

**SQL Server 2017 Running in Linux (OpenShift Container Platform)**

**Hands-On Lab Manual – Running Microsoft SQL Server in OpenShift Container Platform, and connect it using .NET Core Application**

**Published**: April 2017

**Applies to**: Microsoft SQL Server 2017 in Container – Build 14.0.500.272 (CTP2.0)

**Summary**: Docker Containers are an isolated, resource controlled, and portable operating environments. An application inside a container can run without affecting the rest of the system and vice versa. SQL Server in Docker container will provide customers with even more flexibility in their data solution. SQL Server on OpenShift Container Platform gives the power to run and deploy SQL Server in OpenShift Cluster and scale up and down as per the need.

Copyright

The information contained in this document represents the current view of Microsoft Corporation on the issues discussed as of the date of publication. Because Microsoft must respond to changing market conditions, it should not be interpreted to be a commitment on the part of Microsoft, and Microsoft cannot guarantee the accuracy of any information presented after the date of publication.

This Lab Guideline is for informational purposes only. MICROSOFT MAKES NO WARRANTIES, EXPRESS, IMPLIED, OR STATUTORY, AS TO THE INFORMATION IN THIS DOCUMENT.

Complying with all applicable copyright laws is the responsibility of the user. Without limiting the rights under copyright, no part of this document may be reproduced, stored in, or introduced into a retrieval system, or transmitted in any form or by any means (electronic, mechanical, photocopying, recording, or otherwise), or for any purpose, without the express written permission of Microsoft Corporation.

Microsoft may have patents, patent applications, trademarks, copyrights, or other intellectual property rights covering subject matter in this document. Except as expressly provided in any written license agreement from Microsoft, the furnishing of this document does not give you any license to these patents, trademarks, copyrights, or other intellectual property.

© 2017 Microsoft Corporation. All rights reserved.

Microsoft, Active Directory, Microsoft Azure, Excel, SharePoint, SQL Server, Windows, and Windows Server are trademarks of the Microsoft group of companies.

All other trademarks are property of their respective owners.

Contents

[Overview 4](#_Toc481518320)

[What software are installed on Labs 5](#_Toc481518321)

[Setting up your OpenShift environment in three (3) Steps 5](#_Toc481518322)

[Step 1 - Run Docker for Windows 5](#_Toc481518323)

[Step 2 - Run OpenShift Cluster on Windows Lab 7](#_Toc481518324)

[Step 3 - Login to OpenShift Web Console 9](#_Toc481518325)

[Create and browse your MS SQL + .NET Core Container App in Five (5) Steps 10](#_Toc481518326)

[Step 1 - Run SQL Server on OpenShift Container Platform 10](#_Toc481518327)

[Step 2 - Openshift Port-Forwarding from MSSQL Container 14](#_Toc481518328)

[Step 3 -Connect to SQL Server Container using SQL Server Management Studio 15](#_Toc481518329)

[Step 4 - Create Database on MSSQL Container 17](#_Toc481518330)

[Step 5 - Browse the MSSQL Container Application 19](#_Toc481518331)

# Overview

Note: Estimated time to complete this lab is 10-15 Minutes.

Docker Containers are an isolated, resource controlled, and portable operating environments. An application inside a container can run without affecting the rest of the system and vice versa. The SQL Server 2017 is now available for Windows and Linux Docker engine. This will enable SQL Server to deliver a consistent data platform across Windows Server and Linux, as well as on-premises and cloud. SQL Server in Docker container will provide customers with even more flexibility in their data solution. SQL Server on OpenShift Container Platform gives the power to run and deploy SQL Server in OpenShift Cluster and scale up and down as per there need.

This lab was built based on the features and functionality available in the SQL Server container image - Build 14.0.500.272 (CTP 2.0) for Linux on OpenShift container platform. As the Microsoft team continues toward general availability of SQL Server, users can expect to see additional features and capabilities in later releases of the product.

This manual will guide you through the steps to run SQL Server on OpenShift Container Platform and connect .NET Core application with SQL Server running in container. With this lab, you will be able to:

* Run OpenShift cluster on windows lab
* Successfully run SQL Server on OpenShift Container Platform
* Learn how to connect to SQL Server running in container with SQL Server Management Studio
* Create database and tables using SQL Server Management Studio
* Learn how to verify the connectivity between .NET Core application and SQL Server Database

# What software are installed on Labs

The lab is preconfigured with below software and tools.

1. **Docker for Windows**: The Docker for Windows brings the Docker capabilities on Windows machine. It allows us to run the containerized app on Windows System. The detailed instructions for installing Docker on Windows is available [here](https://docs.docker.com/docker-for-windows/install/).
2. **OC Binary** **V1.5.0**: This is OpenShift Command Line utility. Download the OC Binary from [here](https://github.com/openshift/origin/releases/tag/v1.5.0).
3. **MS SQL Docker Image**

The **microsoft/mssql-server-linux** image is used to run container hosted MSSQL Server on OpenShift.

1. **Dot Net Core 1.0**

The **registry.acess.redhat.com/dotnet/dotnetcore-10-rhel7:1.0** image is used to run and host front-end application that’s created on .NET Core 1.0 platform.



1. Lab-scripts

* **01-cluster-up.bat**: Script to run OpenShift cluster on Windows Lab
* **02-port-forward.bat**: Script to forward MSSQL Container port
* **03-PersonDirectory-db-script.sql**: Script to create Database and Table required by application
* **04-cluster-down.bat**: Script to stop OpenShift cluster

1. Visual Studio 2015
2. SQL Server Management Studio

# Setting up your OpenShift environment in three (3) Steps

## Step 1 - Run Docker for Windows

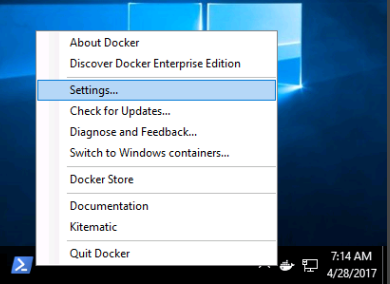
The section below will guide you to run Docker for Windows service on Windows lab. You will be using Lab on Demand Environment to perform this step. Login to the Windows lab environment and follow below steps. Ask the presenter if you don’t have windows lab information provided to you

1. Once logged in to Windows lab, double click on the **Docker for Windows** icon available on **Desktop**. It will start the Docker Engine on Windows within a minute.

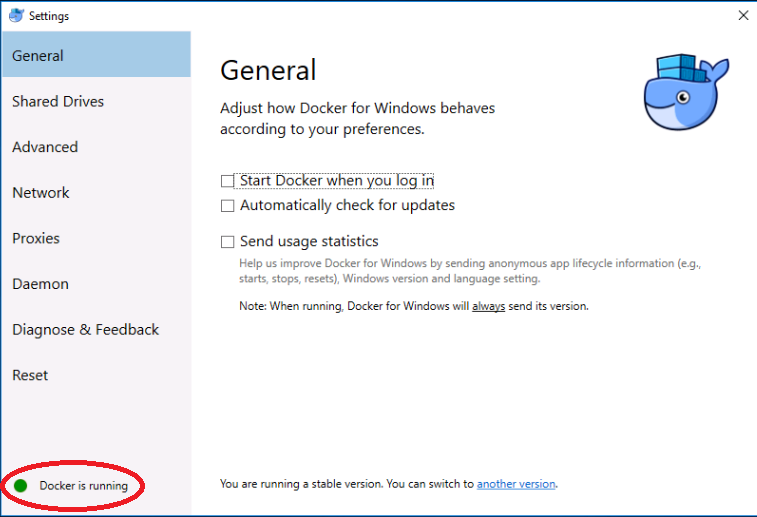
The Docker for Windows allows us to run the containerized app on Windows System. Check out [Docker for Windows](https://docs.docker.com/docker-for-windows/) to know more.



1. To verify whether Docker engine is running, right click on the **Docker** icon available in the **notification task bar** and select **Settings…** from options.



1. On the **Settings** window, check **Docker is running** from the bottom left corner as specified in below screen.



1. Now you have Docker engine running on your Windows machine. Next we will see how to run OpenShift cluster on Windows Lab. Close the Docker **Settings** window and follow the next section.

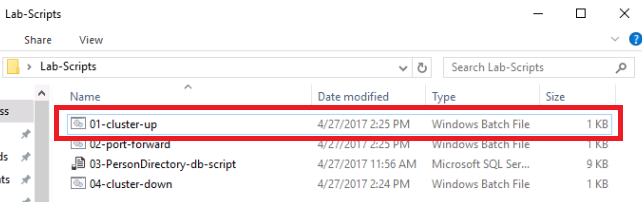
## Step 2 - Run OpenShift Cluster on Windows Lab

The section below will guide you to run OpenShift cluster on Windows lab.

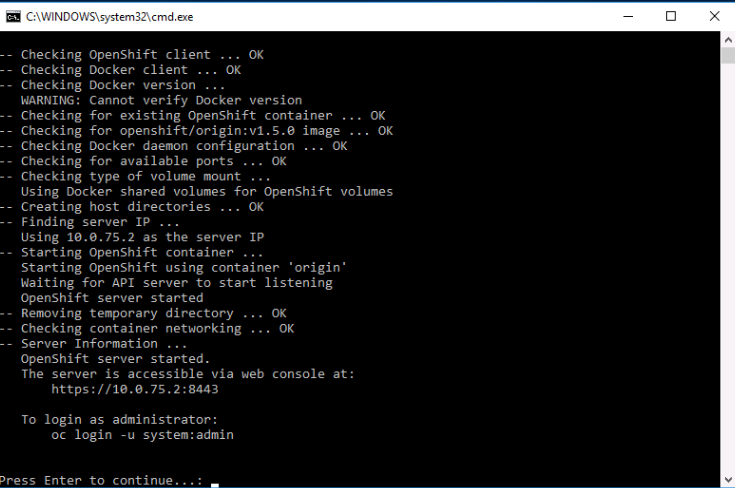
1. Now Docker engine is up and running, it’s time to run the OpenShift cluster on Windows machine. For this, open the **Lab-Scripts** folder from the **Desktop** and run **01-cluster-up.bat** batch file.

The 01-cluster-up.bat file runs the *oc cluster up* command behind the screen. The *oc cluster up* command starts a local OpenShift all-in-one cluster with a configured registry, router, image streams, and default templates. Click [here](https://github.com/openshift/origin/blob/master/docs/cluster_up_down.md) to know more.





1. Once, OpenShift server is started on windows lab environment, the OpenShift details is prompted in the **Command Prompt** as specified in screen below. Press **Enter** to auto launch OpenShift Web Console in a web browser.



**NOTE**: If OpenShift Web Console is not launched in browser, open the **Google Chrome** and navigate to the OpenShift server web console URL displayed in **Command Prompt** [E.g. <https://10.0.75.2:8443>]

1. If you see any SSL certificate privacy error on the browser, ignore it by clicking on **Advance and proceed**.

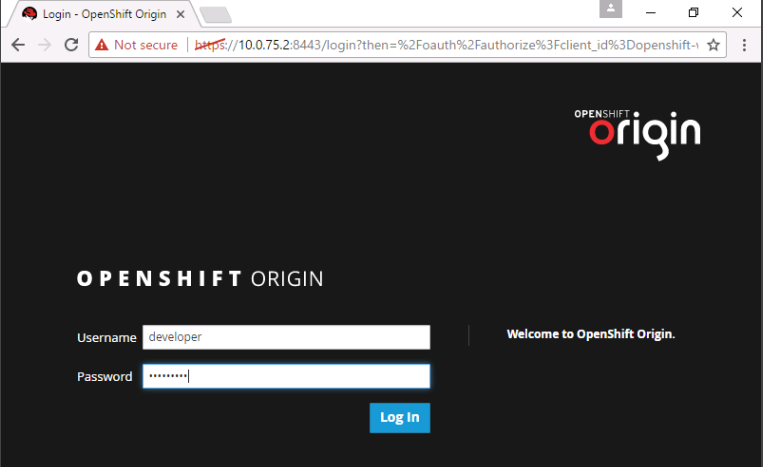
## Step 3 - Login to OpenShift Web Console

The section helps you to login on OpenShift origin web console.

1. On **Welcome to OpenShift** **Origin** page use below credentials to login to OpenShift Web Console and click **Log In** button.

Username: **developer**

Password: **developer**



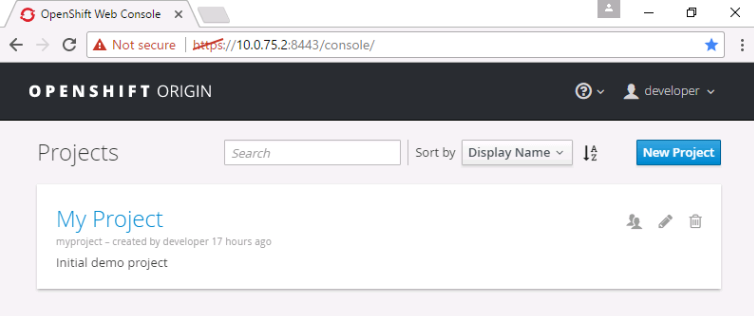
1. You will be redirected to **Projects** page after login. This ensures that the OpenShift cluster is up and running on windows lab.

# Create and browse your MS SQL + .NET Core Container App in Five (5) Steps

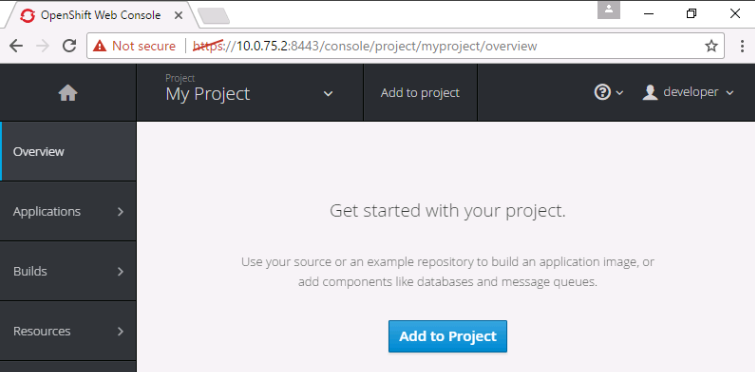
## Step 1 - Run SQL Server on OpenShift Container Platform

Follow below instructions to run SQL Server on OpenShift Container Platform –

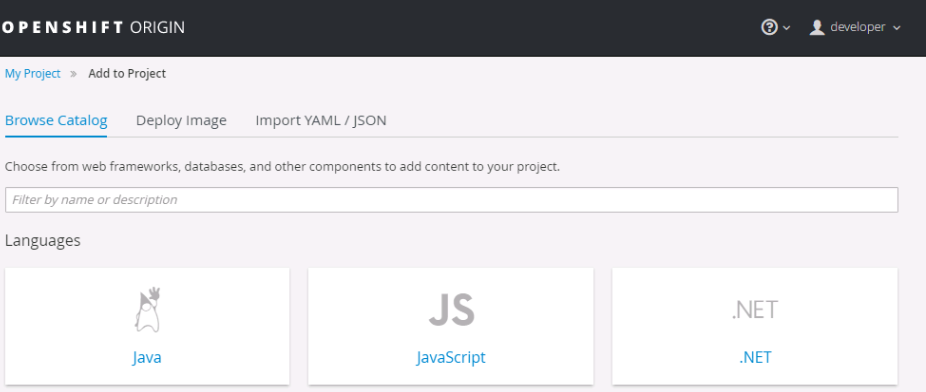
1. On the **OpenShift Web Console** that you have launched in previous step, select **My Project** from the **Projects** screen.



1. Select **Add to Project** button to create the application in the project.

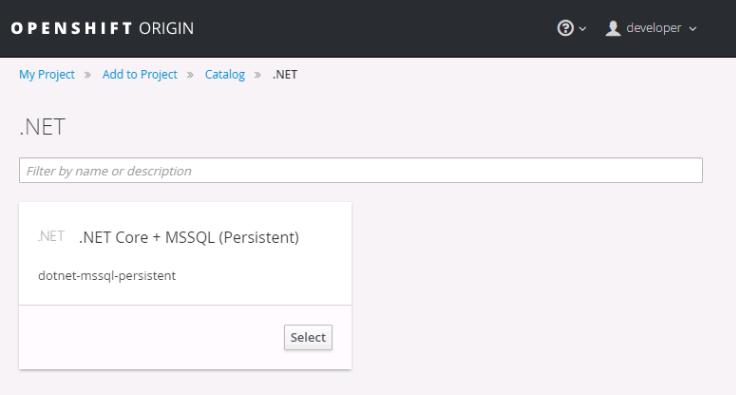


1. Select **.NET** in the **Browse Catalog**



1. Now, click on **Select** buttonin **.NET Core + MSSQL (Persistent)**.

The **.NET Core + MSSQL (Persistent)** is application template which will run two containers on OpenShift Container Platform, one with the front-end application, which is a sample application built using .NET Core 1.0 which will connect with SQL Server container to store and retrieve records.



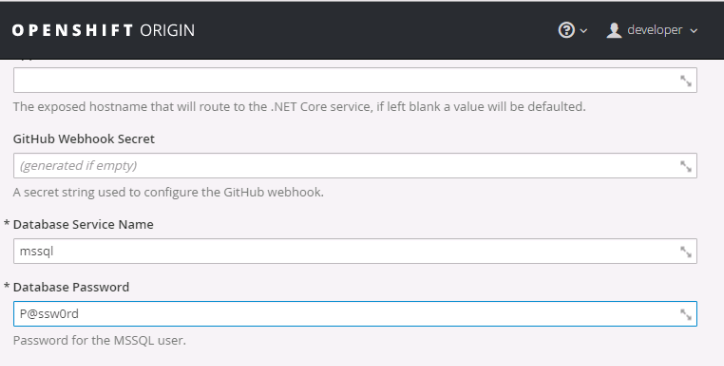
1. Notice the Docker Images used by .NET Core + MSSQL (Persistent) template.

The **registry.acess.redhat.com/dotnet/dotnetcore-10-rhel7:1.0** image is used to run and host front-end application that’s created on .NET Core 1.0 platform.

The **microsoft/mssql-server-linux** image is used to run container hosted MSSQL Server on OpenShift.

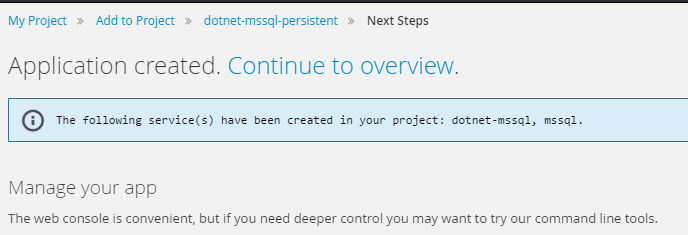


1. On the **.NET Core + MSSQL (Persistent)** application parameter section, scroll down and set the **\*Database Password** parameter value. Leave the other parameter values as default.



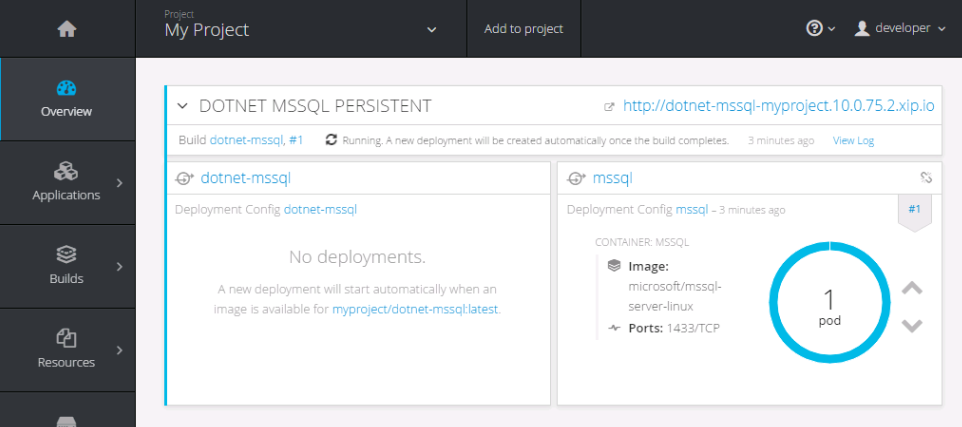
**NOTE**: Make sure to specify a strong password (Minimum length 8 characters, including uppercase and lowercase letters, base 10 digits and/or non-alphanumeric symbols). For this lab use the **P@ssw0rd** as password.

1. Scroll down and click on **Create** button.
2. On the **Application created** screen, click on **Continue to overview** link to view the status of application containers.



1. On the **Overview** page, notice that two different services are created, **dotnet-mssql** and **mssql**. The .NET Core application container will run in **dotnet-mssql** and SQL Server container will run in **mssql**.
2. Within a minute, **mssql** container will be up and running as specified in below screen. Whereas, the **dotnet-mssql** container will take 2-3 minutes to build and deploy application.

The front-end application source code is available at <https://github.com/click2cloud/dotnet-core-mssql-app>.



1. Next, we will do the mssql container port forwarding and will connect with the SQL Server for creating database and tables.

## Step 2 - OpenShift Port-Forwarding from MSSQL Container

This section covers instructions to forward port of mssql container running on OpenShift to localhost.

1. Click on the **Windows PowerShell** icon from the task bar to launch Windows PowerShell window.



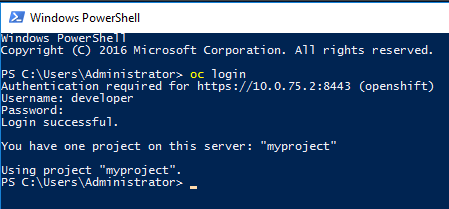
1. Now in the Windows PowerShell run below command

**oc login**

Provide below username and password when asked in command prompt

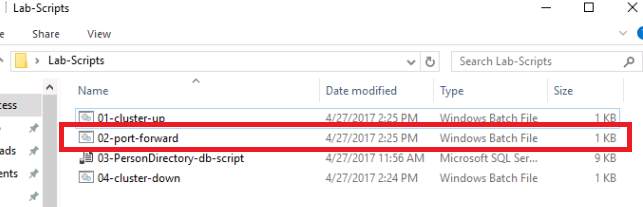
Username: **developer**

Password: **developer**

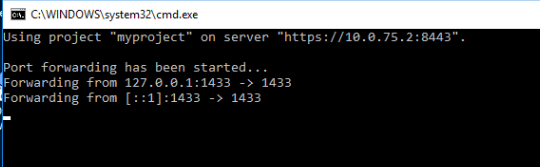


1. To do the port-forwarding of mssql container running on OpenShift, first run the **02-port-forward.bat** batch file available in **Lab-Scripts** folder on **Desktop**.

The 02-port-forward.bat file runs **oc port-forward <pod> <localport>:<remoteport>** in background. Click [here](https://docs.openshift.com/container-platform/latest/dev_guide/port_forwarding.html) to learn more about port-forward command.



1. You will see below output in **Command Prompt**, once port is forwarded successfully.



**NOTE**: Do not close this port-forwarding cmd window else the port-forwarding will stop. You should minimize it and follow the next section.

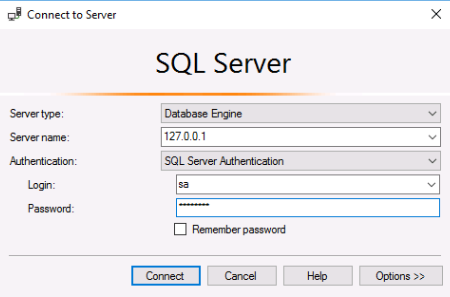
## Step 3 - Connect to SQL Server Container using SQL Server Management Studio

This section covers instructions to connect to SQL Server container running on OpenShift Container Platform using SQL Server Management Studio.

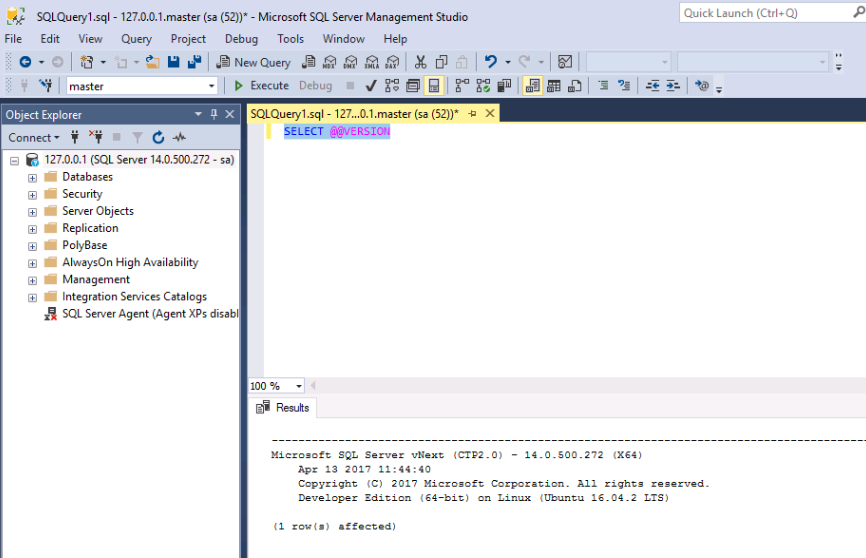
1. Now, open the **Microsoft SQL Server Management Studio** from the Taskbar.



1. On the Connect to Server window, the **Server name** is already set to 127.0.0.1. As we have port forwarded the SQL Server container port to local port, we can connect to server using local IP address i.e. 127.0.0.1.
2. Provide the **Password** provided while creating application on OpenShift and click on **Connect**. Use the **P@ssw0rd** in password.



1. Verify the connection is established from the **Object Explorer** and select the **New Query** option from the toolbar, type **SELECT @@VERSION** and **Execute** to check the SQL Server version as specified in below screen.

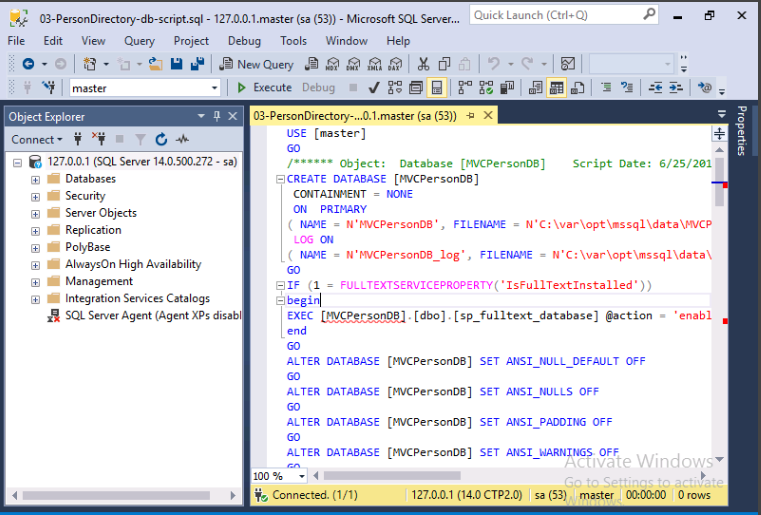


## Step 4 - Create Database on MSSQL Container

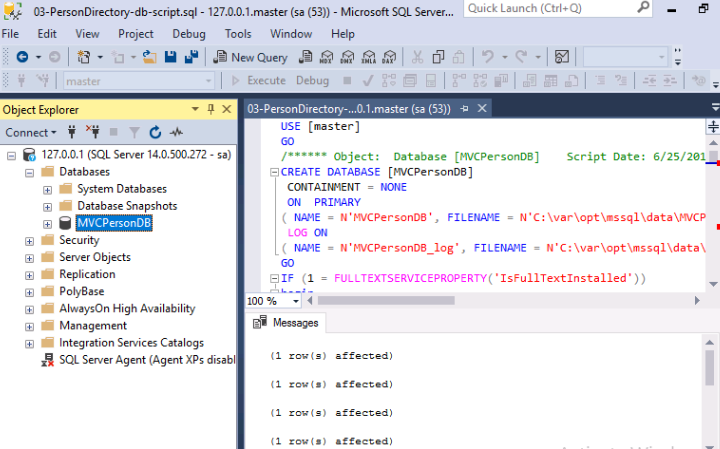
This section covers instructions to create sample **MVCPersonDB** database on SQL Server running on OpenShift Container Platform using SQL Server Management Studio.

1. Now, double click the **03-PersonDirectory-db-script.sql** SQL Script available in **Lab-Scripts** folder on **Desktop**. This will open **03-PersonDirectory-db-script.sql** file in SQL Server Management Studio.

The **03-PersonDirectory-db-script.sql** contains scripts to create **MVCPersonDB** database and **Person** table. It will also insert records in Person table.

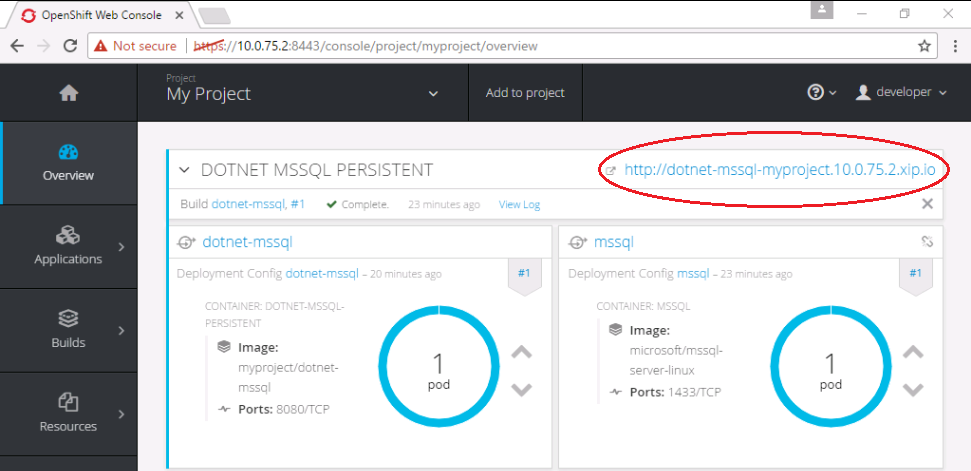


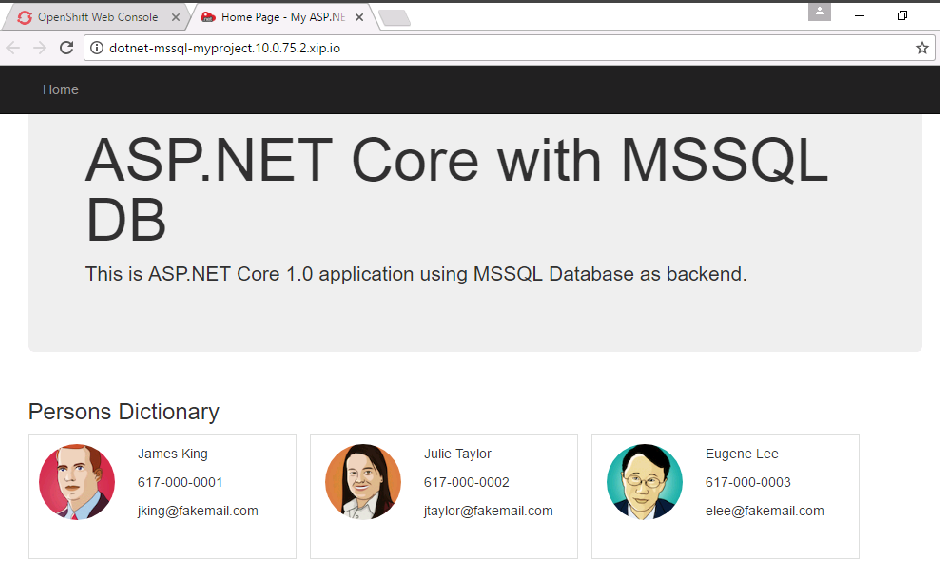
1. Click on **Execute** button available on toolbar of SQL Server Management Studio and verify that scripts are executed successfully.
2. Now, to verify whether the **MVCPersonDB** database created successfully, expand the **Databases** in the **Object Explorer** window available in left panel.



## Step 5 - Browse the MSSQL Container Application

This section covers instructions to browse .NET Core with MSSQL container application running on OpenShift.

1. Now, go back to **OpenShift Web Console** and verify whether the **dotnet-mssql** container is up and running or wait for some time for its running state. Once in running state, click on the route URL as shown in the below screen to browse the application.
2. Try inserting new records and verify the data is inserted in database. This ensures that you have successfully connected to SQL Server running on OpenShift Container Platform.



You have successfully completed the Lab **Run SQL Server in OpenShift Container Platform.**

Thank You…