

**БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
Факультет радиофизики и компьютерных технологий
Кафедра интеллектуальных систем**

Аннотация к магистерской диссертации

**«Шаблон инфраструктуры для развертывания .NET
приложений в DOCKER-контейнерах и их оркестрации»**

специальность 1-98 80 03 «Аппаратное и программно-техническое
обеспечение информационной безопасности»

Якимович Павел Сергеевич

Научный руководитель: Козадаев Константин Владимирович, доцент
кафедры интеллектуальных систем, к.ф.-м.н.

Минск, 2019

ОБЩАЯ ХАРАКТЕРИСТИКА РАБОТЫ

**.NET CORE, DOCKER-КОНТЕЙНЕРЫ, ОРКЕСТРАЦИЯ,
МИКРОСЕРВИСЫ, ОЧЕРЕДЬ СООБЩЕНИЙ, БАЗА ДАННЫХ,
ЛОГИРОВАНИЕ**

Объект исследования – шаблон инфраструктуры для .NET приложений.

Цель работы – создать шаблон инфраструктуры для приложений, разрабатываемых на платформе .NET Core, разворачиваемых в Docker-контейнерах, и оркестрации контейнеров.

В процессе выполнения работы были рассмотрены и проанализированы платформа .NET Core, Docker-контейнеры, технологии, позволяющие изолировать процессы на одном хосте, технологии для оркестрации контейнеров, технологии доступа к базе данных, очереди сообщений, существующие решения для шаблона инфраструктуры.

Реализован шаблон инфраструктуры, который позволяет начать разработку приложения на .NET Core, которое будет разворачиваться в Docker-контейнере. Контейнеры будут оркестрироваться технологией Docker-compose. Для каждого приложения будет возможность использовать разработанные библиотеки доступа к базам данных (любая база, для которой есть провайдер для EntityFramework Core и MongoDB), логирования и очереди сообщений.

Диссертация состоит из введения, общей характеристики работы, 3 глав, заключения, списка литературы. Объем диссертации составляет 68 страниц, включая 15 рисунков, 6 приложений и ссылок использованных источников из 20 наименований.

АГУЛЬНАЯ ХАРАКТАРЫСТЫКА ПРАЦЫ

NET CORE, DOCKER-КАНТЭЙНЕРЫ, АРКЕСТРАЦЫЯ,
МІКРАСЭРВІСЫ, ЧАРГА ПАВЕДАМЛЕННЯЎ, БАЗА ДАДЗЕНЫХ,
ЛАГАВАННЕ

Аб'ект даследавання – шаблон інфраструктуры для .NET прыкладанняў.

Мэта работы – стварыць шаблон інфраструктуры для прыкладанняў, якія распрацоўваюцца на платформе .NET Core, разгортваюцца ў Docker-кантэйнерах, і аркестрацыі кантэйнераў.

У працэсе выканання працы былі разгледжаны прааналізаваныя платформа .NET Core, Docker-кантэйнеры, тэхналогіі, якія дазваляюць ізаляваць працэсы на адным хасце, тэхналогіі для аркестрацыі кантэйнераў, тэхналогіі доступу да базы дадзеных, чэргі паведамленняў, рашэнні якія існуюць для шаблону інфраструктуры.

Рэалізаваны шаблон інфраструктуры, які дазваляе пачаць распрацоўку прыкладання на .NET Core, якое будзе разгортвацца ў Docker-кантэйнеры. Кантэйнеры будуць аркестравацца тэхналогіяй Docker-compose. Для кожнага прыкладання будзе магчымасць выкарыстоўваць распрацаваныя бібліятэкі доступу да баз дадзеных (любая база, для якой ёсць правайдэр для EntityFramework Core і MongoDB), лагаванне і чэргі паведамленняў.

Дысертация складаецца з увядзення, агульной характарыстыкі работы, 3 глаў, заключэння, спісу літаратуры. Аб'ём дысертациі складае 68 старонак, у тым ліку 15 малюнкаў, 6 дадаткаў і спасылак выкарыстаных крыніц з 20 найменняў.

GENERAL DESCRIPTION OF WORK

.NET CORE, DOCKER-CONTAINERS, ORCHESTRATION,
MICROSERVICES, MESSAGE QUEUES, DATABASE, LOGGING

The object of study is the infrastructure template for .NET applications.

The goal of the work is to create an infrastructure template for applications developed on the .NET Core platform, deployed in Docker containers, and containers orchestration.

In the course of the work, the .NET Core platform, Docker containers, technologies allowing to isolate processes on one host, technologies for container orchestration, database access technologies, message queues, and existing infrastructure template solutions were reviewed and analyzed.

An infrastructure template has been implemented that allows you to start developing an application on .NET Core, which will be deployed in a Docker container. The containers will be orchestrated by Docker-compose technology. For each application it will be possible to use the developed database access libraries (any database for which there is a provider for EntityFramework Core and MongoDB), logging and message queue.

The work consists of introduction, general description of work, 3 chapters, conclusion, list of references. The volume of the work is 68 pages, including 15 figures, 6 attachments and references from sources used from 20 titles.