

If you already have a Kentico web site or web application hosted on a traditional web server or prepared for deployment, you can decide to move your project to the Microsoft Azure Cloud Services. There are several tasks you need to perform before the deployment.

i This page provides instructions on converting an existing Kentico project to Kentico Azure project designed for deployment to **Azure Cloud Services**. If you want to deploy your Kentico project to Azure Web Apps, please see [Deploying existing Kentico projects to Azure Web Apps](#).

This scenario expects that you have a Kentico web site or web application project installed in a local file system, connected to a standard Microsoft SQL database. If you want to move your project to Azure Cloud Services, you need to move the media library files, deploy your database to Azure SQL (if you want to have the whole project on Azure, otherwise it is not necessary) and deploy the project files through Visual Studio.

However, please read the **Possible issues** section first and adjust your project according to the recommendations.

Possible issues

The following issues present problems which you should be aware of during the process of converting an existing project to an Azure project for deployment to Cloud Services. Please read through them, assess whether they apply to your project and adjust your project according to the recommendations.

- Storing **metafiles** and **attachments** in the file system (when the **Settings -> System -> Files -> Store files in: File system** setting is selected) is not recommended on the Azure platform. In this case you need to migrate the files to the database or the blob storage:
 - Store files only in the database - select the **Store files in: Database** setting in **Settings -> System -> Files**.
 - Or store such files in the Azure blob storage - select the **Store files in: File system** setting and move the files to the blob storage.
- In the Azure environment, **media files** are stored in the blob storage. After you move your existing media files to the Azure blob storage, the existing links to the media files from content will stop working. You will need to manually re-save them.
- If you store **virtual objects** (form layouts, page layouts, transformations, etc.) in the file system (when the **System -> Virtual objects -> Store all virtual objects in file system** option is enabled), you need to move the objects back to the database before deploying your project to the Azure platform. Virtual objects cannot be stored in the file system on Azure. However, you can store virtual objects in the file system in your local development project if you [develop the project as a web application](#).
- If you have set up a **web farm** in your project, the system can contain unnecessary web farm servers in the database. The web farm servers are generated automatically on Azure. Therefore, we recommend that you delete the old web farm servers in the UI to ensure that a licensing error does not occur.
- Review any **custom code** that works with the file system. Make sure that the code uses CMS.IO. Also migrate relevant files to the blob storage.
- Your **database** may contain custom database objects that are not supported on Azure SQL. This is a limitation of the Azure platform, you might not be able to migrate your database to Azure SQL without modifications. These issues might happen if you have somehow customized your database (stored procedures, indexes, access to tables) or if you have developed custom modules.

Prerequisites - Cloud environment

Before you start converting and deploying your project to the Azure platform, make sure that you have all the necessary services prepared in the [Azure Management Portal](#):

- [Create an Azure Storage](#) – the blob storage is necessary for storing media and other types of files.
- [Create an Azure SQL Server](#) and configure its firewall to accept your IP address – if you want to deploy your database to the Azure platform, you need to prepare an Azure SQL server. You do not need to prepare an empty database though, as the database will be created during the deployment process.
- [Create an Azure Cloud Service](#) – this service will host your project and provide computing power.

Part 1 - Move the media files

When the website runs in the Azure environment, media files are automatically stored in the blob storage. Therefore, we recommend that you move your existing media files to the blob storage as well to have all files in one place.

1. Open the created Azure blob storage.
 - You can use an arbitrary file manager for Microsoft Azure blob storage. We recommend [Cloudberry explorer for Microsoft Azure](#).
2. Create a new container in the root called **cmsstorage**.
3. Move all media library files to the created container including the site name folder.
 - The default location of all libraries of a particular site is `~/<sitename>/media`.
 - The folder structure on the blob storage will look like this: `cmsstorage/<sitename>/media/<medialibraryname>/<medialibraryfiles>`.
4. Convert any uppercase letters to lowercase, because Azure blob storage is case sensitive.

You have prepared your media library storage.

Part 2 - Deploy the database

You can host your project on Microsoft Azure and keep your database in a standard Microsoft SQL server. However, we recommend that you also move your database to Azure, as hosting your database on Azure ensures the best performance.

To learn how to deploy on-premise databases to Azure SQL Database, see the [SQL Server database migration to SQL Database in the cloud](#) article.

For example, you can use the following steps to deploy your database using the *Deploy Database to Microsoft Azure Database* wizard in SQL Server Management Studio:

1. Deploy your SQL server according to the instructions in the [SQL Server database migration to SQL Database in the cloud](#) article.
2. Change the connection string in the **web.config** file of your Kentico project to connect to the deployed Azure SQL database. You can use the following template:

```
<connectionStrings>
  <add name="CMSConnectionString" connectionString="Server=DATABASE_SERVER.
database.windows.net;Database=DATABASE_NAME;User ID=DB_SERVER_USER_NAME;
Password=DB_SERVER_PASSWORD;Trusted_Connection=False;Encrypt=True;Connection
Timeout=120;" />
</connectionStrings>
```

Replace the upper-case words with your own data.

Test if the database works as expected with your project. You now have a project that uses an Azure SQL database.

Part 3 - Convert the project

The best way of converting a normal project to an Azure project is to install a blank new Kentico Azure project and transfer your customizations.

1. [Install an empty Kentico Azure project](#) using the Kentico Installer.
2. Move all your customizations of the code base in your project to the new Kentico Azure project.
 - We recommend that you compare empty, not customized project with your project using some external compare tool.
3. Copy any additional files required by the site, such as custom user controls or stylesheet design files.
4. Include the copied files in your project:
 - a. Open Visual Studio and enable the **Show all files** button at the top of the Solution Explorer.

- b. Right click the given files and select **Include In Project**.
- c. Check the properties (**Right click -> Properties**) of all files that were included into the project and make sure that the value of their **Build Action** property is set to *Content*.

Your project is now converted to an Azure project.

Finish - Configure and deploy the Azure project

When your project is converted, transfer the listed settings from the web.config file to the ServiceConfiguration.Cloud.cscfg file and perform the described configurations. After that, your project is ready to be deployed to the cloud.

1. Make sure that you copy the value of the **CMSShashStringSalt** key from the web.config file in your original project to the web.config file in your Azure project.
 - If you do not copy this value, all macros and other features that use hashing will become invalid in the Azure project and you will have to re-save them.
2. Specify the connection string in the **ServiceConfiguration.Cloud.cscfg** file in the **CMSApp** and **SmartSearchWorker** sections. You can use the following template:

```
<ConfigurationSettings>
  <Setting name="CMSConnectionString" value="Server=DATABASE_SERVER.
database.windows.net;Database=DATABASE_NAME;User ID=DB_SERVER_USER_NAME;
Password=DB_SERVER_PASSWORD;Trusted_Connection=False;Encrypt=True;Connection
Timeout=120;" />
</ConfigurationSettings>
```

Replace the upper-case words with your own data.

3. Configure the blob storage settings and **Smart search** settings in the **ServiceConfiguration.Cloud.cscfg** file as specified in [Basic configuration](#).
4. When your project is configured, [deploy your Azure project](#).

Your whole project is now deployed to the Azure platform.

If you now wish to further develop your project locally, copy the settings to the **ServiceConfiguration.Local.cscfg** file and change the connection string to connect to a local database (instead of the production database on Azure). We also recommend that you prepare another Azure blob storage and connect it to your local project. See [Developing Kentico Azure projects locally](#) for more information. You can set up [Content staging](#) to synchronize the media files between the blob storages.

Repairing media library links and macro expressions

If you have created links from content (pages) to the media library files as [permanent links](#), they should work correctly even after deploying your project to the Azure environment.

However, if you have created them as direct links (with the file name in the URL), you need to correct them to point to the new address of the media files:



Media library links with Azure CDN

Configuring Azure CDN also changes the location of media library files. If you plan to configure the Azure CDN for your project, **re-save the media library links after you configure the CDN** for your project. Otherwise, you will have to re-save the links again.

- Open your Azure website and re-save the existing links to the media library files from content (pages). Correct links to the media files on a blob storage should begin with `/CMSPages/GetAzureFile.aspx`.

Also check that existing [macro expressions](#) work correctly.