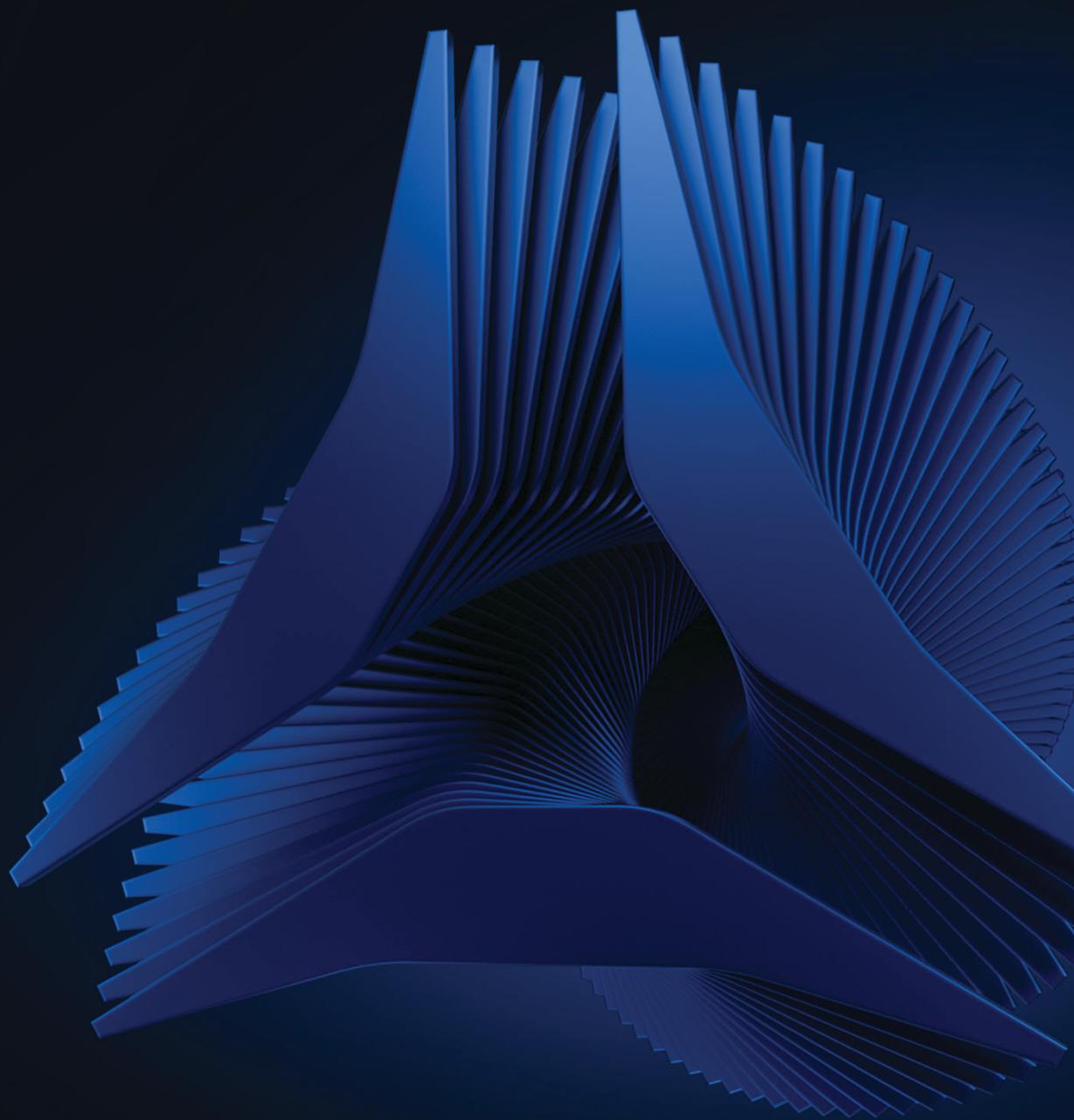




Руководство по эксплуатации системы автоматизированного развертывания веб-приложений «Деплойми»





О продукте

1.1 Основные понятия

1.1.1 Инструменты автоматизации

Dokku — консольное приложение для автоматизации развертывания и управления жизненным циклом веб-приложений на платформе под управлением ОС семейства Debian/Ubuntu/CentOS

Docker — программное обеспечение для автоматизации развёртывания и управления приложениями в средах с поддержкой контейнеризации

Dockerfile — конфигурационный файл веб-приложения для сборки в docker

Buildpack — набор скриптов определяющих тип приложения и автоматизирующих установку программной платформы и необходимых зависимостей

Ansible — система управления конфигурациями используется для автоматизации развертывания необходимого для работы Деплойми ПО

1.1.2 Программные платформы, фреймворки, СУБД

Node.js — программная платформа, основанная на движке V8 (транслирующем JavaScript в машинный код), превращающая JavaScript из узкоспециализированного языка в язык общего назначения

TypeScript — язык программирования, расширяющий возможности JavaScript

NestJS — фреймворк для создания масштабируемых серверных приложений на TypeScript

NATS — высокопроизводительная система обмена сообщениями с открытым исходным кодом для микросервисных архитектур

Redis — в данной реализации выполняет задачи системы обмена сообщениями

PostgreSQL — СУБД

React — JavaScript-библиотека для создания пользовательских интерфейсов

Next.js — Фреймворк для создания веб-приложений на React

Бэкенд — программно-аппаратная часть сервиса

Фронтэнд — клиентская сторона пользовательского интерфейса к программно-аппаратной части сервиса.

VPS — виртуальная машина под управлением ОС семейства Debian/Ubuntu



1.2 Цели и назначение

Программное решение «Деплойми» (далее - Деплойми) предназначено для управления автоматизацией развертывания и управления жизненным циклом веб приложений используя графический интерфейс браузерного веб-приложения. Основными возможностями системы являются

Администратор

- › Управление пользователями
- › Управление площадками
- › Управление справочниками

Пользователь

- › Создание площадок, на которых будет производиться развертывание приложений.
- › Развертывание приложения автоматически определяя их тип с помощью Heroku buildpacks
- › Разворачивать приложения с помощью Dockerfile
- › Создавать и управлять базами данных: привязывать к приложениям, экспортировать и импортировать дампы
- › Управлять переменными окружения приложения
- › Привязать к приложению домен
- › Устанавливать приложению базовую авторизацию
- › Устанавливать SSL сертификат Let's Encrypt

1.3 Архитектура

1.3.1 Бекенд

Бекенд Деплойми — микросервисное NestJS приложение. Схема архитектурного решения представлена в **приложении 1** руководства по эксплуатации

Микросервисы:

- › Auth Service — модуль авторизации, создание и обновление токенов
- › Dictionary Service — модуль справочников системы. Билдпаки, Типы приложений, Фреймворки
- › Dokku Service — модуль, взаимодействующий с пользовательской площадкой. Развертывание и управление dokku на площадке пользователя.
- › Notification Service — модуль уведомления пользователя, отправка электронной почты об изменении на площадке
- › User Service — модуль работы с данными пользователя.
- › Socket Service — оповещения пользователя о событиях, возникающих в процессе работы с приложением через websocket соединение.



API:

- › Admin API — Администрирование системы
- › User API — Использование системы
- › Взаимодействие микросервисов друг с другом осуществляется через NATS.
- › Взаимодействие системы с пользовательской площадкой осуществляется через Redis
- › Для хранения данных, генерируемых микросервисами, используется экземпляр PostgreSQL привязанный к конкретному микросервису

1.3.2 Фронтэнд

Фронтэнд Деплойми — два Next.js приложения: приложение администратора системы и приложение пользователя системы

1.3.3 Принцип работы системы

- › Пользователь регистрируется или проходит LDAP авторизацию в системе Деплойми
- › Пользователь заполняет форму подключения VPS, если привязанных площадок в системе еще нет
- › Деплойми, используя Ansible, производит на целевой VPS установку Dokku и дополнений к нему расширяющие функциональность базовой версии Dokku
- › Пользователь заполняет форму создания приложения
- › Деплойми выполняет на целевой площадке Dokku команды для создания приложения и подключения к нему нужных сервисов
- › Пользователь, руководствуясь подсказками выводимыми Деплойми, выполняет консольные команды для отправки кода на целевую площадку
- › Деплойми показывает в веб-интерфейсе статус развертывания приложения и позволяет манипулировать жизненным циклом веб-приложения



2.1 Бекенд

2.1.1 Запуск бекенда на одном узле (локальной машине)

Для запуска бекенда нужно установить следующие компоненты:

- > Nodejs версии LTS
- > PostgreSQL версии LTS
- > NATS версии LTS
- > Redis версии LTS
- > Создать экземпляры PostgreSQL, для упрощения запуска задать имена БД как показано значениях по умолчанию

Установить Nodejs зависимости

- > `$ npm install`

В директории docs описаны необходимые для запуска переменные окружения и их значения по умолчанию

Произвести запуск микросервисов (на примере микросервера Auth Service, аналогично запустить все микросервисы бекенда):

- > Выполнить миграции: `$ SERVICE=auth-service npm run migrations`
- > Запустить микросервис `$ SERVICE=auth-service npm run start`

2.1.2 Запуск бэкенда в условиях контейнеризации или на отдельных VPS

Бекенд поддерживает изоляцию микросервисов друг от друга, что позволяет создать кластерную инфраструктуру системы, выделив наиболее ресурсоемкие компоненты под балансировщик или же на узел с производительной конфигурацией. Архитектура кластера зависит от конкретных задач и рассчитывается индивидуально.

2.2 Фронтэнд

Фронтенд состоит из двух отдельно запускаемых приложений: Пользовательской и Администраторской.

2.2.1 Запуск фронтенда на одном узле (локальной машине)

Для запуска фронтенда нужно:

- > Установить Nodejs версии LTS
- > Прописать в конфигурационных файлах `env-config.js` нужные переменные окружения
- > Установить зависимости - `$ npm install`



- › Выполнить сборку приложения \$ npm run build
- › Запустить приложение \$ npm run start

2.2.2 Запуск фронтенда в условиях контейнеризации или на отдельных VPS

Фронтенд поддерживает контейнеризацию и может работать под балансировщиком нагрузки. Благодаря тому, что пользовательская часть отделена от администраторской

2.2 Пользователь по умолчанию

После выполнения миграций в системе будет создан суперпользователь с именем `admin@local.host` и паролем `inigmo2` для управления пользователями и настройки предустановленных типов приложений через веб-интерфейс администратора.

Информация о персонале

3.1 Персонал, обеспечивающий работу Системы на местах

Пользователи Системы должны обладать навыками работы с персональным компьютером на уровне пользователя и опыт использования веб-браузера.

Для работы с Системой пользователю необходимо изучить руководство пользователя размещенное на первой странице Деплойми

Администратор Системы должен владеть навыками работы с персональным компьютером на уровне уверенного пользователя. Желательно знание основ работы вычислительной техники и программного обеспечения в локальных сетях, опыт администрирования веб-серверов на *nix-образных ОС.

3.1 Персонал, обеспечивающий техническую поддержку и модернизацию

Специалисты, обеспечивающие техническую поддержку и развитие Системы, должны обладать следующими знаниями и навыками:

- › владение персональным компьютером на уровне специалиста;
- › знание функциональных возможностей Системы и особенностей работы с ними;
- › знания технологий и языков: Javascript, TypeScript, SQL.

Специалисты, осуществляющие модернизацию Системы, помимо вышеперечисленного, должны иметь квалификацию инженер-программист.



Архитектура Деплойми

