Министерство науки и высшего образования Российской Федерации

ФГАОУ ВО «Уральский федеральный университет   
имени первого Президента России Б.Н. Ельцина»

Институт новых материалов и технологий

Кафедра «Теплофизика и информатика в металлургии»

**Автоматизация сборки проекта с помощью Jenkins**

**ОТЧЕТ**

**по практической работе № 4**

**по дисциплине «Основы методологии Development Operation»**

Направление 09.03.02 «Информационные системы и технологии» (уровень бакалавриата)

Образовательная программа  
09.03.02/33.02 «Информационные системы и технологии» (СУОС)

Студент

группы НМТ-413901 И.В.Машуков

Преподаватель:

профессор, д.т.н. В.В.Лавров

Екатеринбург

2024

СОДЕРЖАНИЕ

[1.1 Цель работы 3](#_Toc185529509)

[1.2 Ход проведения работы 3](#_Toc185529510)

[1.2.1 Разработка тестового приложения Visual Studio .NET Core 3](#_Toc185529511)

[1.2.2 Создание контейнера с использованием Docker 3](#_Toc185529512)

[1.2.3 Создание файла docker-compose.yml 3](#_Toc185529513)

[1.2.4 Создание файла Jenkinsfile 3](#_Toc185529514)

[1.2.5 Размещение приложения в системе контроля версий GitHub 4](#_Toc185529515)

[1.2.6 Настройка автоматической сборки проекта в Jenkins и интеграции с GitHub 4](#_Toc185529516)

[1.2.7 Выполнение автоматической сборки проекта в Jenkins и анализ результатов 4](#_Toc185529517)

[1.2.8 Демонстрация автоматической сборки и доставки проекта при корректировке функциональности веб-приложения 5](#_Toc185529518)

[1.3 Выводы 7](#_Toc185529519)

[Приложение А Листинг программного кода приложения «Калькулятор» (контроллер) 9](#_Toc185529520)

[Приложение Б Листинг программного кода приложения «Калькулятор» (appsettings.json) 10](#_Toc185529521)

[Приложение В Листинг программного кода приложения «Калькулятор» (appsettings.Development.json) 11](#_Toc185529522)

[Приложение Г Листинг программного кода приложения «Калькулятор» (Program.cs) **Ошибка! Закладка не определена.**](#_Toc185529523)

[Приложение Д Листинг файла Dockerfile **Ошибка! Закладка не определена.**](#_Toc185529524)

[Приложение Е Листинг файла docker-compose.yml **Ошибка! Закладка не определена.**](#_Toc185529525)

[Приложение Ж Листинг файла Jenkinsfile 12](#_Toc185529526)

# **Практическая работа «Автоматизация сборки проекта с помощью Jenkins»**

1.1 Цель работы

Цель данной работы – научиться пользоваться процессом непрерывной интеграции программного обеспечения на Jenkins.

1.2 Ход проведения работы

### 1.2.1 Разработка тестового приложения Visual Studio .NET Core

Вернем приложение “11\_Calculator” к моменту, когда была завершена практическая работа №1, для этого просто загрузим соответствующий коммит из ветки master.

### 1.2.2 Создание контейнера с использованием Docker

Создание контейнера в данной работе осуществляется автоматически через docker-compose.yml, настройки которого будут выполнены во время сборки через Jenkins.

### 1.2.3 Создание файла docker-compose.yml

Создадим docker-compose.yml со следующим содержанием (Рисунок 1):

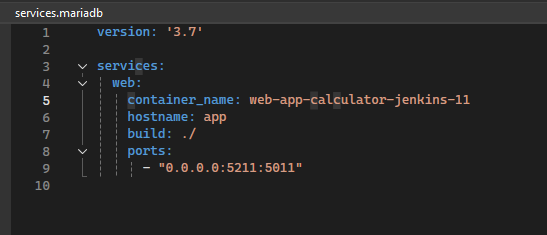


Рисунок 1 – содержание файла docker-compose.yml

### 1.2.4 Создание файла Jenkinsfile

Создадим Jenkinsfile со следующим содержанием (Рисунок 2):

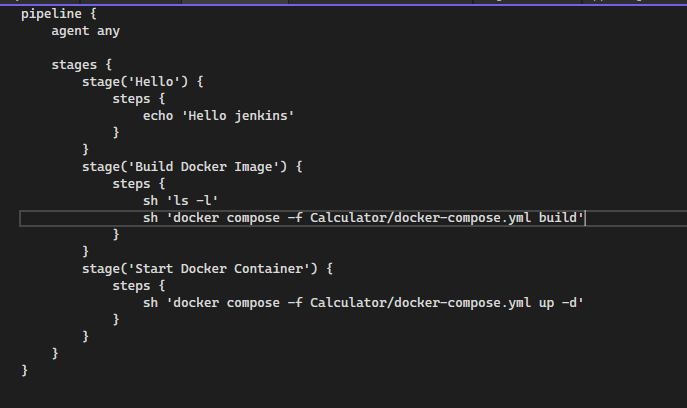


Рисунок 2 – содержание файла Jenkinsfile

Данные инструкции будут выполнены при запуске сборки в Jenkins.

### 1.2.5 Размещение приложения в системе контроля версий GitHub

Разместим наше приложение в отдельной ветке “jenkins”, репозитория “11\_Calculator”, доступного по ссылке:

https://github.com/MashukovI/11\_Calculator.git

### 1.2.6 Настройка автоматической сборки проекта в Jenkins и интеграции с GitHub

Перейдем по адресу 93.88.178.186:8080 и создадим новый item с названием “11\_Calculator”, который будет копировать настройки item “Test”.

### 1.2.7 Выполнение автоматической сборки проекта в Jenkins и анализ результатов

Запустим сборку и получим положительный результат во 4-х стадиях (Рисунок 3)

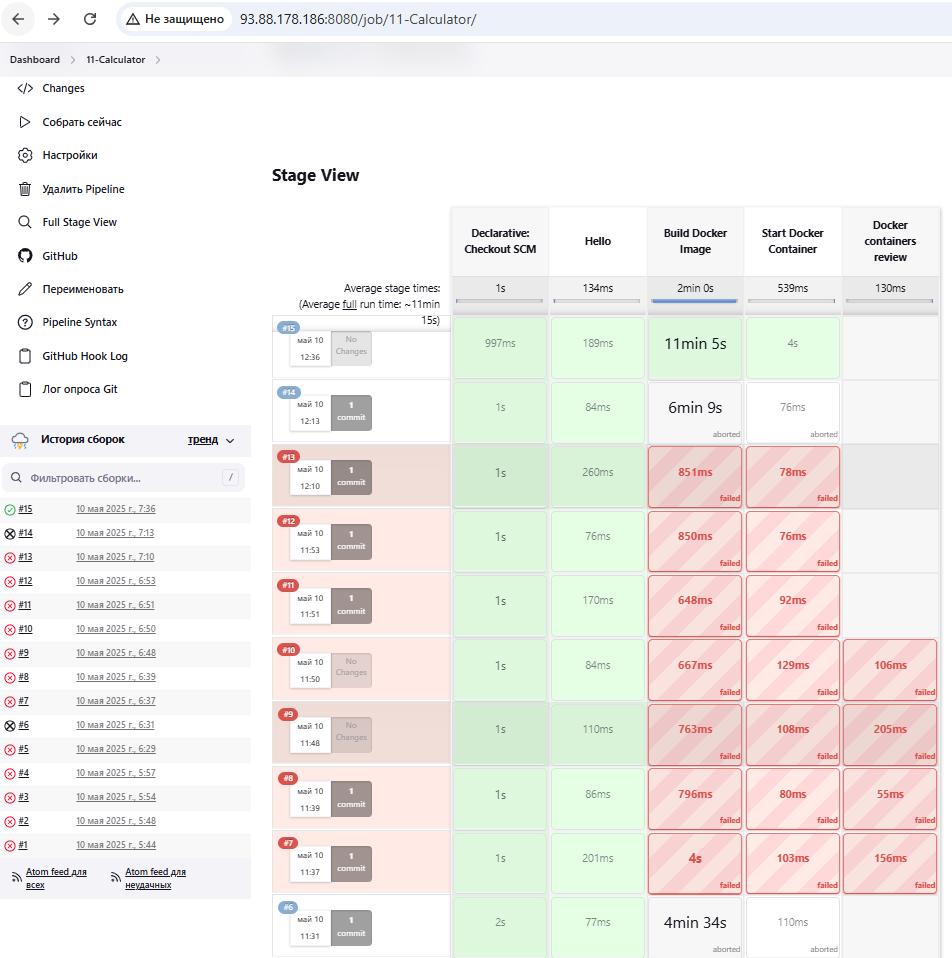


Рисунок 3 – успешная сборка.

Как можно заметить, сборка удалась не с первого раза, но все ошибки были устранены.

На сервере был создан контейнер, который был указан в docker-compose.yml (Рисунок 4).

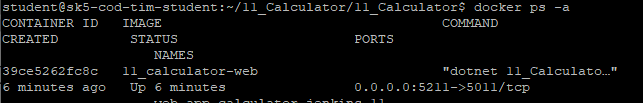


Рисунок 4 – созданный контейнер.

### 1.2.8 Демонстрация автоматической сборки и доставки проекта при корректировке функциональности веб-приложения

Произведем изменения в Index приложения и поменяем с “Машуков Иван Владимирович НМТ-413901” на “Должник Машуков И.В. из группы НМТ-413901” (Рисунок 5) и зафиксируем изменения на GitHub.

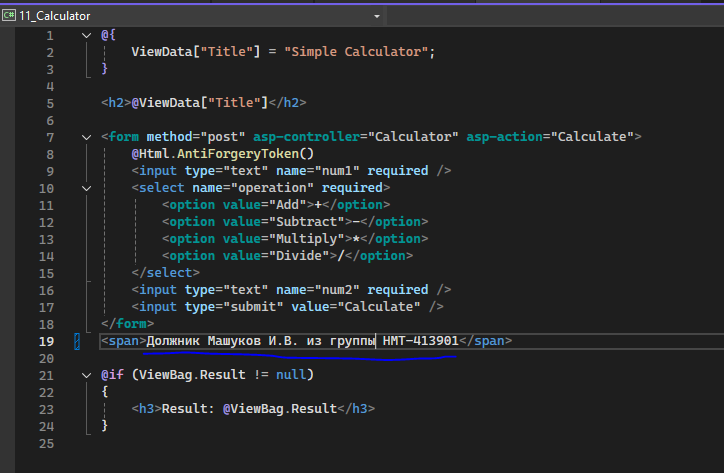


Рисунок 5 – изменения на главной странице

Произведем сборку на Jenkins и получим положительный результат (Рисунок 6).

Изображение выглядит как текст, снимок экрана, программное обеспечение, Значок на компьютере

Автоматически созданное описание

Рисунок 6 – успешная сборка

Перейдем на 93.88.178.186:5211 и увидим изменения (Рисунок 7), которые были в последнем коммите ветки jenkins.

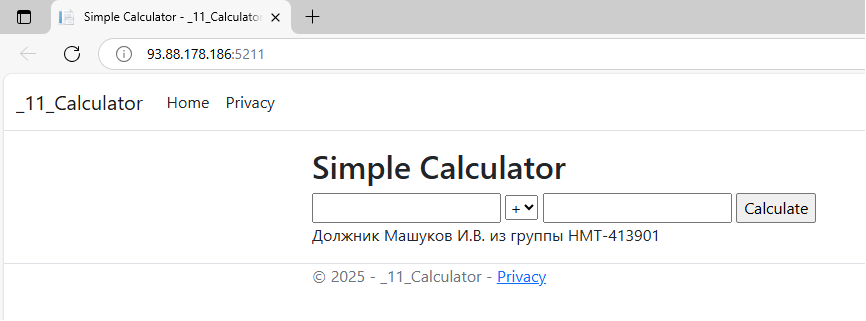


Рисунок 7 – обновленная версия калькулятора.

1.3 Выводы

В данной практической работе было изучена система непрерывной интеграции программного обеспечения Jenkins. Данная система позволяет автоматизировать процесс выгрузки приложения. В моём случае она создаем контейнер и выгружает в него последнюю версию приложения “11\_Calculator”.

Приложение А Листинг программного кода приложения «Калькулятор»   
(контроллер)

using Microsoft.AspNetCore.Mvc;

namespace Calculator.Controllers

{

public enum Operation { Add, Subtract, Multiply, Divide }

public class CalculatorController : Controller

{

[HttpGet]

public IActionResult Index()

{

return View();

}

[HttpPost]

[ValidateAntiForgeryToken]

public IActionResult Calculate(double num1, double num2, Operation operation)

{

double result = 0;

switch (operation)

{

case Operation.Add:

result = num1 + num2;

break;

case Operation.Subtract:

result = num1 - num2;

break;

case Operation.Multiply:

result = num1 \* num2;

break;

case Operation.Divide:

result = num1 / num2;

break;

}

ViewBag.Result = result;

return View("Index");

}

}

}

Приложение Б Листинг программного кода приложения «Калькулятор» (appsettings.json)

{

"Kestrel": {

"Endpoints": {

"Http": {

"Url": "http://0.0.0.0:5011 " // Заменить последние 2 цифры порта на порядковый номер из студенческого журнала. Например, порт 5012 соответствует номеру 12

}

}

},

"Logging": {

"LogLevel": {

"Default": "Information",

"Microsoft": "Warning",

"Microsoft.Hosting.Lifetime": "Information"

}

},

"AllowedHosts": "\*"

}

Приложение В Листинг программного кода приложения «Калькулятор» (appsettings.Development.json)

{

"Logging": {

"LogLevel": {

"Default": "Information",

"Microsoft.AspNetCore": "Warning"

}

}

}

Приложение Г Листинг файла Jenkinsfile

pipeline {

agent any

stages {

stage('Hello') {

steps {

echo 'Hello jenkins'

}

}

stage('Build Docker Image') {

steps {

sh 'ls -l'

sh 'docker compose -f ./11\_Calculator/docker-compose.yml build'

}

}

stage('Start Docker Container') {

steps {

sh 'docker compose -f ./11\_Calculator/docker-compose.yml up -d'

}

}

}

}