Coding Challenge System (CCS) Team

Deployment Document

Document Category: DEV

Version: 0.2.1

Status: complete

Last Updated: March 20, 2022

Table of Contents

[Deployment and Setup 3](#_Toc100575468)

[Pre-requisites 3](#_Toc100575469)

[Web Server 3](#_Toc100575470)

[Database Server 3](#_Toc100575471)

[Setup & Configurations 3](#_Toc100575472)

[AMS Roles 3](#_Toc100575473)

[Coding Challenge System Configuration 4](#_Toc100575474)

[Publishing Steps 4](#_Toc100575475)

[IIS Configuration 16](#_Toc100575476)

# Deployment and Setup

## Pre-requisites

Before installing the Coding Challenge System, please make sure the following environment requirements are met.

### Web Server

#### Hardware Requirements

1. Minimum CPU requirement per instance: 2.7GHz CPU (2 processors)
2. Required minimum ram: 8GB of RAM
3. Minimum disk space per instance: 250 MB of disk space.
4. Minimum network capacity requirements (1gigE or 10gigE): 5-10 Mb / sec of minimum network capacity.

#### Software Requirements

1. Software architecture: 32-bit or 64-bit architecture
2. .Net Core 3.1
3. IIS version 10

### Database Server

#### Hardware Requirements

1. Hardware Requirements
2. Minimum CPU requirement per instance: 2.7GHz CPU (2 processors)
3. Required minimum ram: 8GB of RAM
4. Minimum disk space per instance: 200 MB of disk space.
5. Minimum network capacity requirements (1gigE or 10gigE): 5-10 Mb / sec of minimum network capacity.
6. Clara Linked Server

#### Software Requirements

1. Microsoft SQL Server
2. Linked Server to Clara

## Setup & Configurations

### AMS Roles

First, logon to AMS for the environment you’re deploying to.

The application code is CCS.

The roles needed for using this system are the Student (ST), Teacher (TE) and Coordinator (CO) roles.

### Coding Challenge System Configuration

Web configuration

|  |  |  |
| --- | --- | --- |
| Attribute Name | Example Value | Notes |
| Database Name | CCS |  |
| Database Server | csdev.cegep-heritage.qc.ca |  |
| AppPool | CSAppPool | Your application must use its own CSAppPool |
| URL to application | https://csdev.cegep-heritage.qc.ca/Projects/CCS/Login |  |
| AMS Server | Csdev.cegep-heritage.qc.ca |  |
| Database | Microsoft SQL Server |  |
| Server Name | csdev.cegep-heritage.qc.ca |  |
| Password | TEAMCCS | Only for dev |
| Username | TEAMCCS | Only for dev |
| Integrated Security | False |  |
| MultipleActiveResultSets | False |  |

Table 21 Example Dependent Service Configuration

## Publishing Steps

#### Database deployment

1. Follow this link to [CSAZURE](https://csazure.cegep-heritage.qc.ca:8080/F2021-DevProject/_git/Code%20Challenge?path=%2FDocumentation%2FScripts%2FCCS_Script.sql). In CS Azure, Navigate to Documentation/Scripts and download CCS\_Script.sql in the latest CCS release branch (the latest release has the highest versioning numbers as part of the branch name, for example CCS\_Release0.0.1).

A screenshot of a computer

Description automatically generated with medium confidence

1. Before running the script make sure you have an appPool in IIS that is free to run this instance of CCS. The appPool must not have an CCS already linked to it. We recommend using CSAppPool1 as there should only ever be 1 deployed CCS on the IIS.
2. Navigate to lines 84-92 and chance references to CSAppPool1 to whichever appPool you end up wanting to use in IIS. We recommend leaving it CSAppPool1 unless there are issues running it on that appPool.
3. On the server the system is going to be deployed to, open SQL Server Management Studio and run the script to create the empty database.

#### Pre-Deployment Steps

1. Get the project from CS Azure by navigating to the following [link](https://csazure.cegep-heritage.qc.ca:8080/F2021-DevProject/_git/Code%20Challenge?path=%2FDocumentation%2FScripts%2FCCS_Script.sql). Open it in Visual Studio. If the solution is already open you can skip to step 4.  
   Text

   Description automatically generated
2. Ensure that the service reference for the LoginService is to the login service in the appropriate server (If the server is CSTEST, make sure the reference is set properly to CSTEST and not CSDEV as shown in the example below).  
   The link for CSTEST: <https://cstest.cegep-heritage.qc.ca/Login/Login.asmx>?WSDL  
   The link for CSDEV: <http://csdev.cegep-heritage.qc.ca/Projects/Login/Login.asmx?WSDL>

Graphical user interface, text

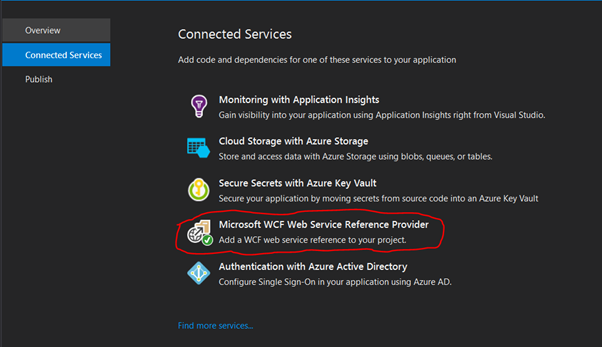
Description automatically generated

A picture containing text

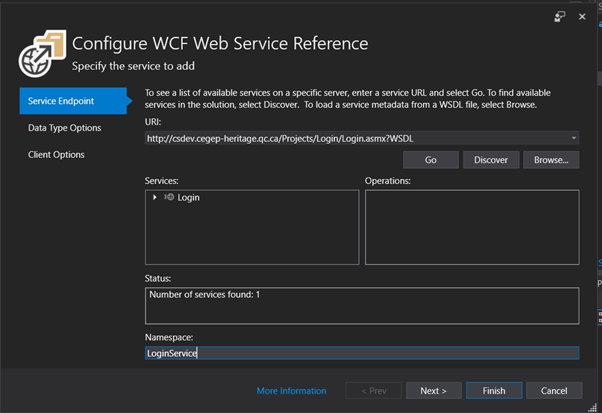
Description automatically generated

To do that, right-click on the existing LoginService folder, click on “Delete” and click “Ok” on the warning.

Then double-click on the “Connected Services” and select **Microsoft WCF Web Service Reference Provider**



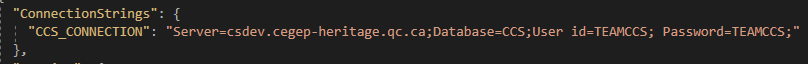
The URI should be associated with the appropriate AMS Service dependant on what server CCS is being deployed to.



1. Make sure to change the connection string to the appropriate database server.

Text

Description automatically generated



1. Ensure that SQL Server connection string in CCSContext.cs file matches the one used in appsettings.json.
2. In CCSContext.cs:

Text

Description automatically generated

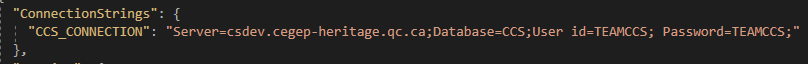
A screenshot of a computer

Description automatically generated with medium confidence

1. In appsettings.json:

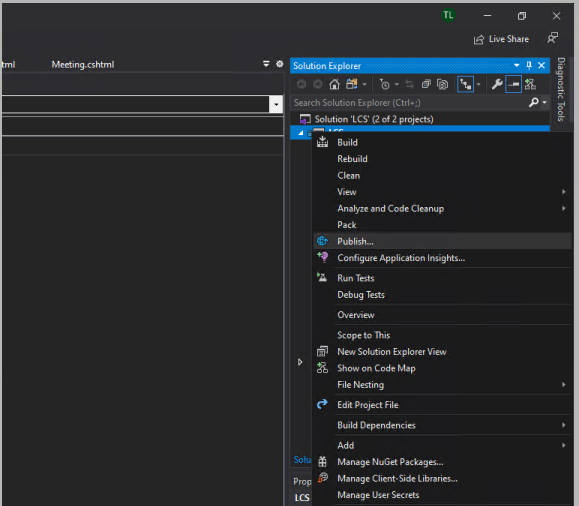
Text

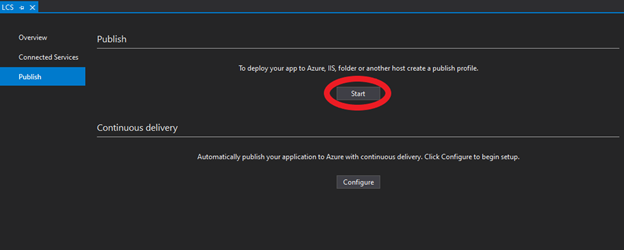
Description automatically generated



#### Publishing

1. Right click on the CCS solution and click on publish.





You may have to create a new IISProfile to deploy if one is already created, if so just select the “new” button.

1. Be sure to select Web Server (IIS) and Web Deploy

Graphical user interface, application

Description automatically generated

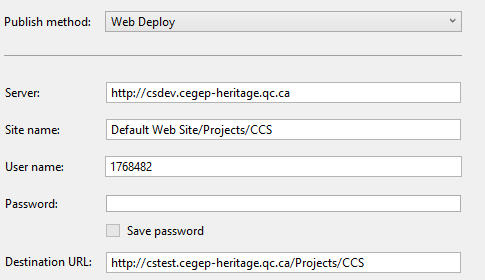
A screenshot of a computer

Description automatically generated with medium confidence

Graphical user interface, application

Description automatically generated

1. Create a publishing profile as per server to be deployed to, for example:



1. After finished, simply make sure the profile is selected and select publish.

Text

Description automatically generated

### IIS Configuration

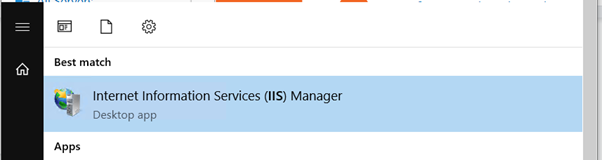
In order for authentication to work properly, the application pool needs to be configured on IIS.

1. Remote into the application server (CSDEV, CSTEST, CSPROD etc)

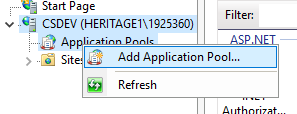
Graphical user interface, text, application

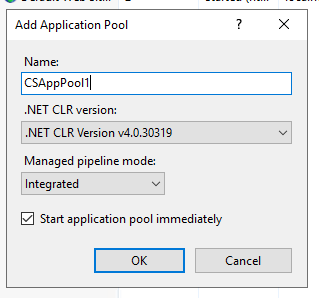
Description automatically generated

1. Navigate to IIS Server Manager



1. Create a new AppPool to use for the Application





1. Right-click on the Application and check the advanced settings.

Graphical user interface, application

Description automatically generated

1. Click the “…” on the application pool line.

Graphical user interface, application

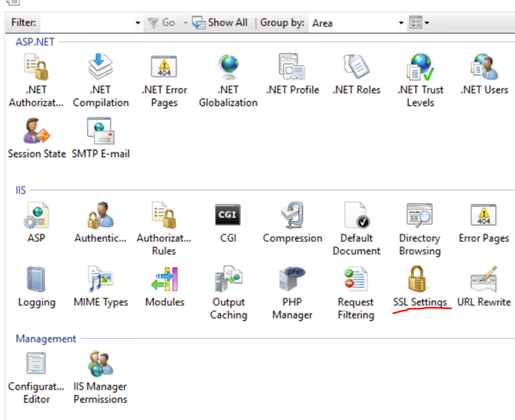
Description automatically generated

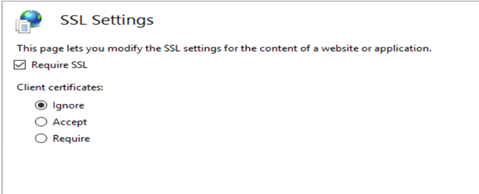
1. Select the CSAppPool you wish to use, (only for CCS) from the dropdown and select “Ok”.

Graphical user interface, text, application, email

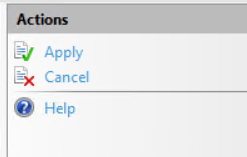
Description automatically generated

1. Select the Application and make sure the SSL Settings are checked to require SSL.



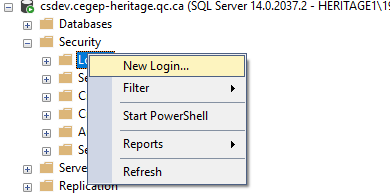


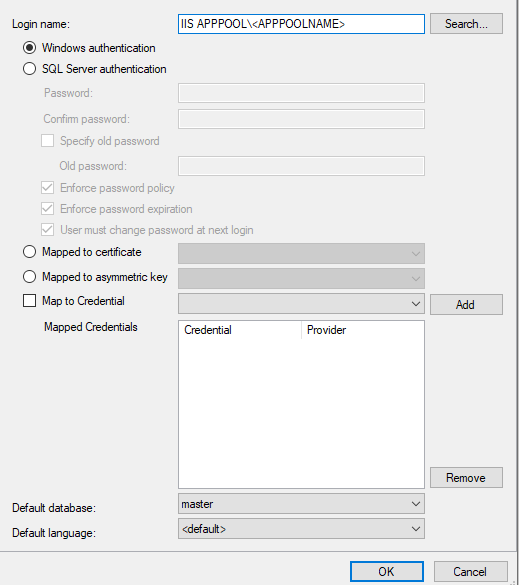
1. Click “Apply” on the right.



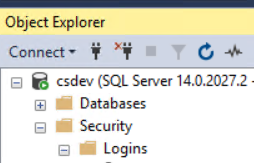
1. Open SSMS
2. Connect to the appropriate database server.
3. On SSMS, under Security > Logins, right-click on the IIS APPPOOL/ CSAppPool# user and click properties, if you can’t see the IIS APPPOOL you created earlier, you must add it as a new login.

How to add a new Login:

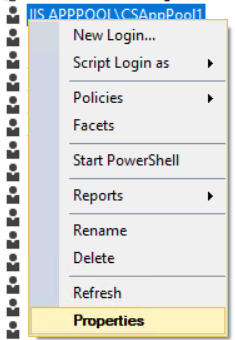




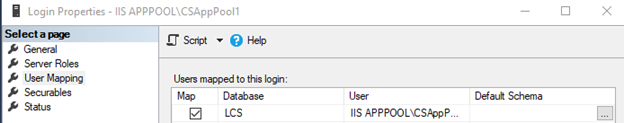
How to click on Properties:

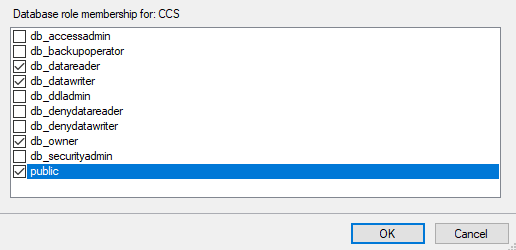


Inserting image...



1. Under User Mappings, select the Application database and assign the datawriter, datareader, and dbowner roles and click ‘Ok’.





1. You also need to give the same permissions for the master