|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  |  | | --- | --- | --- | |  |  |  | | МИНОБРНАУКИ РОССИИ | | | | Федеральное государственное бюджетное образовательное учреждение  высшего образования  **«МИРЭА – Российский технологический университет»**  **РТУ МИРЭА** | | |   Институт Информационных технологий | |
|  | |
| Кафедра Инструментального и прикладного программного обеспечения (ИиППО) | |
|  | |
|  | |

|  |  |
| --- | --- |
| **ОТЧЕТ**  **ПО ПРАКТИЧЕСКОЙ РАБОТЕ № 8**  «Развертывание docker-контейнеров на cloud-сервере» | |
| **по дисциплине** | |
| **«**Настройка и администрирование сервисного программного обеспечения**»** | |
|  | |
| Выполнил студент группы ИКБО-20-21 | Сидоров С. Д. |
| Принял преподаватель | Хозяинов А.В. |

|  |  |  |
| --- | --- | --- |
| Лабораторная работа выполнена | «\_\_»\_\_\_\_\_\_\_202\_ г. | *(подпись студента)* |
|  |  |  |
| «Зачтено» | «\_\_»\_\_\_\_\_\_\_202\_ г. | *(подпись руководителя)* |

Москва 2023

Цель работы

Облачный хостинг предназначен для размещения сайтов, приложений и других сервисов для доступа к этим размещенным ресурсам через сети передачи данных.

Наиболее известные cloud хостинги GearHost, Amazon Web Services (AWS), Google Cloud, Heroku, Microsoft Azure.

Каждый облачный хостинг имеет интерфейс управления.

Рекомендуемый cloud хостинг для выполнения заданий практических работ - Heroku.Теоретическое введение

Выполнение работы

Развернуть контейнеры на хостинге. Развернутую архитектуру на хостинге объяснить и продемонстрировать преподавателю. Предполагается, что следующие несколько практических работ будут ориентированы на работу с cloud хостингом.

На cloud хостинге в своем аккаунте выполнить следующие задания:

Задание 1: развернуть docker-контейнер с Jenkins на cloud-хостинге. Объяснить, для чего предназначен Jenkins. Продемонстрировать пример применения Jenkins-контейнера на cloud-хостинге. Проект назвать docker\_jenkins. Результаты представлены на листинге 1 и рисунках 1, 2.

Листинг 1 – Docker-compose.yml для Jenkins

|  |
| --- |
| services:  jenkins:  image: jenkins/jenkins  ports:  - "8080:8080"  - "50000:50000" |



Рисунок 1 – Запуск контейнера в фоновом режиме

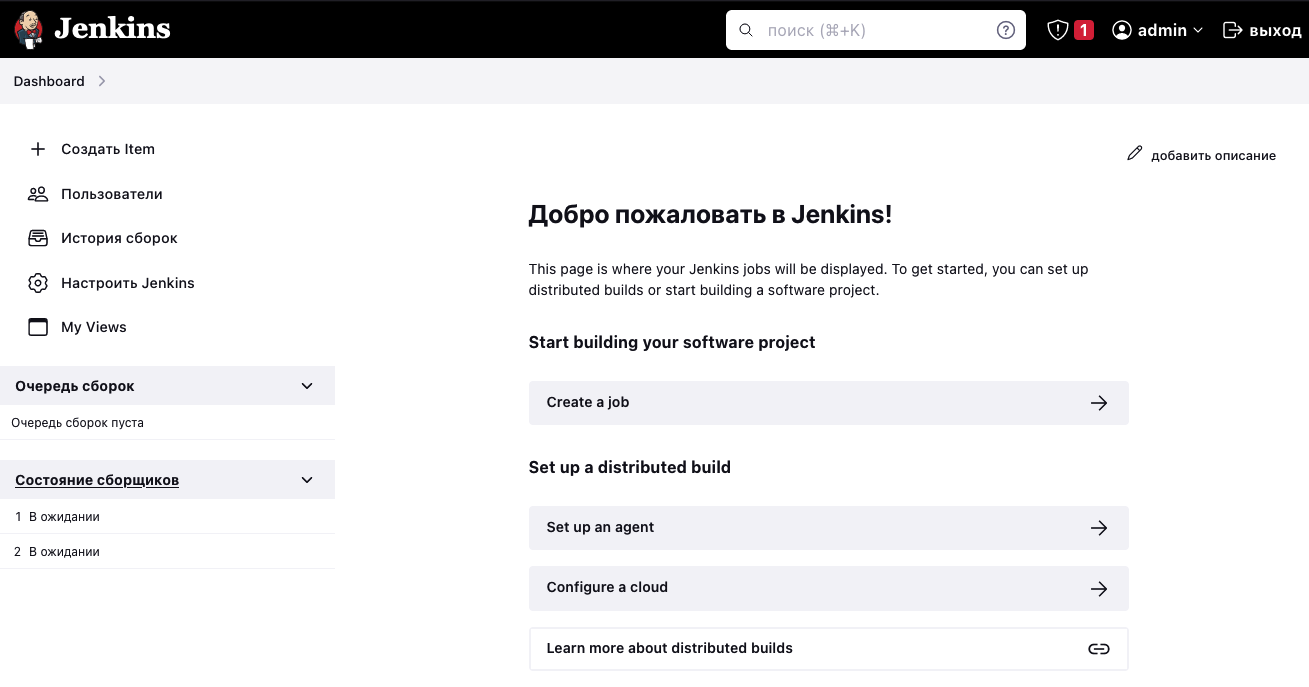


Рисунок 2 – Страница Jenkins

Задание 2: развернуть docker-контейнер с Maven на cloud-хостинге. Объяснить, для чего предназначен Maven. Продемонстрировать пример применения Maven-контейнера на cloud-хостинге. Проект назвать docker\_maven. Результат показан на листинге 2 и рисунках 3, 4.

Листинг 2 – Dockerfile для docker\_maven

|  |
| --- |
| FROM maven:3.8.5-openjdk-17-slim AS build WORKDIR /app COPY src /app/src COPY pom.xml /app RUN mvn -f /app/pom.xml clean package  FROM openjdk:17-jdk-slim COPY --from=build /app/target/maven\_docker-0.0.1-SNAPSHOT.jar /app/my-app.jar EXPOSE 8081 ENTRYPOINT ["java", "-jar", "/app/my-app.jar"] |

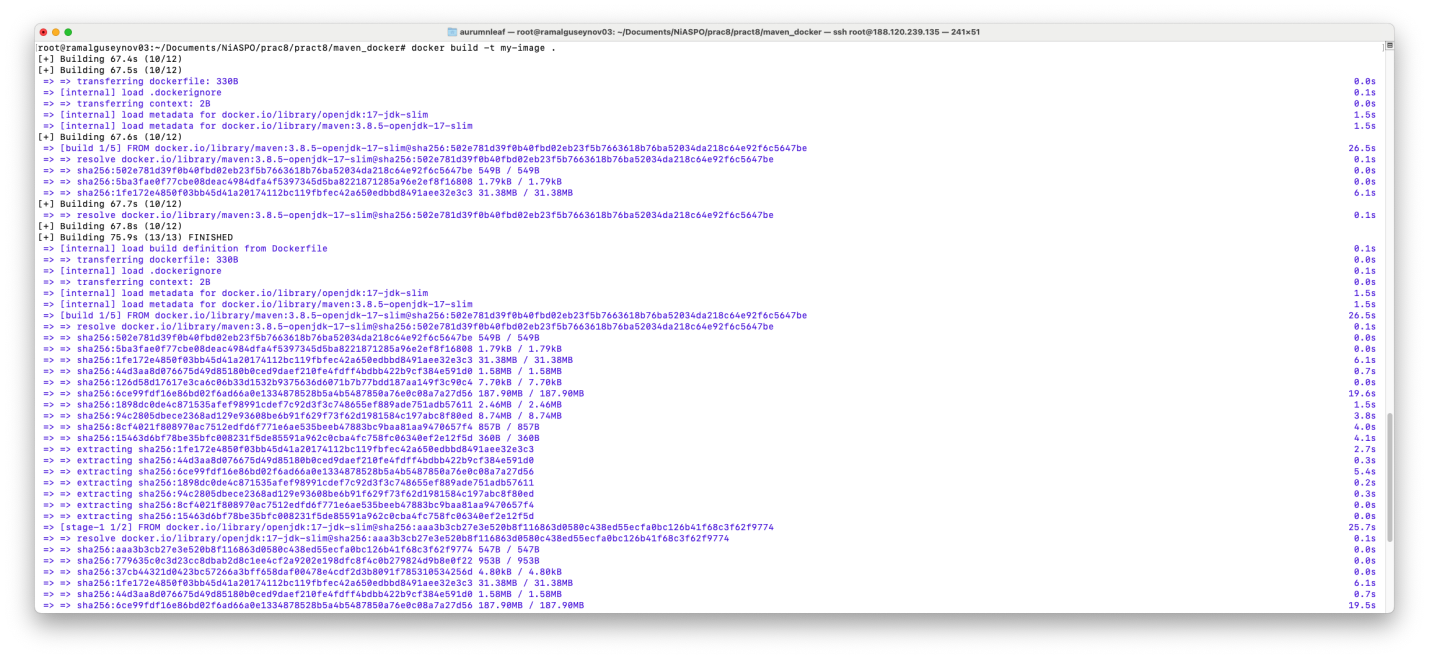


Рисунок 3 – Сборка docker\_maven

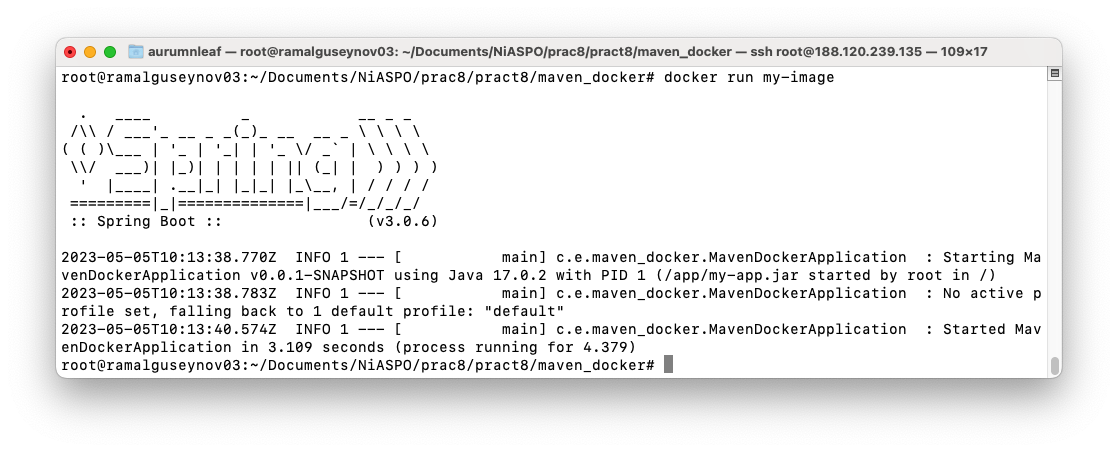


Рисунок 4 – Запуск docker-maven

Задание 3: развернуть docker-контейнер с CMS Drupal на cloud-хостинге. Объяснить, для чего предназначен Drupal. Продемонстрировать пример применения Drupal-контейнера на cloud-хостинге. Проект назвать docker\_drupal. Результат представлен на листинге 6 и рисунке 3.

Листинг 3 – Dockerfile для приложения docker\_drupal

|  |
| --- |
| version: '3.3'  services:  drupal:  image: drupal:latest  ports:  - 81:80 |

Продолжение листинга 3

|  |
| --- |
| volumes:  - drupal\_modules:/var/www/html/modules  - drupal\_profiles:/var/www/html/profiles  - drupal\_themes:/var/www/html/themes  - drupal\_sites:/var/www/html/sites  restart: always  db:  image: mysql  command: --default-authentication-plugin=mysql\_native\_password  restart: always  environment:  MYSQL\_ROOT\_PASSWORD: 1234  volumes:  drupal\_modules:  drupal\_profiles:  drupal\_themes:  drupal\_sites: |

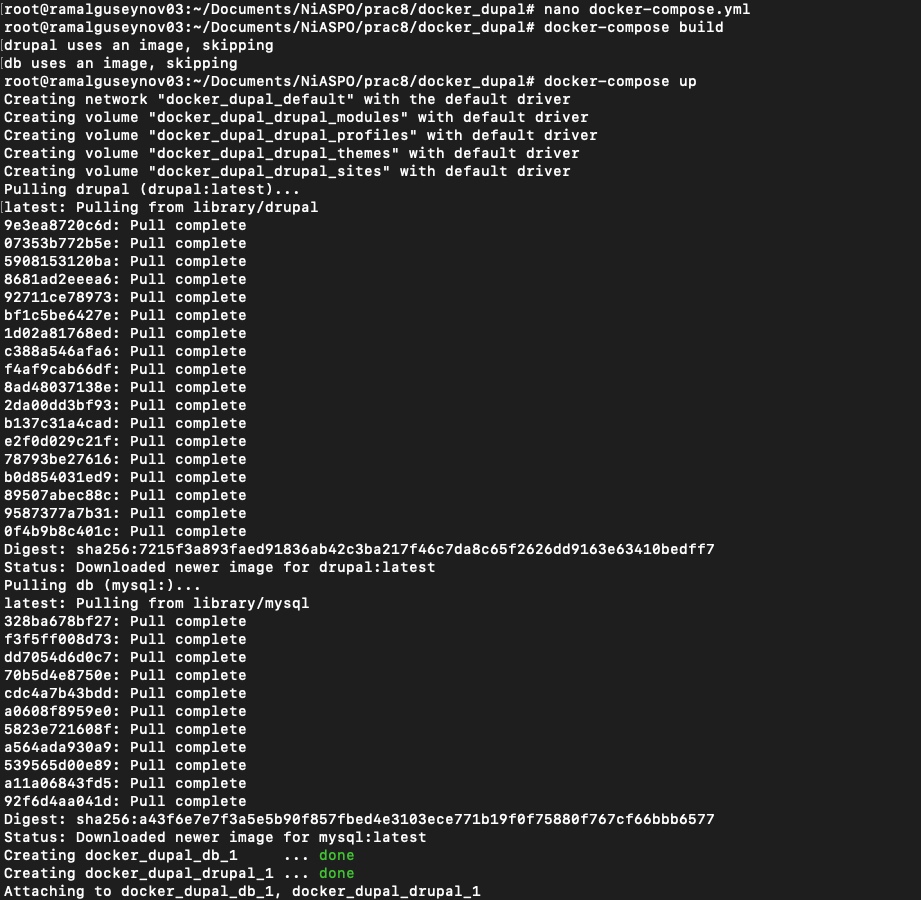


Рисунок 5 – Запуск docker\_drupal

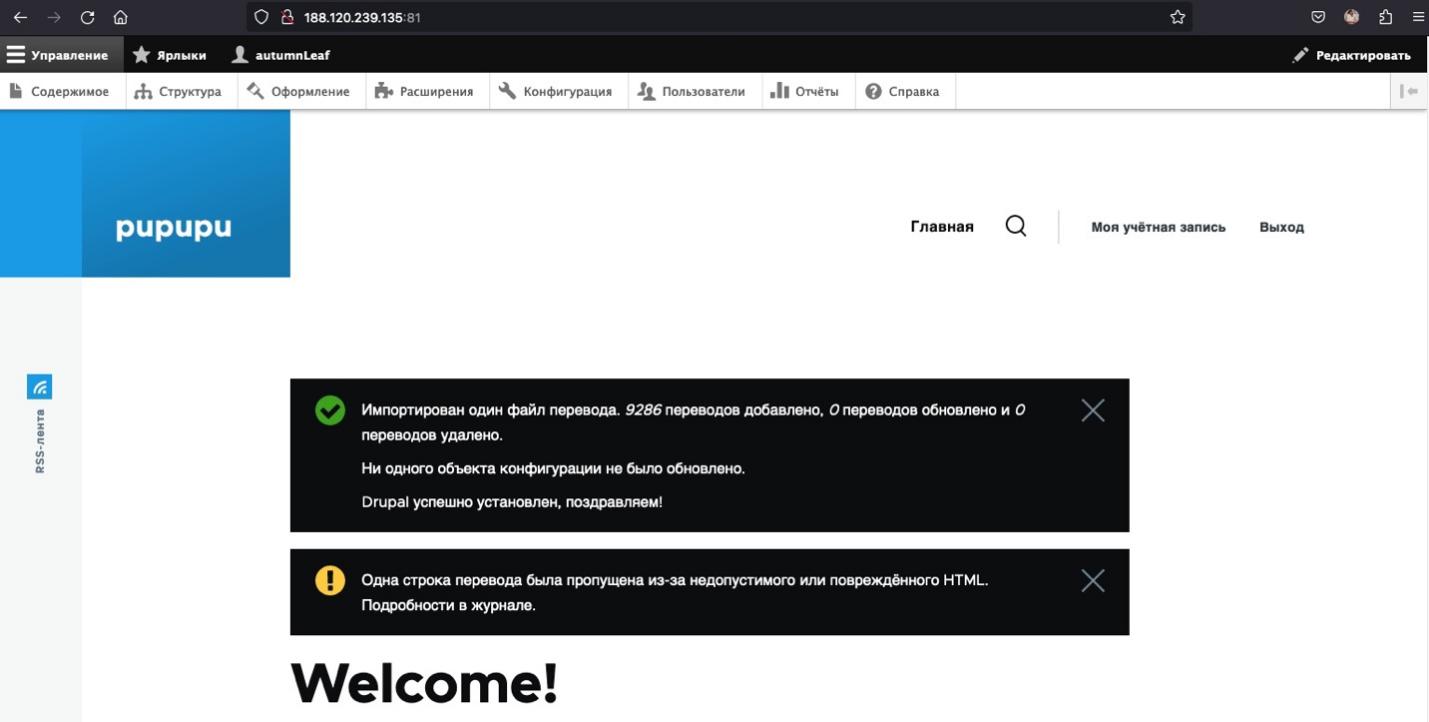


Рисунок 6 – Веб интерфейс docker\_drupal

Задание 4: развернуть docker-контейнер с СУБД Couchbase на cloud хостинге. Объяснить, для чего предназначен Couchbase. Продемонстрировать пример применения Couchbase контейнера на cloud-хостинге. Проект назвать docker\_couchbase. Результат представлен на листинге 4 и рисунках.

Листинг 4 – docker-compose для couchbase

|  |
| --- |
| version: '3'  services:  couchbase:  image: couchbase:latest  ports:  - "8091-8094:8091-8094"  - "11210:11210"  environment:  - COUCHBASE\_CLUSTER\_RAMSIZE=512  - COUCHBASE\_USERNAME=admin  - COUCHBASE\_PASSWORD=password |



Рисунок 7 – Веб интерфейс стартовой страницы couchbase

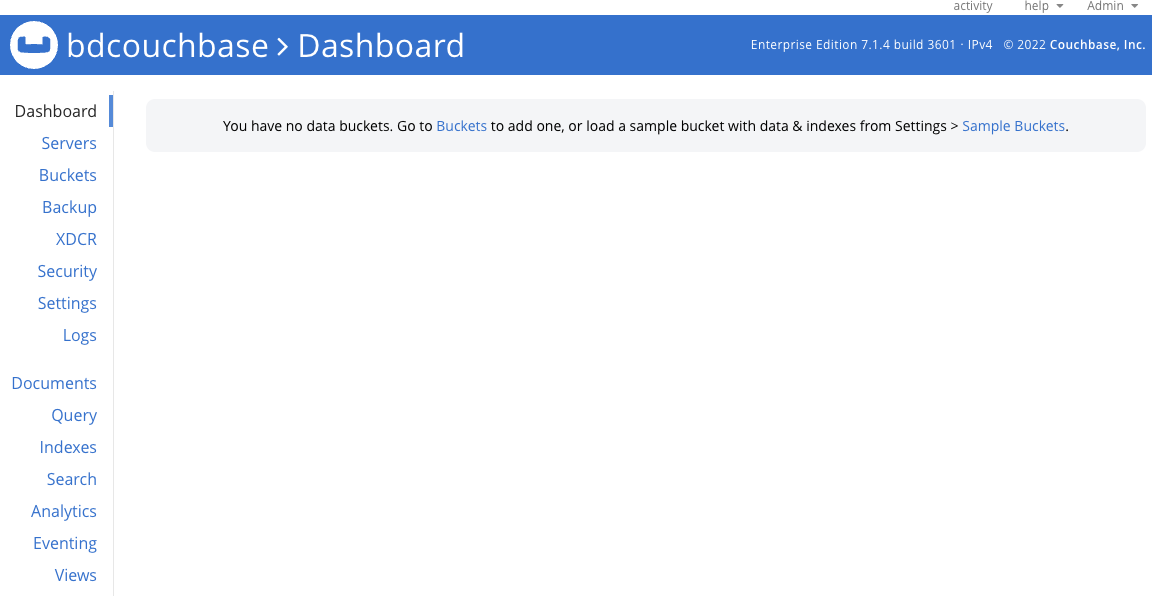


Рисунок 8 – Веб интерфейс администратора

Задание 5: развернуть docker-контейнер с Node.js на cloud-хостинге. Объяснить, для чего предназначен Node.js. Продемонстрировать пример применения Node.js-контейнера на cloud-хостинге. Проект назвать docker\_nodejs. Было использовано приложение из практической работы №7. Результат представлен на рисунках 9 - 10.

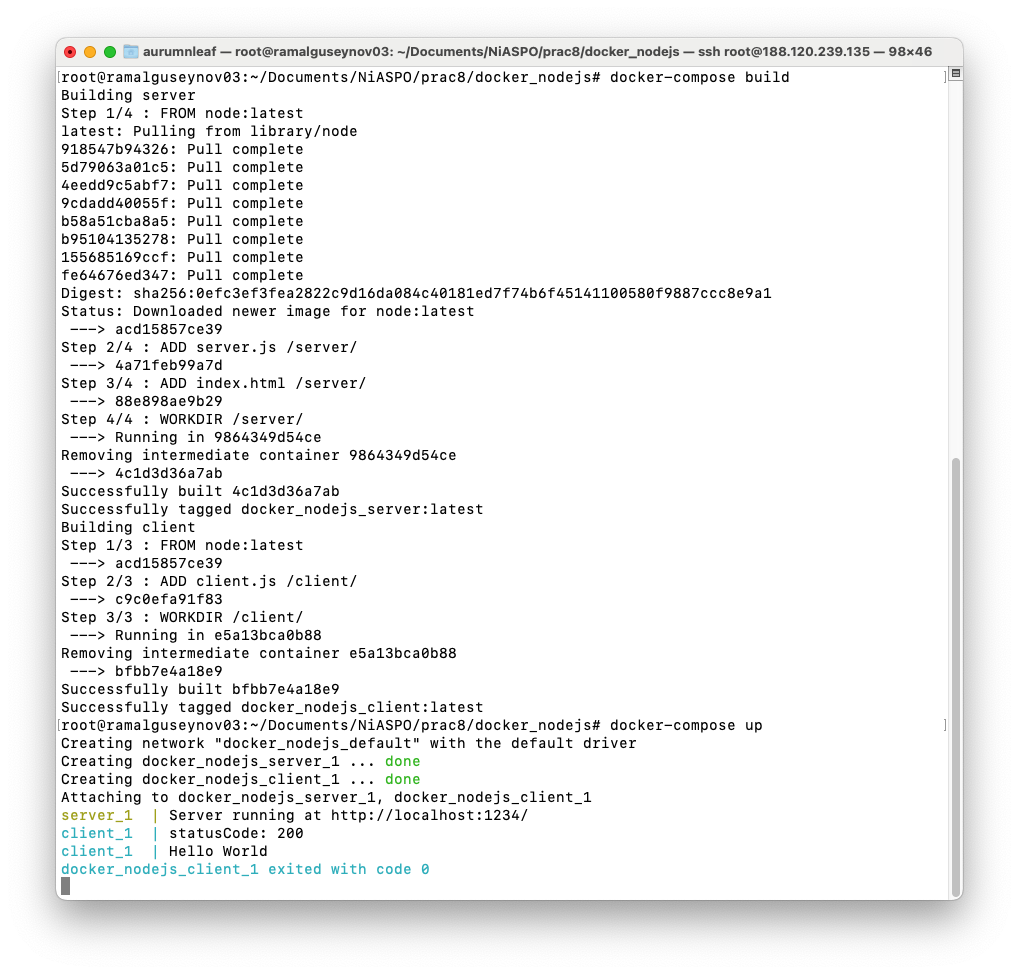


Рисунок 9 – Сборка docker\_nodejs

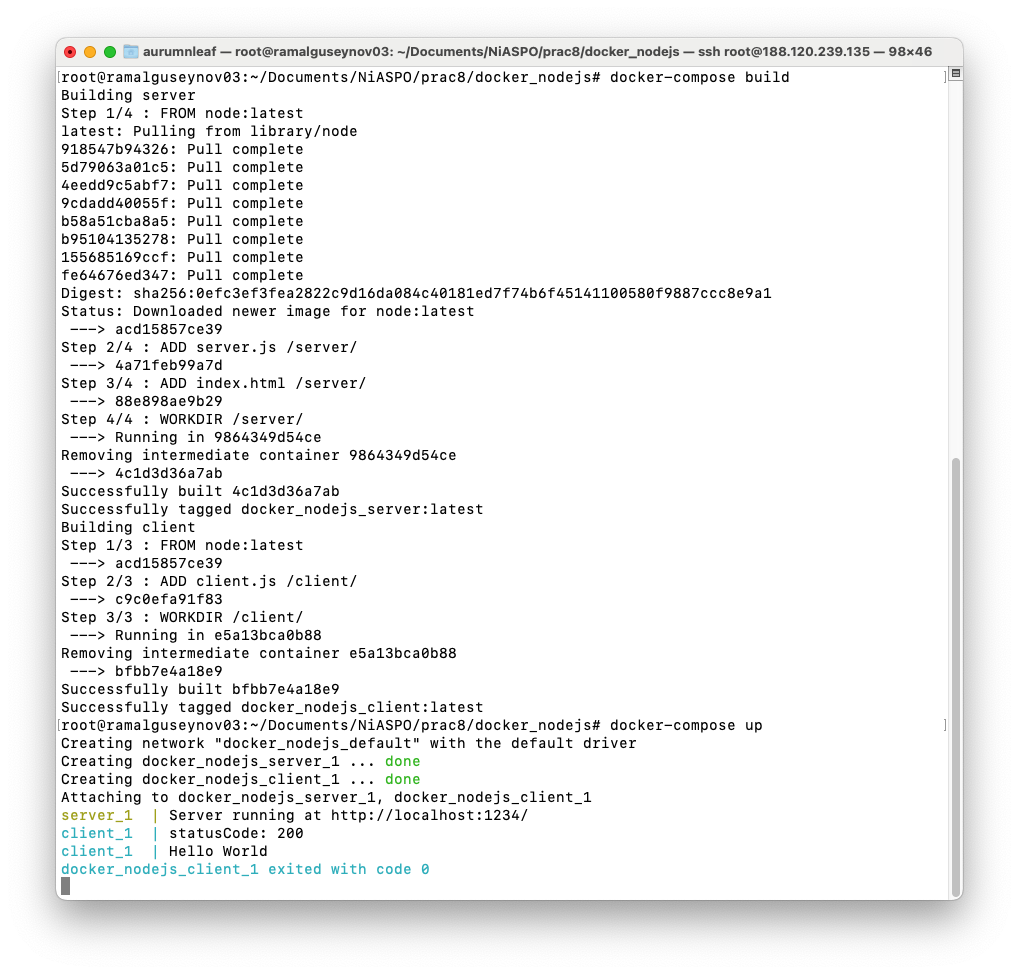


Рисунок 10 - запуск docker\_nodejs

**ВЫВОД**

В ходе данной работы был получен опыт создания докер контейнеров на cloud хостинге, были изучены docker-compose и Dockerfile.