Compo B2B Platform

Руководство по развертыванию

Компо софт. Москва 2023

Подготовка инфраструктуры для тестового или рабочего окружения. Пример.

Для развертывания Compo B2B Platform необходим сервер под управлением Linux-совместимой операционной системы. Рекомендуется Debian или Astra Linux последних стабильных версий.

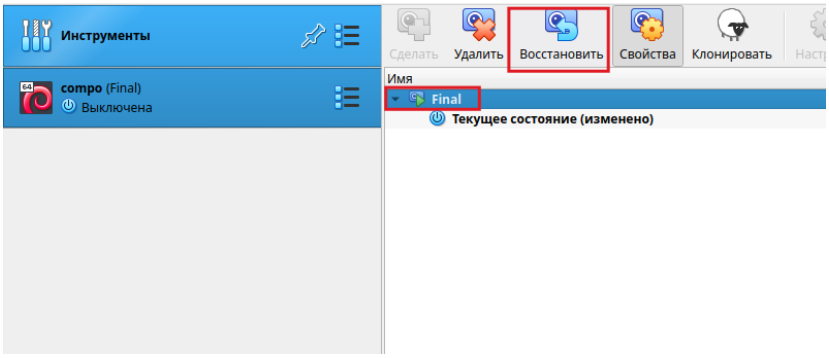
Продуктивный сервер (минимальные требования):
CPU: от 8 ядер (рекомендуется 16)
RAM: от 32 Gb (рекомендуется 64)
SSD: от 200 gb (рекомендуется 1Тб)

Требуемое программное обеспечение продуктивного сервера:
СУБД: PostgreSQL / PostgresPro (не ниже 12.9), ElasticSearch 7.15
Проксирование: Nginx
JVM: OpenJDK 17.0.5
Опционально: Kibana 7.15
Брокер сообщений: Rabbitmq (не ниже 3.8.2)
systemd сервисы: Compo B2B Platform, Compo CLI, Compo Integration BUS, Compo Administration System

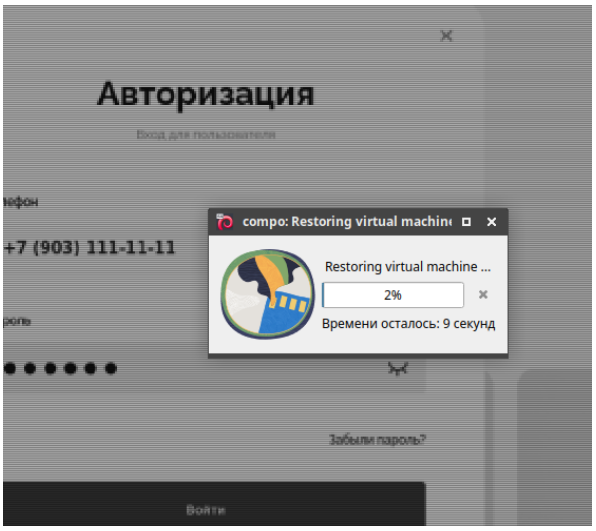
Описание экземпляра ПО

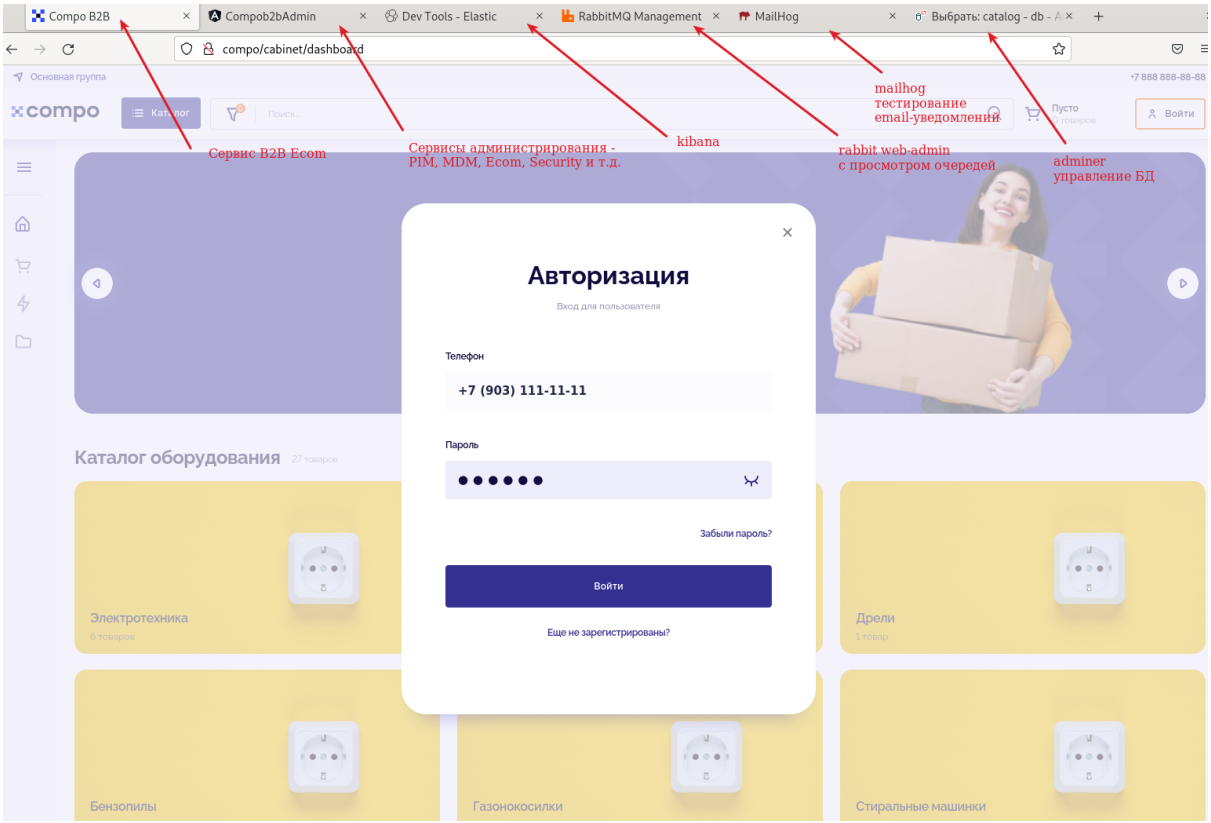
Для удобства экземпляр программного обеспечения поставляется в виде снимка для виртуальной машины Oracle VM VirtualBox 7.0.6

Необходимо скачать снимок “Final” и восстановить состояние по этому снимку. Машина настроена на базе ОС Debian 11.

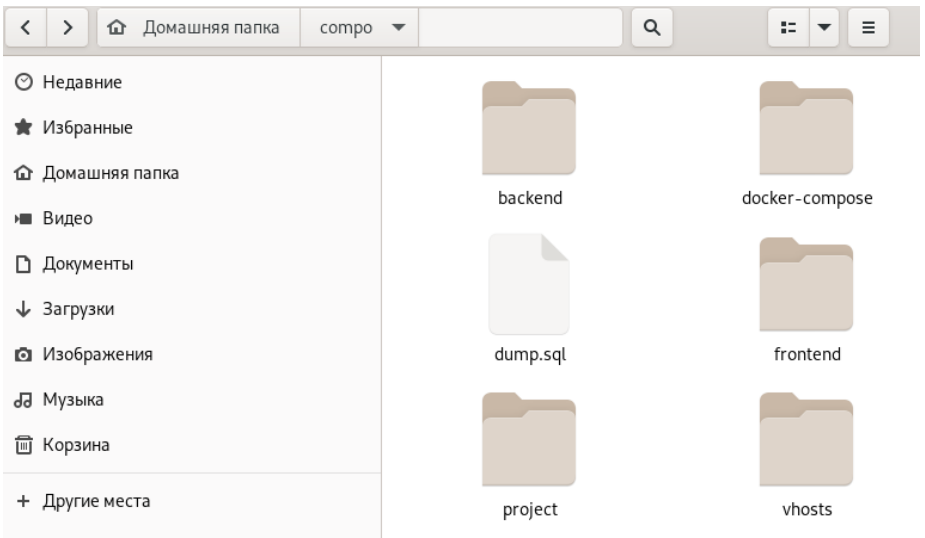


После восстановления и запуска снимка в виртуальной машине можно будет увидеть открытое окно браузера с вкладками работы и управления как платформы так и инфраструктурного программного обеспечения.





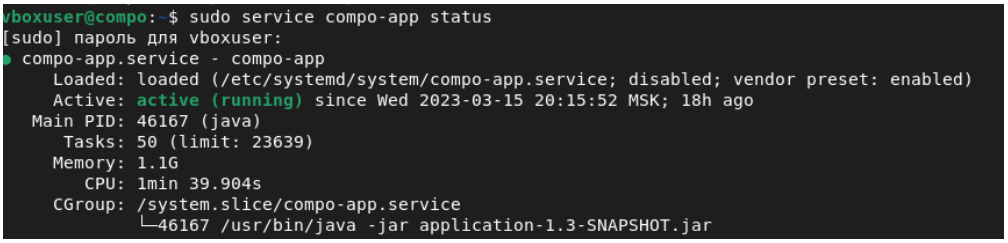
Также будет открыт терминал в рабочей дирректории и сама рабочая директория будет открыта в файловом менеджере.



Описание компонентов платформы Compo B2B Platform

Платформа представляет из себя 3 сервиса, сконфигурированных в systemd:

- compo-admin.service - Java-сервис(spring boot fat jar) для обеспечения работы бэкенда системы администрирования и ее подсистем. Порт 8086
- compo-app.service - Java-сервис(spring boot fat jar) B2B Ecommerce платформы и ее подсистем. Порт 8080
- compo-integration.service - Java-сервис(spring boot fat jar) интеграционной шины Compo



Также помимо веб-сервисов доступно инфраструктурное java(spring boot fat jar) cli-приложение Compo, которое запускается из терминала с помощью команды compo-cli

Для обеспечения работы фронтенда предоставлено две сборки angular-приложения для B2B Ecom платформы и системы администрирования. Находящихся по адресам:

- /home/vboxuser/compo/frontend/web-admin
- /home/vboxuser/compo/frontend/web-app

Для обеспечения инфраструктурного окружения подняты контейнеры с PostgreSql, rabbitMq, mailhog, elasticsearch, kibana - с помощью docker-compose файла находящегося по адресу:

- /home/vboxuser/compo/docker-compose/stack.yml
- команда поднятия(уже выполнена) sudo docker-compose -f stack.yml up –detach
- команда закрытия контейнеров при необходимости sudo docker-compose -f stack.yml down

Описание рабочей директории /home/vboxuser/compo

- backend - в директории сложены исполняемые java сервисы и логи
- frontend - JS-сборки ангулар-приложений
- docker-compose - конфигурация для контейнеров и папка .docker с voluma-ми и проекцией дополнительных файлов из контейнеров
- vhosts - конфигурация nginx
- dump.sql - дамп БД
- project - физическое хранилище файлов (фото, pdf и т.д.)

Описание базовых конфигураций сервисов

/etc/systemd/system

Конфигурация compo-app.service:

```
[Unit]
Description=compo-app

[Service]
WorkingDirectory=/home/vboxuser/compo/backend/application
ExecStart=/usr/bin/java -jar application-1.3-SNAPSHOT.jar
#Restart=always
#RestartSec=10
SyslogIdentifier=compo-app
User=vboxuser

[Install]
WantedBy=multi-user.target
```

Конфигурация compo-admin.service:

```
[Unit]
Description=compo-admin

[Service]
WorkingDirectory=/home/vboxuser/compo/backend/administration
ExecStart=/usr/bin/java -jar administration-1.3-SNAPSHOT.jar
Restart=always
RestartSec=10
SyslogIdentifier=compo-admin
User=vboxuser

[Install]
WantedBy=multi-user.target
```

Конфигурация compo-integration.service:

```
[Unit]
Description=compo-integration

[Service]
WorkingDirectory=/home/vboxuser/compo/backend/integration
ExecStart=/usr/bin/java -jar integration-app-1.3-SNAPSHOT.jar
#Restart=always
#RestartSec=10
SyslogIdentifier=compo-integration
User=vboxuser

[Install]
WantedBy=multi-user.target
```

Конфигурация контейнеров:

```
# Use postgres/test user/password credentials
version: '3.1'

services:

  db:
    image: postgres
    restart: always
    environment:
      POSTGRES_PASSWORD: example
      PGDATA: /var/lib/postgresql/data/pgdata
    ports:
      - 5432:5432
    volumes:
      - data00:/var/lib/postgresql/data
  adminer:
    image: adminer
    restart: always
    ports:
      - 8081:8080

  mailhog:
    image: mailhog/mailhog:latest
    restart: always
    ports:
      - 1025:1025
      - 8025:8025

  elasticsearch:
    image: docker.elastic.co/elasticsearch/elasticsearch:7.15.0
    container_name: elasticsearch
    environment:
      - node.name=elasticsearch
      - cluster.name=es-docker-cluster
      - discovery.seed_hosts=es02,es03
      - cluster.initial_master_nodes=elasticsearch
      - bootstrap.memory_lock=true
      - "ES_JAVA_OPTS=-Xms2048m -Xmx2048m"
    ulimits:
      memlock:
        soft: -1
        hard: -1
    volumes:
      - data01:/usr/share/elasticsearch/data
    ports:
      - 9200:9200
    networks:
      - elastic

  logstash:
    image: docker.elastic.co/logstash/logstash:7.15.0
    ports:
      - "5044:5044"
      - "5000:5000/tcp"
      - "5000:5000/udp"
      - "9600:9600"
    environment:
```

```
      LS_JAVA_OPTS: -Xms256m -Xmx256m
    networks:
      - elastic
    depends_on:
      - elasticsearch

  kibana:
      image: docker.elastic.co/kibana/kibana:7.15.0
      container_name: kibana
      ports:
        - 5601:5601
      environment:
        ELASTICSEARCH_URL: http://elasticsearch:9200
        ELASTICSEARCH_HOSTS: '["http://elasticsearch:9200"]'
      networks:
        - elastic
  rabbitmq:
    image: rabbitmq:3-management
    container_name: rabbitmq
    hostname: "rabbit"
    volumes:
      - ./.docker/rabbitmq/data/:/var/lib/rabbitmq/
      - ./.docker/rabbitmq/logs/:/var/log/rabbitmq/
    environment:
      RABBITMQ_ERLANG_COOKIE: erlang_cookie
      RABBITMQ_DEFAULT_USER: guest
      RABBITMQ_DEFAULT_PASS: guest
    ports:
      - 5672:5672
      - 15672:15672
    networks:
      - rabbit
volumes:
  data00:
    driver: local
  data01:
    driver: local


networks:
  elastic:
    driver: bridge
  rabbit:
    driver: bridge
```

Ручной запуск сервисов

Мы прилагаем архив compo-platform.zip - который позволит вручную запустить все системы при необходимости, следуя инструкциям ниже.

Для того чтобы запустить платформу на произвольной машине. ПО машины должно соответствовать требованиям приведенным в начале этого документа.

Необходимо развернуть инфраструктурное окружение. Это можно сделать с помощью docker-compose файла stack.yml описанного в данной инструкции в разделе **Конфигурация контейнеров.**

Так же нужно сконфигурировать локальный nginx или apache с проксированием запросов. Пример для nginx можно найти в данном документе, в разделе **Конфигурация проксирования nginx.**

В запущенную базу данных необходимо загрузить дамп из dump.sql.
Это можно сделать с помощью клиента psql, пример:

```
psql -U postgres -h 127.0.0.1 -d mydbname < dump.sql
```

Далее необходимо сконфигурировать сервисы. Рядом с каждым jar-файлом сервиса лежит его application.properties файл, в котором прописываются учетные данные для доступа к БД, rabbit, smtp, elasticsearch и т.д. Необходимо запустить cli приложение из папки cli. Это можно сделать с помощью команды:

```
java -jar cli-app-1.3-SNAPSHOT.jar
```

После запуска консольного приложения необходимо выполнить команды refresh и refresh-catalog

Пример:

```
compo-cli>refresh
```

Команда refresh - проиндексирует содержимое базы данных в elasticsearch а refresh-catalog - пересчитает ключи и количество товаров в категориях для дерева каталога.

Далее можно запустить остальные сервисы:

```
java -jar administration-1.3-SNAPSHOT.jar
java -jar application-1.3-SNAPSHOT.jar
java -jar integration-1.3-SNAPSHOT.jar
```

При этом nginx сконфигурировать надо будет для локальных доменов compo и compo-admin - т.к. текущие сборки frontend настроены на работы с этими доменами.

Учетные данные для доступа к тестовой системе и ее инфраструктуре

Linux пользователи
логин: vmboxuser
пароль: Compo$23engine

логин: root
пароль: Compo$23engine

Compo B2B Ecommerce
http://compo
логин: 79031111111
пароль: 123456

Compo PIM/MDM
http://compo-admin
логин: test_manager
пароль: 123456

Kibana
http://localhost:5601

Rabbit web admin
http://localhost:15672
логин: guest
пароль: guest

Adminer
http://localhost:8081
логин: postgres
пароль: example