA и Production. Также возможно развернуть Development и Test среды, для того чтобы полностью соответствовать подходу DTAP, но данный подход добавит сложности и лишней работы по поддержке каждый среды.

1. Development

Таблица 2.3 – Development-среда.

Сервер	Количество	Количество процессоров	Объем ОЗУ	Объем памяти (дисков)
Application server (сервер приложения)	1	2	8 GB	30 GB
Job server (сервер отложенных задач)	1	4	16 GB	30 GB
Real-time server (сервер реального времени)	1	2	4 GB	30 GB
Сервер базы данных	1	2	4 GB	30 GB
Сервер хранения файлов	1	-	-	25 GB
SMTP сервер[2]	1	-	-	-
IMAP сервер[2]	1	-	-	-
Deployment server (сервер развертывания)[3]	1	1	0.5 GB	5 GB

2. Test

Такие же требования, как и к Development среде (Таблица 2.3).

3. Acceptance (QA)

Таблица 2.4 – Acceptance (QA).

Сервер	Количество	Количество процессоров	Объем ОЗУ	Объем памяти (дисков)
Application server (сервер приложения)	2	2	8 GB	30 GB
Job server (сервер отложенных задач)	1	4	16 GB	30 GB
Real-time server (сервер реального времени)	1	2	4 GB	30 GB
Сервер базы данных[4]	2	2	8 GB	50 GB
Сервер хранения файлов	1	-	-	50 GB
SMTP сервер[5]	1	-	-	-
IMAP сервер[5]	1	-	-	-
Deployment server (сервер развертывания)[6]	1	1	0.5 GB	5 GB

[2] SMTP и IMAP сервера – общие ресурсы, которые обычно уже доступны внутри организации.
[3] Сервер развертывания общий для всех сред
[4] Вторая база данных является read-only репликой
[5] SMTP и IMAP сервера – общие ресурсы, которые обычно уже доступны внутри организации.
[6] Сервер развертывания общий для всех сред

4. Production

Таблица 2.5 – Production-среда.

Сервер	Количество	Количество процессоров	Объем ОЗУ	Объем памяти (дисков)
Application server (сервер приложения)	2	2	8 GB	30 GB
Job server (сервер отложенных задач)	2	4	16 GB	30 GB
Real-time server (сервер реального времени)	1	2	4 GB	30 GB
Сервер базы данных[4]	2	2	16 GB	50 GB
Сервер хранения файлов	1	-	-	100 GB
SMTP сервер[5]	1	-	-	-
IMAP сервер[5]	1	-	-	-
Deployment server (сервер развертывания)[6]	1	1	0.5 GB	5 GB

2.2 Требования к ПО

2.2.1 Application, job, search, realtime сервера

ОС

Таблица 2.6 – ОС.

Название	Версия
Red Hat Enterprise Linux	Мажорная версия: >= 8 Минорная версия: последняя
Ubuntu LTS	>=22.04 LTS

Docker

Должны быть установлена последняя версия Docker и Docker Compose.

Если вы используете стандартный logging driver (json), то убедитесь, что он настроен выполнять ротацию логов как описано в данном источике. Рекомендуется использовать следующей настройки в daemon.json:

```
{
      "log-driver": "json-file"
      "log-opts": {
      "max-size": "100m",
      "max-file": "10",
      }
}
```

Настройки ОС

Требуются следующие настройки ОС, указанные в таблице 2.7.

Таблица 2.7 – Настройки ОС.

Сервер	Имя	Значение

Все	Количество открытых файлов для процесса (soft limit)	8192
Все	Количество открытых файлов для процесса (hard limit)	65536
Realtime	`fs.file-max`	2097152
Realtime	Количество открытых файлов для процесса (hard limit)	100000
Search	`vm.max_map_count`	262144

Данные настройки могут быть автоматически заданы ansible-скриптом во время процесса развертывания. Для того, чтобы выполнить данные настройки, пользователь на удаленном хосте должен иметь sudo-права.

2.2.2 База данных

Для базы данных требуется MySQL (или MySQL-совместимый) сервер. В данный момент MySQL 5.7 и младше не поддерживается – рекомендуется использовать MySQL 8.0 и старше. В подсистемы хранения должна использоваться InnoDB версии v1.1 или старше.

2.2.3 Deployment сервер

На Deployment сервере должен быть установлен Ansible последней версии.

2.2.4 Сетевое взаимодействие

Балансировщик

Таблица 2.8 – Входящие соединения.

Источник	Порт	Протокол	Описание
Внешняя сеть	443	https	Web и API запросы. Websocket данные для сервера реального времени.

Таблица 2.9 – Исходящие соединения.

Назначение	Порт	Протокол	Описание
Application server (сервер приложения)	443	https	Web и API запросы.
Real-time server (сервер реального времени)	443	https	Websocket данные.

Application server (сервер приложения)

Таблица 2.10 – Входящие соединения.

Источник	Порт	Протокол	Описание
Балансировщик	443	https	Web и API запросы.

Таблица 2.11 – Исходящие соединения.

Назначение	Порт	Протокол	Описание
Job server (сервер отложенных задач)	443	https	Запросы к движку поиска
Job server (сервер отложенных задач)	11211	tcp	Запросы к базе данных кэширования.

Database server (сервер базы данных)	3306	tcp	Запросы к базе данных.
SMTP сервер	25	smtp	Исходящая почта
Real-time server	5672	tcp	Публикация сообщений в очередь сообщений

Job server (сервер отложенных задач)

Таблица 2.12 – Входящие соединения.

Источник	Порт	Протокол	Описание
Application server (сервер приложения)	443	https	Запросы к движку поиска
Application server (сервер приложения)	11211	tcp	Запросы к базе данных кэширования

Таблица 2.13 – Исходящие соединения.

Назначение	Порт	Протокол	Описание
Database server (сервер базы данных)	3306	tcp	Запросы к базе данных.
SMTP сервер	25	smtp	Исходящая почта
IMAP сервер	143, 993	imap, imaps	Входящая почта
Real-time server	5672	tcp	Публикация сообщений в очередь сообщений

Real-time server (сервер реального времени)

Таблица 2.14 – Входящие соединения.

Источник	Порт	Протокол	Описание
Балансировщик	443	https	Web и API запросы.

Таблица 2.15 – Исходящие соединения.

Назначение	Порт	Протокол	Описание
IMAP сервер	143, 993	imap, imaps	Входящая почта

Database server (сервер базы данных)

Таблица 2.16 – Входящие соединения.

Источник	Порт	Протокол	Описание
Application server (сервер приложения)	3306	tcp	Запросы к базе данных.
Job server (сервер отложенных задач)	3306	tcp	Запросы к базе данных.

Исходящие соединения. Соединения между несколькими БД не включены в данный список.

Deployment server (сервер развертывания)

Входящие соединения. Данный компонент не принимает никаких соединений.

Таблица 2.17 – Исходящие соединения.

Источник	Порт	Протокол	Описание
Application server (сервер приложения)	22	ssh	Ansible
Job server (сервер отложенных задач)	22	ssh	Ansible
Real-time server (сервер реального времени)	22	ssh	Ansible

2.2.5 Ящики электронной почты

Приложение RVS отправляет и получает почту с нескольких ящиков, которые должны быть созданы и доступны перед началом установки.

В таблице 2.18 представлен список всех ящиков, которые могут быть использованы приложением и пример их наименования.

Таблица 2.18 – Ящики электронной почтв

Environment переменная, отвечающая за ящик	Пример
ITRP_INBOUND_MAILBOX	r-service@example.com Письмо, отправленное на r-service+account1@example.com и r-service+account2@example.com (то есть при использовании синтаксиса со знаком +) должно оказаться в ящике r-service@example.com
ITRP_ERRORS_MAILBOX	r-service-errors@example.com
ITRP_INFO_MAILBOX	r-service-info@example.com
ITRP_SUPPORT_MAILBOX	r-service-support@example.com
ITRP_NOREPLY_MAILBOX	noreply@example.com
MONIT_ALERT_MAILBOX	r-service-monitor@example.com
MONIT_FROM_MAILBOX	r-service-monitor@example.com

2.2.6 Конфигурация

Скрипты для развертывания используют информацию, указанную в инвентаре, разных переменных группы и environment, docker compose файлах.

Инвентарь, переменные группы и docker compose файлы создаются во время этапа “Генерация конфигурации”. Информация, из которой эти данные генерируется, берется из так называемой rack-конфигурации, которая находится по пути environments/my-company/rack.yml.

2.2.7 Rack-конфигурация

Файл состоит из двух частей:

- секции vars, которая содержит общую информацию о среде;
- секции servers, которая содержит информацию о конфигурации сервера и определяет компоненты, которые будут запущены на каждом сервере.

Примером такой rack-конфигурации, которую можно использовать для начала, является файл environments/example.com/rack.yml

Секция vars описывает общую конфигурацию среды. Пример:

```
vars:
repository_prefix: images.example.com/ops/itrp
storage_directory: /var/r-service/storage
docker_compose_command: docker-compose
```

Переменные, указанные в таблице 2.19, доступны для настройки.

Таблица 2.19 – Переменные, доступные для настройки.

Название	Описание
`data_directory`	Путь до базовой директории, где будут располагаться все файлы данной среды. По умолчанию: `/opt/itrp`
`docker_data_directory`	Путь до директории, где Docker хранит свои данные. По умолчанию: `/var/lib/docker`
`docker_compose_command`	Используйте “`docker-compose`” если вы не используется плагин Docker Compose. По умолчанию: `docker compose`
`remote_user`	Пользователь, от лица которого будут выполнены все действия в Ansible-плейбуках. По умолчанию: `deployer`
`remote_group`	Группа пользователя, которая будет использована для выполнения действий в Ansible-плейбуках. По умолчанию: `deployer`
`repository_prefix`	Префикс пути Docker репозитория, который будет использоваться для загрузки образов. По умолчанию: `docker.rr-th.com/itrp`
`storage_directory`	Путь, по которому доступен примонтированный ко всем серверам сервер хранения файлов
`use_search_cluster`	Используйте «`true`» чтобы развернуть движок поиска в кластере. В ином случае не задавайте
`use_proxy`	Используйте «`true`» если прокси-сервер необходим для того, чтобы подключиться к интернету. По умолчанию: `false`
`use_secure_db`	Используйте “`true`”, если ваш сервер БД использует защищенные соединения. По умолчанию: `false`
`env.http_proxy` `env.https_proxy`	Адрес прокси сервера. Данные переменные доступны только тогда, когда `use_proxy - true`

Стандартные значения для каждой переменной определены в файле plays/templates/default_vars.yml. Этот файл также содержит теги каждого docker-образа.

Секция servers содержит список всех серверов и компоненты приложения, которые на них установлены. Пример:

```
servers:
192.168.1.100:
categories:
- web
services:
rails:
```

Каждый сервер может иметь следующие секции:

- environment – содержит environment переменные, которые перезапишут environment переменные из .env файлов (см. след. секцию);
- categories – определяет роль сервера. Валидные значения: web, job, realtime, search (или несколько ролей одновременно);
- services – определяет сервисы, которые работают на данном сервере. Каждый сервис имеет следующие параметры:
  a. scale – количество контейнеров которое должно быть создано. По умолчанию – 1. Если задать данное значение в 0, то данный контейнер не будет создан по умолчанию (используется для core сервиса, который нужен для выполнения ad-hoc комманд);
  b. environment – содержит environment переменные, которые перезапишут значения из environment блока сервера (см. выше) и .env файлов (см. след. секцию).

Пример:

```
search:
environment:
ES_JAVA_OPTS: "-Xms2g -Xmx2g"
```

2.2.8 Переменные окружения (.env файлы)

Разные сервисы приложения настраиваются с помощью переменных окружения средствами, называемыми .env-файлов. Эти файлы настраиваются заказчиком и должны быть перемещены в папку environments/my-company для их распространения на сервера скриптами.

Есть четыре файла:

- .env: переменные для всех серверов;
- .env.web: переменные только для серверов роли web;
- .env.job: переменные только для серверов роли job;
- .env.realtime: переменные только для серверов роли realtime.

Файлы с примерами, которое могут быть использованы как стартовая точка для ваших файлов, находятся в директории environments/example.com.

В таблице 2.20 представлен список всех доступных переменных.

Таблица 2.20 – Доступные переменные.

Название	Описание

	Основные переменные
ITRP_DOMAIN	Доменное имя, по которому будет доступна система
ITRP_API_SUBDOMAIN	Поддомен, по которому будет доступно REST API в системе. По умолчанию: api
ITRP_GRAPHQL_SUBDOMAIN	Поддомен, по которому будет доступно GraphQL API в системе. По умолчанию: graphql
ITRP_OAUTH_SUBDOMAIN	Поддомен, по которому будет доступны OAuth v2 endpoints в системе. По умолчанию: oauth
ITRP_LICENSE	Лицензионный ключ для текущей среды. Предоставляется вендором при первичной инсталляции и обновлениях
ITRP_SHOW_QA_MESSAGE	Используйте “true”, чтобы отображать баннер “Вы подключены к QA среде R-Service”. По умолчанию: false
ITRP_COOKIE_SECRET_TOKEN	Задайте как случайный токен достаточной длинны
ITRP_OTP_SECRET	Задайте как случайный токен достаточной длинны
ITRP_OTP_SALT	Задайте как случайный токен достаточной длинны
ITRP_ERRORS_MAILBOX	Почтовый ящик, с которого будут отправляться уведомления об ошибках (HTTP 500)
ITRP_INFO_MAILBOX	Почтовый ящик, который будет получать отчеты об использовании каждую неделю и месяц
ITRP_NOREPLY_MAILBOX	Почтовый ящик, который будет использоваться как from адрес, если ответ на письмо не предусмотрен
ITRP_SUPPORT_MAILBOX	Используется в письмах регистрации как from и reply-to адрес
ITRP_INBOUND_MAILBOX	Ящик входящей почты. Используется для добавления комментариев к запросам, проблемам, релизам, изменениям, задачам и многому другому в системе.mailto:noreply@example.com
ITRP_BOUNCED_MAILBOX	Используется как return-path во всех письмах
ITRP_ASSESTS_HOSTS	Список поддоменов, разделенный запятой, по адресам которых возможно получить ресурсы приложения (статические файлы – CSS, JavaScript, изображения и шрифты). Несколько поддоменов позволяют браузерам получать несколько ресурсов одновременно.
ITRP_SSL	Всегда true. Показывает то, что приложение доступно только по https.
ITRP_DEFAULT_SUPPORT_URL	URL, который будет указан в письмах регистрации
ITRP_SUPPORT_ACCOUNT_ID	ID Пространства, которое будет использоваться как support-пространство в среде. Именно с этого аккаунта будет доступна /support консоль
ITRP_INBOUND_FORWARDED_TO_ATTRIBUTE	Как описано в соответствующей статье БЗ, оригинальный reply-to ящик который включает директиву +<account_name> должен быть доступен если письмо переслано в ящик входящей почты. Используйте эту переменную для того, что бы задать имя нового header
ITRP_INBOUND_ALLOW_BLANK_RETURN_PATH	Письма без return-path оцениваются как спам и игнорируются по умолчанию. Задайте это значение в “true” если return-path очищается при передаче писем.
ITRP_INBOUND_CATCH_ALL	Задайте в “true” когда используется catch-all ящик для обработки входящих писем. Если эта переменная false – то тогда

	приложение будет использовать “+sitename” для установки соответствия пространства к письму
ITRP_INBOUND_EMAIL_ERRORS_ACCOUNT	Sitename пространства, где будут хранится ошибки приема входящих писем
ITRP_SECURE_COOKIES	Задайте в «false» если вход не работает. Некоторые балансировщики не передают зашифрованный cookie клиентам.
ITRP_SAML_DESTINATION	Задайте в «false» чтобы удалить свойство Destination в samlp:AuthnRequest. Это иногда необходимо для такого, чтобы предовратить redirect-loop в AD FS 2.1.
ITRP_TRUSTED_PROXIES	Укажите через запятую список доверенных прокси-клиентов, которые получают доступ к приложению из приватной подсети (к примеру, 172.16.0.0/12)
ITRP_PROXY_HOST	Адрес прокси сервера
ITRP_PROXY_PORT	Порт прокси сервера
ITRP_NO_PROXY_HOSTS	Разделенный запятой список адресов, которые не должны проксироваться
ITRP_ALLOWED_VIDEO_DOMAINS	Разделенный запятой список доменов, с которых возможно добавлять видео. По умолчанию: www.youtube-nocookie.com,player.vimeo.com
Короткие ссылки	
ITRP_SHORT_URL_DOMAIN	Поддомен для сервиса коротких ссылок. По умолчанию: io.ITRP_DOMAIN
ITRP_SHORT_URL_SCHEME	Протокол, который используется в коротких ссылках. По умолчанию: https://
ITRP_SHORT_URL_MAX_TOKENS	Максимальное количество коротких ссылок, которое может быть создано на пространство. По умолчанию: 1000000
ITRP_SHORT_URL_MAX_RESERVED	Максимальное количество коротких ссылок, которое может быть зарезервировано на пространство. По умолчанию: 10000
Хранение файлов	
STORAGE_MAX_FILESIZE	Максимальный размер файла, который может быть загружен в мегабайтах. По умолчанию: 20
STORAGE_LOCAL_SECRET_KEY	Задайте как случайный токен достаточной длинны
База данных	
DB_WRITER_HOST	Адрес основного сервера базы данных
DB_WRITER_PORT	Порт основного сервера базы данных
DB_WRITER_USERNAME	Имя пользователя сервера базы данных, под которым будет пытаться авторизоваться приложение. Убедитесь, что у данного пользователя есть права на создание базы данных/схемы при первичной инсталляции
DB_WRITER_PASSWORD	Пароль сервера базы данных
DB_WRITER_DATABASE	Имя базы данных/схемы.
DB_READER_HOST	Адрес вторичного сервера базы данных. Используется для более read-требовательны задач: аналитики, экспортов и многого другого
DB_READER_PORT	Порт вторичного сервера базы данных
DB_READER_USERNAME	Имя пользователя сервера базы данных

DB_READER_PASSWORD	Пароль сервера базы данных
DB_READER_DATABASE	Имя базы данных/схемы.
DB_MAX_EXECUTION_TIME_SUPPORTED	Используйте «true», если сервер базы данных поддерживает системную переменную max_execution_time. По умолчанию: true
DB_SSL_CA_CERT	Задайте в «» (пустая строка) когда сервер базы данных не работает с SSL
DB_SSL_CA_PATH	Задайте в «» (пустая строка) когда сервер базы данных не работает с SSL
DB_SSL_CIPHER	Укажите предпочитаемый алгоритм для SSL-шифрования. По умолчанию: DHE-RSA-AES256-SHA
DB_SSL_MODE	Используйте “DISABLED” для того, чтобы выключить проверку SSL-сертификата. Доступные значения: • DISABLED • PREFERRED • REQUIRED (не проверяет, но требует – используйте для самоподписанных сертификатов) • VERIFY_CA • VERIFY_IDENTITY По умолчанию: VERIFY_IDENTITY
DB_TLS_CIPHERSUITES	Наборы шифров, которые допустимы для зашифрованных соединений, использующих TLSv1.3
DB_TLS_MIN_VERSION	Минимальная допустимая версия протокола TLS. Доступные значения: • 1 для TLSv1 • 2 для TLSv1.1 • 3 для TLSv1.2 • 4 для TLSv1.3 По умолчанию: 3
DB_TLS_MAX_VERSION	Максимальная допустимая версия протокола TLS
Кэширование (Memcached)	
MEMCACHED_HOST	Адрес сервера Memcached; обычно – адрес job-сервера, на котором работает memcached
MEMCACHED_MEMORY	Максимальное количество памяти, которое можно использовать для хранения объектов. По умолчанию: 512
MEMCACHED_SECRET_TOKEN	Задайте как случайный токен достаточной длинны
Redis	
REDIS_HOST	Адрес сервера Redis; обычно – адрес realtime-сервера, на котором работает redis
REDIS_PORT	Порт для подключения к redis; обычно 6379
REDIS_PASSWORD	Задайте пароль если ваш сервер Redis требует его для подключения
Поиск (OpenSearch)	

ELASTICSEARCH_URLS	Адреса поисковых серверов, разделенных запятой. Могут включать протокол и порт (как пример - https://search1.internal:1234,https://search2.internal:1234)
Clacks (сервис обработки входящей почты)	
CLACKS_ADDRESS	Адрес IMAP(S) сервера
CLACKS_PORT	Порт сервера; обычно 143 или 993
CLACKS_USERNAME	Имя пользователя
CLACKS_PASSWORD	Пароль
CLACKS_ENABLE_SSL	Используйте “false”, если используется IMAP вместо IMAPS
CLACKS_MAILBOX	Папка, который необходимо проверять на наличие входящих писем. По умолчанию: INBOX
CLACKS_ARCHIVEBOX	Папка, куда будут перемещаться обработанные сообщения, к примеру ARCHIVE. Важно сохранять размер папки INBOX маленьким – IMAP начинает тормозить когда в одной папке много писем.
CLACKS_DELETE_AFTER_FIND	Используйте “false” чтобы оставлять все письмо в INBOX. Только для отладки
Функциональность реального времени	
RABBIT_MQ_HOST	Адрес RabbitMQ; обычно адрес realtime-сервера
FAYE_URL	Публичный url для realtime сервера. По умолчанию: https://realtime.ITRP_DOMAIN
FAYE_SECRET_TOKEN	Задайте как случайный токен достаточной длинны
FAYE_REDIS_HOST	Адрес Redis; обычно – адрес realtime-сервера, на котором работает redis
Генерация PDF	
PDFGENERATOR_URL	Адрес сервиса генерации pdf. Обычно – https://pdf (используется внутренняя сеть docker). По умолчанию: https://pdf.ITRP_DOMAIN
Почта - DKIM	
MAIL_DKIM_ENABLED	Используйте “true” чтобы включить DKIM аутентификацию. По умолчанию: false
MAIL_DKIM_DOMAIN	Домен для DKIM. По умолчанию: ITRP_DOMAIN
MAIL_DKIM_SELECTOR	Selector, добавляемый к домену, используется для поиска информации о публичном ключе DKIM. По умолчанию: default

MAIL_DKIM_PRIVATE_KEY	Приватный ключ DKIM, который используется для генерации TXT-записей.
	
	Чтобы сгенерировать пару публичный/приватный ключ, которая может быть использована как значение MAIL_DKIM_PRIVATE_KEY:
	`$ openssl genrsa -out dkim.private.key`
	`2048`
	`$ openssl rsa -in dkim.private.key -out dkim.public.key -pubout -outform PEM`

Остальное	

GOOGLE_MAPS_JAVASCRIPT_API_KEY	Используйте ваш API-ключ для Google Maps. Требуется для использования приложения Google Maps из магазина приложений
MONIT_ALERT_MAILBOX	Почтовый ящик, который будет получать уведомления от monit когда сервисы перезапускаются или когда они не могут быть перезапущены; обычно – тот же ящик, что и ITRP_ERRORS_MAILBOX
MONIT_FROM_MAILBOX	Адрес, используемый как From в письмах от monit
POSTFIX_RELAY_HOST	Адрес SMTP сервера
POSTFIX_SMTP_TLS_SECURITY_LEVEL	Возможные значения: none, may или encrypt. По умолчанию: may. См. https://www.postfix.org/postconf.5.html#smtp_tls_security_level
NGINX_ADD_X_FORWARDED_HEADERS	Задайте в «true» чтобы добавить заголовки “X-Forwarded-For” и “X-Forwarded-Proto” к запросам, в случае если ваш балансировщик этого не делает. Не рекомендуется
S6_LOGGING_SCRIPT	Конфигурация логгирования внутри контейнера. По умолчанию: T 1 n10 s104857600 T
UNICORN_WORKER_PROCESSES	Количество worker-процессов Unicorn (сервера веб-приложения)
TZ_LOG	Часовой пояс, используемый при логгировании. По умолчанию: UTC

3 УСТАНОВКА

Следующие шаги должны быть выполнены, чтобы установить или обновить приложение:

- загрузить дистрибутив;
- сгенерировать конфигурационные файлы;
- сделать бэкап БД;
- выполнить первичную конфигурацию серверов;
- загрузить образы приложения;
- настроить сервера;
- остановить все работающие сервисы;
- скопировать актуальную конфигурацию на сервера;
- выполнить миграции БД;
- обновить сервер поиска;
- запустить сервисы.

Если не указано другое, то все данные команды выполняются на deployment-сервере.

3.1 Загрузить дистрибутив

Для каждой версии приложения предоставляется tar-архив который содержит:

- Ansible скрипты;
- команды для загрузки образов из docker-репозитория компании PP-Tex.

Распакуйте tar-архив на deployment-сервере:

```
$ mkdir -p ~/r-service/r123
$ tar -xvf r-service-r123.tar.gz -C ~/r-service/r123
```

Рекомендуется создавать отдельную директорию под каждый релиз, чтобы в случае чего откат на предыдущую версию был максимально прост. В примере используется версия r123 как пример.

3.2 Сгенерировать конфигурационные файлы

Переместите файлы, содержащие rack-конфигурацию и environment переменные в директорию environments/my-company/

После этого, выполните следующую команду:

```
$ ansible-playbook plays/generate-configuration.yml
```

Эта команда выполняет Ansible playbook, который сгенерирует множество конфигурационных файлов, используемых во время установки.

3.3 Бэкап базы данных

Создайте бэкап базы данных который может быть использован на случай если потребуется откатить обновление.

3.4 Выполнить первичную конфигурацию серверов

Важно! Выполняйте данный шаг только во время первичной установки приложения. Данный шаг должен быть пропущен во время обновления приложения.

Выполните следующую команду:

```
$ ./scripts/bootstrap -e my-company -u <user>
```

Если используется paswordless-пользователь, то добавьте аргумент -k <private_key> чтобы передать приватный ключ вместо пароля.

Эта команда создаст пользователя “deployer” (uid=2000) который будет использоваться во время установки.

Запуск данного скрипта требует, чтобы remote_user (который указывается в rack-конфигурации) имел sudo-права.

3.5 Загрузить образы приложения

Образы контейнеров, содержащих последние версии сервисов приложения, загружаются автоматически из docker-репозитория компании PP-Tex.

Выполните следующую команду:

```
$ ./scripts/download -e my-company
```

Если используется paswordless-пользователь, то добавьте аргумент -k <private_key> чтобы передать приватный ключ вместо пароля.

Эта команда запросит у вас логин и пароль от docker-репозитория компании PP-Tex и загрузит образы контейнеров на соответствующие им сервера.

3.6 Настроить сервера

Пропустите данный шаг, если данная настройка выполняется сторонними инструментами конфигурации серверов (таких как Puppet).

Выполните следующую команду:

```
$ ./scripts/configure -e my-company
```

Эта команда настроит системные настройки и настройки ядра.

Запуск данного скрипта требует, чтобы remote_user (который указывается в rack-конфигурации) имел sudo-права.

3.7 Остановить все работающие сервисы

Выполните следующую команду:

```
$ ./scripts/down -e my-company
```

Эта команда остановит все текущие работающие сервисы на серверах (средствами docker-compose).

3.8 Скопировать актуальную конфигурацию на сервера

Выполните следующую команду:

```
$ ./scripts/provision -e my-company
```

Эта команда скопирует актуальную конфигурацию на сервера, предварительно создав необходимые директории.

Запуск данного скрипта требует, чтобы remote_user (который указывается в rack-конфигурации) имел sudo-права.

3.9 Выполнить миграции БД

Перед выполнениями миграций БД требуется запустить сервер поиска.

Если используется кластер:

```
$ ./scripts/up -e my-company -l search
```

Иначе, при использовании одного сервера, залогиньтесь на job-сервер, на котором работает сервер поиска и выполните следующие команды:

```
$ cd /opt/itrp
$ docker compose up -d nginx-search search
```

После этого выполните следующую команду:

```
$ ./scripts/migrate -e my-company
```

Важно! При первичной установке откройте файл plays/run-migrations.yml и удалите комментарии на заданиях “create database” и “create schema”.

3.10 Обновить сервер поиска

После установки или обновления приложения информация внутри сервера поиска должна быть пересоздана.

Залогиньтесь на сервер, на котором работает сервер Memcached и выполните следующие команды:

```
$ cd /opt/itrp
$ docker compose up -d memcached
$ itrp-exec rake es:import:all
```

3.11 Запустить сервисы

Выполните следующую команду:

```
$ ./scripts/up -e my-company
```

Эта команда запустит все сервисы на серверах (средствами docker-compose).

3.12 Создать первое пространство

Важно! Выполняйте данный шаг только во время первичной установки приложения. Данный шаг должен быть пропущен во время обновления приложения.

Для администрирования системы необходимо иметь «support» пространство. Из данного пространства будет доступна support-консоль, из которого возможно создавать другие пространства и администрировать систему.

Для его создания необходимо залогиниться на job-сервер и выполнить следующую команду:

```
$ itrp-exec rails c
```

После, выполните следующую команду (предварительно изменив значения на

```
Account.create_with_owner!(
    account: {
        id: 1,
        name: "R-Service Support",
        sitename: "rs-support",
        directory_service: false
    },
    person: {
        name: "<name>",
        primary_email: "your@email.here"
    }
)
```

ваши):

Данная команда создаст стандартное пространство с ID 1.

Задайте ID данного пространства в переменную ITRP_SUPPORT_ACCOUNT_ID и выполните шаги:

- остановить все работающие сервисы;
- скопировать актуальную конфигурацию на сервера;

- запустить сервисы.

После этого на указанную почту придет письмо с информацией по входу. После входа, настройте двухфакторную аутентификацию и только тогда вы сможете войти в support-консоль по ссылке https://rs-support.ITRP_DOMAIN/support.

4 ТЕСТИРОВАНИЕ

После установки или обновления приложения несколько проверок могут быть выполнены для того, чтобы убедиться, что все сервисы работают правильно.

Перед начало убедитесь, что:

- настройка «Отправлять уведомления по электронной почте» задана в «Всегда»;
- настройка «Показывать всплывающее окно уведомления» задана в «Всегда».

Данные настройки могут быть найдены в меню Мой профиль > Настройки уведомлений.

4.1 Импорт/Экспорт

Перейдите в вид “Команды” и экспортируйте несколько записей (команд) выбрав “Экспортировать все” из бургер-меню сверху таблицы (выглядит как три точки). Скачайте результат экспорта из вида “Экспорты”.

Это проверит то, что отложенные задачи работают и хранилище файлов было примонтированно правильно.

4.2 Входящая и исходящая почт0430

Создайте новый запрос, отправив письмо на входящий ящик электронной почты. Убедитесь, что в запросе есть какое-то уникальное слово (например, «яблоко» или «банан») – это поможет позднее при тестировании функциональности поиска.

Этот новый запрос должен отобразиться в виде «Запросы» в течение нескольких минут (письму может потребоваться много времени для того, чтобы прибыть на почтовый сервер).

Это проверит следующие моменты:

- сервис Clacks соединился с IMAP сервером и обрабатывает почту;
- исходящая почта работает корректно (вы должны были получить письмо о том, что новый запрос был создан);
- функциональность реального времени работает (вы должны были получить уведомление в браузере о том, что новый запрос был создан).

4.3 Вложения

Прикрепите вложение в запрос, созданный на прошлом шагу. Проверьте, что вы можете его скачать после сохранения запроса.

4.4 Поиск

Выполните поиск по уникальному слову, которое вы использовали, создавая запрос через почту. Этот запрос должен высветиться в результатах поиска.

Это проверит то, что сервер поиска работает корректно и индексы поиска корректно реиндексируются отложенными задачами.

4.5 Функциональность реального времени

Данная функциональность уже была проверена в шаге 4.2 «Входящая и исходящая почта».

4.6 PDF

Перейдите в вид “Дизайн PDF” и выберите дизайн названный “По умолчанию Dashboard”. Нажмите на “Создать образец” и выберите любую панель мониторинга. Проверьте, что PDF сгенерировался и выглядит правильно.

5 ОБСЛУЖИВАНИЕ СИСТЕМЫ

5.1 Support-консоль

Аккаунт, у которого такое же ID, как указано в ITRP_SUPPORT_ACCOUNT_ID переменной известен как “R-Service Support” пространство.

Люди, которые имеют роль Аналитика Сервис-деска в этом пространстве и включили двухфакторную аутентификацию могут получить доступ к support-консоли которая поддерживает несколько функций:

- управление пространствами;
- просмотр статистики и информации о разных сервисах приложения: БД, сервера поиска и серверов кэширования (Redis и Memcached);
- просмотр отложенных задач и их планирование;
- просмотр каждой из очереди сообщений и её размер.

Чтобы получить доступ к этой консоли, перейдите по ссылке /support внутри вашего пространства.