<SportStore>

devops solution implementation document

Version 1.0

For

<CLIENT>

Contents

[1 Introduction 5](#_Toc533782317)

[1.1 Purpose 5](#_Toc533782318)

[1.2 Definitions 5](#_Toc533782319)

[1.3 References 5](#_Toc533782320)

[2 Background 6](#_Toc533782321)

[2.1 System Overview 6](#_Toc533782322)

[2.2 Business Case 6](#_Toc533782323)

[2.2.1 Business Goals 6](#_Toc533782324)

[2.2.2 Major Features 7](#_Toc533782325)

[2.3 Business Constrains 7](#_Toc533782326)

[2.4 Service Level Agreement 8](#_Toc533782327)

[2.4.1 Scope and Applicability 8](#_Toc533782328)

[2.4.2 Service Quality Guarantees 8](#_Toc533782329)

[2.4.3 Definitions 8](#_Toc533782330)

[3 Technology stack Description 9](#_Toc533782331)

[3.1 Environment Diagram 9](#_Toc533782332)

[3.1.1 Representation 9](#_Toc533782333)

[3.1.2 Element Catalog 9](#_Toc533782334)

[3.1.3 Variability 9](#_Toc533782335)

[3.1.4 Reasoning 9](#_Toc533782336)

[3.2 Deployment Diagram 10](#_Toc533782337)

[3.2.1 Representation 10](#_Toc533782338)

[3.2.2 Element Catalog 10](#_Toc533782339)

[3.2.3 Variability 10](#_Toc533782340)

[3.2.4 Reasoning 10](#_Toc533782341)

[4 Risk Analysis 11](#_Toc533782342)

[5 Operation Plan 12](#_Toc533782343)

[5.1 Transition Phase 12](#_Toc533782344)

[5.1.1 Infrastructure 12](#_Toc533782345)

[5.1.2 Environments 12](#_Toc533782346)

[5.1.3 Provisioning 13](#_Toc533782347)

[5.1.4 Management 13](#_Toc533782348)

[5.1.5 Monitoring 14](#_Toc533782349)

[5.2 Operation Phase 14](#_Toc533782350)

[5.2.1 Service Operation Support 14](#_Toc533782351)

[5.2.2 CMOD Activities 14](#_Toc533782352)

[5.2.3 Continuous Improvement Process 14](#_Toc533782353)

[6 Appendix A – References 15](#_Toc533782354)

Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| Date | Version | Section | Author |
| 1/15/2019 | 1.0 | Initial Release | Zakharchenko V |
| 1/15/2019 | 1.1 | Technology stack description | Bahmet S |
| 1/15/2019 | 1.2 | Deploy environment | Husak O |
|  |  |  |  |
|  |  |  |  |

# Introduction

## Purpose

## Definitions

The Definitions section lists the acronyms and terms used in this document which might possess lesser familiarity or double meaning to the reader.

|  |  |  |
| --- | --- | --- |
| # | Term | Definition |
|  | VM | Virtual machine |
|  | SQL | Structured Query Language |

# Background

## System Overview

The project shows how to create a ASP.NET core web application, that is connected to the SQL database, package it in docker container and deploy it in VM.

## Business Case

The intent of this project is developing cloud based online sport store.

Data managers should be able to add product entry via admin panel including name of product, basic information about the product: name, description, price, category. Preferable way to keep products is secure blob storage. The solution will be deployed in Docker container.

Customers should be able to view general product information and also add chosen product to the Cart.

### Business Goals

The section enumerates essential business goals for the solution.

|  |  |
| --- | --- |
| # | Description |
|  | MVP of Online sport store should be delivered in 3 month |
|  | Support online access for customers and data managers |
|  | Implement DevOps best practices |
|  | Optimize expenses for the 1st quarter of running MVP solution to $2K per month |

### Major Features

The section enumerates solution major features.

|  |  |
| --- | --- |
| # | Description |
|  | Achieve product entry time 15 min per 1 product |
|  | Achieve expenses threshold $2K per month for the 1st quarter counting from project start date |
|  | Anticipated load is 1000 users/min and 5000 users |
|  | There is 4 envs: development, testing, staging and production |

## Business Constrains

The section lists the constraints that provided the significant influence over the architecture accounted for in the designed solution. These can be of business, technical, resource, and other types.

|  |  |
| --- | --- |
| # | Description |
|  | The company puts a limited budget toward building the solution assuming up to 5 people total participating in the solution implementation and deployment. |
|  | To reach the company's business objectives the first version of the solution is expected to be built within 3 months from the start of implementation. |
|  | Online sport store users should have secure access to the web interface |

# Technology stack Description

This section enumerates preferable stack technologies and tools that were used in solution.

Frameworks:

* ASP.NET core
* Ms Unit Test
* Entity Framework

Develop environment:

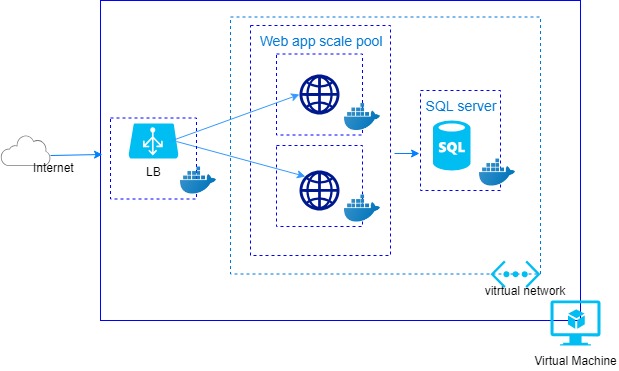
* Visual studio 2017
* Git

Deploy environment:

* Docker
* Docker-compose

## Environment Diagram

### Representation



This reference diagram shows how to deploy VMs and a virtual network configured for Web application using SQL Server on Windows.

### Element Catalog

Table of annotated elements

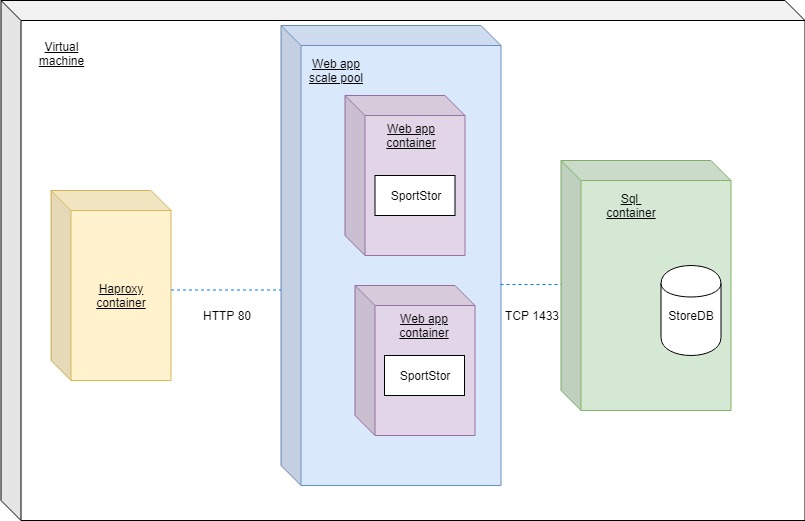
|  |  |
| --- | --- |
| Name | Description |
| Virtual network | Docker virtual network allows resources such as VMs to securely communicate with each other. |
| Virtual Machine | Linux virtual machine |
| LB | Azure load balancer distributes inbound traffic according to rules and health probes |
| Web scale set | The number of docker containers can automatically increase or decrease in response to demand or a defined schedule. |
| Sql server | Sql server container |

.

## Deployment Diagram

### Representation

This diagram shows big picture of resources that is provisioned into Docker container. A scale set makes it easy to deploy and manage a set of identical containers, and decrease load on web app. As the load on the containers is decreased in proportion to scale number by the load balancer.



### Element Catalog

Table of annotated elements

|  |  |
| --- | --- |
| Name | Description |
| Haproxy container | Haproxy load balancer distributes inbound traffic according to rules |
| Web app scale poolt | The number of web containers. |
| Web app container | The instanse of Web application |
| Sql container | The instanses of SQL server |
| SportStore | The artifact of web application |
| StoreDB | The artifact of database |

### Variability

Linux and Windows can be used interchangeably since the infrastructure isn't dependent on the operating system. This solution forses to replace ASP.NET framework by ASP.NET core.

### Reasoning

Migrating to the Linux platform brings some profit. According calculation renting of Linux resources reduse hosting expanses.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | Docker Container | | Docker Resource | Virtual Machine | | VM Resource |
|  | Web | SQL |  | Web | SQL |  |
| OS | - | - | 0 | OS(ubuntu) for VM | OS(ubuntu) for VM | 2 |
| CPU | 1 Virtual Core | 1 Virtual Core | 2 | For OS Minimum: 1 core 1.4 GHz 64-bit processor For Web Service Minimum: 1 core 1.4 GHz 64-bit processor | For OS Minimum:  1 core 1.3 GHz x64 For SQL Minimum: 1 core 1.4 GHz x64 | 4 |
| RAM | 1GB | 1GB | 2 | 2 GB (for OS and Web service) | 2 GB (for OS and SQL service) | 4 |
| HDD | 1,8 GB | 1,8 GB | 3,6 | 10 GB (For OS and all services) | 5GB OS 6GB for MSSQL | 21 |
| Compatibility | cross-platform | cross-platform | + | Depends on hypervisor | Depends on hypervisor | - |
| RepeatAbility | + | + | + | Depends on hypervisor | Depends on hypervisor | - |

# Operation Plan

Typically, each service has two key phases of its lifecycle: Transition Phase and Operation Phase.



The state while the solution stays in active development mode is called **Transition Phase**. During this period main activities related to Infrastructure management will be performed by system engineering team, especially by Configuration Managers (CMs). Their goal is to implement all technical solutions to make the product ready to enter the Operation Phase.

Transition Phase ends up with the Production Release.

Operations Team will be responsible for performing the activities during the **Operation Phase**. Key goal at this phase is to maintain the desired quality of service and Service Level Agreement.

## Transition Phase

The transition phase shall meet specific requirements that provides reliable team developing of ASP.NET core web application. In general the transition phase solves the following task: teams collaboration, share code, track work, unit testing and ship software.

### Infrastructure

#### Virtualized Resources:

* **Web and SQL** container. Availability of Virtual machines on Linux, Window 2012 server and Windows 2016 server
* **Virtual network**. Every docker containers is deployed into a virtual network that could be accessed through load balancer.
* **Load balancers**. Use [Azure Load Balancer](https://docs.microsoft.com/en-us/azure/load-balancer/) to distribute network traffic from internet to the web tier.

### Environments

This section enumerates tools that help to implement reliable and flaxable solution in order to develop ASP.NET core Web application.

#### Development Environment

Changing application source code and source control are provided by the following tools:

* Visual studio 2017 Community
* Git.

#### Staging Environment

The application is hosted in docker containers. The big picture of resources is provided on the deployment diagram.

### Provisioning

#### Packaging

Docker provides creating packages as part of a build process.

#### Deployment

Deployment involves two steps:

* Provisioning the Database resources. At the project are used powershell script.
* Deploying the application (code, binaries, and content files). This step includes running whole containers with help of Docker-compose.

### Management

#### System Security Management

Web scalable set is exposed indirectly to the Internet. Clients connect using the public IP address associated with load balancer.