Telegram bot «Boostra concierge»
Руководство по установке и развертыванию ПО

1. Введение

Настоящее руководство описывает процесс установки программного обеспечения «Boostra concierge».

2. Используемые зависимости

Ruby 3.1.4;
PostgreSQL 14;
Redis 6+;
Chromium.
Разработка через Docker.
Для сборки PDF из HTML используется gem grover. Под капотом используется puppeteer.

3. Требования к программному обеспечению серверной части

Требуемый доступ к серверу: SSH;
Серверная платформа Debian;
Веб-сервер Nginx 1.21;
СУБД PostgreSQL 13.3.

4. Требования к хостингу

Процессор: 4 core или выше.
Оперативная память: 4Gb RAM или выше.
Объем дискового пространства: 60Gb или выше.
Виртуальный выделенный сервер или физический сервер.

5. Требования к панели администратора и telegram-боту

Для использования программного обеспечения пользователь должен иметь постоянный доступ к сети Интернет, любой современный браузер и клиент telegram. Оборудование пользователя должно соответствовать рекомендуемым требованиям для функционирования браузера, через который пользователь использует программное обеспечение. Для использования программного обеспечения производитель рекомендует пользователю использовать следующие браузеры:
Google Chrome 87.0 и выше;
Mozilla Firefox 84.0 и выше;
Safari 14.0;
Opera 72.0 и выше.
Программное обеспечение распространяется в виде Telegram - бота.
Программное обеспечение предоставляется пользователю в виде готового к работе сервиса, пользователь не производит самостоятельную установку и

настройку программного обеспечения, а работает в уже настроенном и готовом к работе программном обеспечении, развернутом на оборудовании производителя. После завершения процедуры регистрации пользователь получает возможность использовать программное обеспечение в соответствии с его функциональным назначением. Для получения доступа к продукту пользователю достаточно перейти на веб-сайт https://b-concierge.online используя любой из браузеров, предложенных в п. 2., и оставить заявку на подключение к боту.
Тестовые доступы для экспертной проверки

Telegram-бот
https://t.me/chatgpt_fintech_1_bot

№	Шаг	Критерий готовности
1	Подготовка данных и конфигурирование	1. Сгенерированы логин/пароль для подключения приложений к инфраструктурным компонентам (дополнить списком пользователей) • Mongo - для каждой БД (9) • Postgres - для каждой БД (6) • S3 - AC/SC • Repo - пользователь для передачи в image pool secret (Kuberenetes) 2. Подготовлен S3 • создать bucket (уточнить количество и настройки) 3. Подготовлены Kafka/Zookeeper • создать топики • сконфигурировать политики • Zoo *extended types enabled* 4. Созданы БД • Postgres - liquibase • Mongo - mongock 5. Создано и проверено подключение к Identity Manager (LDAP) • логин/пароль для подключения, получить и прописать

2	Верификация площадки	1. Предоставлен доступ к площадке для исполнителей 2. Проверена сетевая связанность всех инфраструктурных компонентов • На АРМ сетевая доступность и УЗ/токен для подключения к KubeCTL(OC) ○ права на создание объектов ▪ namespace ▪ config map ▪ deployment ▪ deployment config ▪ secrets ▪ services ▪ ingress ▪ routs • В Kuberenetes (наличие репозитория и его доступность) ○ image pool secret (настроен на репозиторий дистрибутивов) 3. Создан проект OpenShift 4. OpenShift должен выставлять в подсеть заказчика адреса front-end (code/graph/prometheus) 5. Проверена доступность компонент на разных узлах инфраструктуры • Mongo • Postgres

№	Шаг	Критерий готовности
		• Elastic • Kafka • Zookeeper • S3 • Prometheus Проверка УЗ администраторов для настройки инфраструктурных компонентов
3	Репо наполняется дистрибутивами	1. OpenShift получает конфигурации и логин/пароль (в соответствие с порядком установки) 2. OpenShift забирает дистрибутивы из репозитория, по конфигурациям создает поды и настраивает их по конфигурациям
4	Порядок установки служб и сервисов соответствует порядку их запуска	1. Выполнены работы согласно таблицам запуска сервисов
5	Верификация инсталляции	1. Smoke test успешно пройден

Дополнительные требования для установки

1. Для успешной локальной установки необходимо скачать официальную версию Docker Desktop для локального развертывания системы. Инструкция по установке программы https://www.docker.com/products/docker-desktop

Формирование файла и каталога compose

1. Создать каталог compose и перейти в него

2. Внутри каталога compose создать файл docker-compose.yml со следующим содержанием:

```
version: '3'
services: # App
sbb-nlu:
image: sbb-nlu:01.003.00
mem_limit: 2048m restart: on-
failure environment:
KAFKA_CONSUMER_BOOTSTRAP_SERVERS: 'kafka:9092'
KAFKA_PRODUCER_BOOTSTRAP_SERVERS: 'kafka:9092'
REDIS_HOST: 'redis'
# Infrastructure redis:
image: redis:latest ports: - 6379:6379
healthcheck: test: ["CMD", "redis-cli", "ping"]
interval: 5s timeout: 30s retries: 50
restart: unless-stopped
redis-commander:
image:
rediscommander/redis-
commander:latest
environment:
- REDIS_HOSTS=local:redis:6379 ports:
- "8000:8081"
zookeeper:
image: confluentinc/cp-zookeeper:latest environment:
ZOOKEEPER_CLIENT_PORT: 2181
ZOOKEEPER_TICK_TIME: 2000 ports:
- 2181:2181
kafka:
image: confluentinc/cp-kafka:latest
depends_on: - zookeeper ports: -
29092:29092 environment:
KAFKA_BROKER_ID: 1
KAFKA_ZOOKEEPER_CONNECT: zookeeper:2181
KAFKA_ADVERTISED_LISTENERS:
PLAINTEXT://localhost:29092,PLAINTEXT_HOST://kafka:9092
KAFKA_LISTENER_SECURITY_PROTOCOL_MAP:
PLAINTEXT:PLAINTEXT,PLAINTEXT_HOST:PLAINTEXT
KAFKA_INTER_BROKER_LISTENER_NAME: PLAINTEXT
KAFKA_OFFSETS_TOPIC_REPLICATION_FACTOR: 1
```

Загрузка образа sbb-nlu из дистрибутива

1. Скопировать файл sbb-nlu.tar из дистрибутива в созданный каталог compose 2. Открыть командную строку и перейти в каталог compose 3. Выполнить команду:
docker load -i sbb-nlu.tar
Запуск программы
1. Открыть командную строку
2. Перейти в каталог compose в режиме командной строки: cd <путь до каталога>/compose
3. Ввести команду: docker-compose up
4. Начнется скачивание образов с Docker Hub
5. После развертывания образов система будет готова к работе 6. Далее см. руководство пользователя