BOE Pro

Azure Deployment Guide

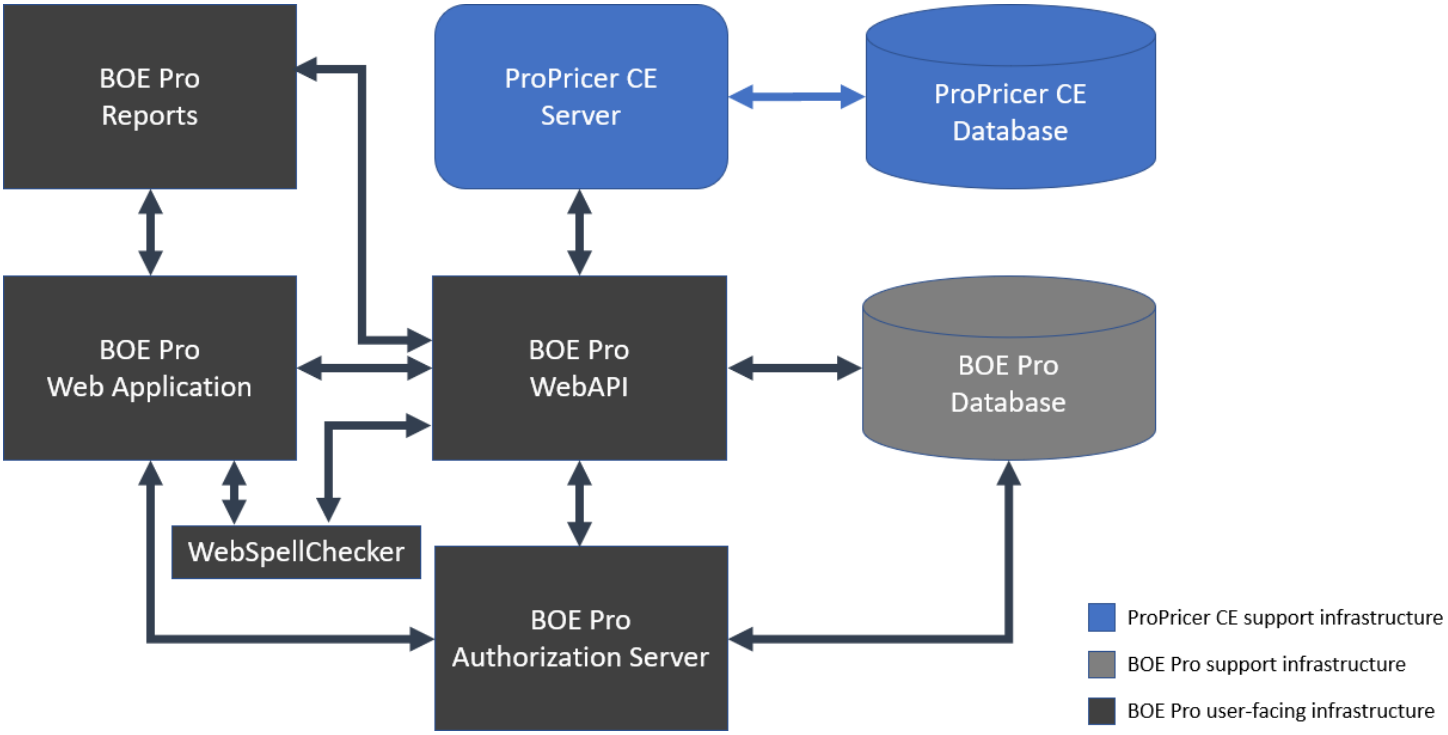
Revised July 27, 2023

Contents

Introduction

BOE Pro, an n-tier web application developed with ASP.NET Core, supports various deployments options. The focus of this guide is deployment on Microsoft Azure using App Service.

Architecture



Requirements and recommendations

Minimum requirements

- Azure account with required permissions to create App Services.
- Windows App Service Plan with Production or Isolated plan depending on your workload.
    - The Isolated plan is strongly recommended. It includes improved performance and security features, and is ideal for providing access to company-approved users only.
- Web Apps required:

App Name	.NET version (Runtime stack)
Authorization Server	.NET 6
Reports	.NET 6
WebAPI	.NET 6
Web Application	.NET 6

- SQL Server (Azure SQL Database or SQL Server 2016 or greater).
- Optional: SSL certificate or certificates for custom domains.
- Optional: PFX certificate for encryption of authentication tokens.
- Optional: Azure Application Registration to enable Azure Active Directory login and client certificate.
- Optional: Separated Linux App Service Plan for the WebSpellChecker component.

Known limitations

- BOE Pro fully supports Azure Active Directory logins, but Windows authentication is not supported when using Azure. However, BOE Pro supports Windows authentication when it is running on Windows Server.
- Bookmarking the URL of the Log In page can cause login issues for users. To quickly open BOE Pro in a browser, users should bookmark the web application start URL instead.

Self-contained deployment packages

All components, including the .NET 6 libraries and the .NET 6 runtime, come with the application, and are isolated from other .NET Core applications. Self-contained deployment packages include an executable.

Typically, each package will be a separate Azure App Services (Web App) in one and the same App Service Plan. Since the packages are separate self-contained deployments, you can deploy each one in a different App Service Plans.

> The ZIP packages are available in the ProPricer Support Portal.

BOE Pro Authorization Server

BOEPro_Authorization_[version]_win-x64.zip

BOE Pro Authorization Server provides a centralized login logic. This component requires a .NET 6 App Service.

BOE Pro WebAPI

BOEPro_WebAPI_[version]_win-x64.zip

Back-end BOE Pro Web API that receives requests from the BOE Pro web application, provides the WebSockets implementation for live collaboration updates, communicates with the database to persist the information, and links BOE Pro with your installation of ProPricer 9. This component requires a .NET 6 App Service.

BOE Pro Reports

BOEPro_Reports_[version]_win-x64.zip

BOE Pro report engine. Processes reporting requests from the Web Application and communicates with the WebAPI to retrieve the information. This component requires a .NET 6 App Service.

BOE Pro Web Application

BOEPro_WebApp_[version]_win-x64.zip

Front-end BOE Pro Web Application. This component uses only static files (JS, HTML, CSS, etc.). This component requires a .NET 6 App Service.

BOE Pro Database Setup

BOEPro_DatabaseSetup_[version]_win-x64.zip

Console application that creates and upgrades BOE Pro databases. This component uses .NET 6 single .exe deployment (.NET 6 contained in the .exe).

WebSpellChecker (Optional)

Image at Azure Container Registry https://propricer.azurecr.us

Docker image to provide Spell, Grammar, and Autocomplete in BOE Pro.

Create or upgrade a BOE Pro database

BOE Pro Database Setup is a command line tool that allows the database administrators to create and upgrade BOE Pro databases.

> The BOEProDatabaseSetup tool is contained in the WebAPI package, allowing you to use it from the **Console** option of the WebAPI App after deploying the WebAPI package.

Usage

`BOEProDatabaseSetup [options] [command]`

Options

`-v|--version` Show version information.

`-?|-h|--help` Show help information.

Commands

`addsysadmin` Add a system administrator account to a BOE Pro database.

`create` Create a new BOE Pro database.

`upgrade` Upgrade a BOE Pro database to the current version.

create command

Create a BOE Pro database.

Usage

```
BOEProDatabaseSetup create [options]
```

Options

`-?|-h|--help` Show help information.

`-f|--scripttofile <fileName>` Output the script to a file.

`-s|--server <servername>` Name of the database server.

`-w|--windowsauth` Use Windows authentication.

`-d|--dbname <databasename>` Name of the database. Default value is `BOEPro`.

`-al|--adminlogin <login>` BOE Pro admin user email. Default value is `sysadmin@propricer.com`.

`-ap|--adminpass <password>` BOE Pro admin user password. Default value is `sysadmin`.

`-an|--adminname <name>` BOE Pro admin username. Default value is `System Administrator`.

`-u|--dbauser <dbalogin>` Database user login.

`-p|--dbapass <dbapassword>` Database user password.

`-e|--useexistingdb` Use an existing database (for example, when using Azure SQL Database).

Examples

Create a BOE Pro database on an SQL Server on a VM using Windows Authentication:

```
BOEProDatabaseSetup create -s sqlserver.eastus.cloudapp.azure.com -d BOEPro -al
sysadmin@mycompany.com -ap MyStrongPassword4BOEPro -an "System Administrator" -w
```

Create a BOE Pro database on an Azure SQL Database using SQL Authentication:

```
BOEProDatabaseSetup create -s mycompany.database.windows.net -u myazureuser -p
MyAzurePassword -d BOEPro -al sysadmin@mycompany.com -ap MyStrongPassword4BOEPro
-an "System Administrator" -e
```

upgrade command

Upgrade an existing BOE Pro database.

Usage

```
BOEProDatabaseSetup upgrade [options]
```

Options

`-?|-h|--help` Show help information.

`-s|--server <servername>` Name of the database server.

`-w|--windowsauth` Use Windows authentication.

`-u|--dbauser <dbalogin>` Database user login.

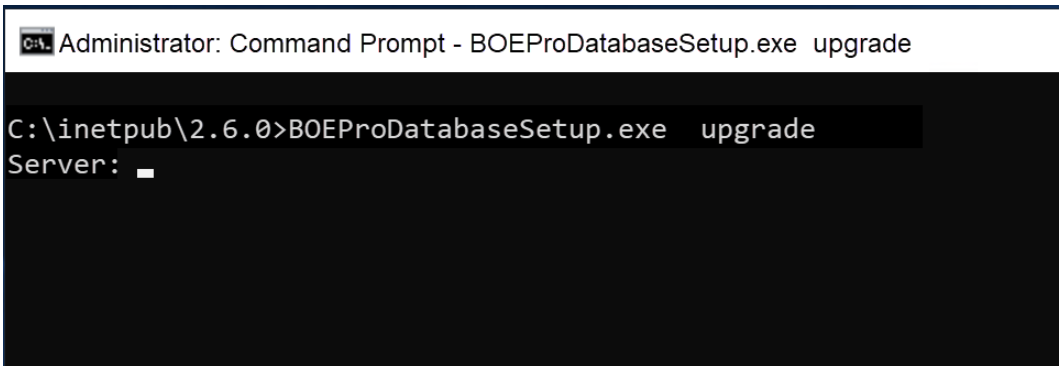
`-p|--dbapass <dbapassword>` Database user password.

`-d|--dbname <databasename>` Name of the database. Default value is `BOEPro`.

`-f|--scripttofile <fileName>` Output the script to a file.

Upgrade a BOE Pro database

1. Download the **BOEProDatabaseSetup** package.
2. Unzip **BOEProDatabaseSetup.exe**.
3. Open the **Command Prompt** window.
4. Go to the folder where **BOEProDatabaseSetup** is unzipped.
5. At the command prompt, use the **upgrade** command to upgrade your database:
   BOEProDatabaseSetup upgrade



6. If you need to use Windows Authentication, you can add **-w**:
   BOEProDatabaseSetup upgrade -w
7. Enter the information requested by the tool.

Examples

Upgrade a database named BOEPro on an SQL Server on VM using Windows Authentication:

```
BOEProDatabaseSetup upgrade -s sqlserver.eastus.cloudapp.azure.com -d BOEPro -w
```

Upgrade a database named BOEPro on an Azure SQL Database using SQL Authentication:

```
BOEProDatabaseSetup upgrade -s mycompany.database.windows.net -u myazureuser -p
MyAzurePassword -d BOEPro
```

addsysadmin command

Create an administrator-type user in a previously created BOE Pro database.

Usage

```
BOEProDatabaseSetup addsysadmin [options]
```

Options

`-?|-h|--help` Show help information.

`-s|--server <servername>` Name of the database server.

`-w|--windowsauth` Use Windows authentication.

`-d|--dbname <databasename>` Name of the database. Default value is `BOEPro.`

`-al|--adminlogin <login>` BOE Pro admin user email. Default value is `sysadmin@propricer.com`.

`-ap|--adminpass <password>` BOE Pro admin user password. Default value is `sysadmin`.

`-an|--adminname <name>` BOE Pro admin username. Default value is `System Administrator`.

`-u|--dbauser <dbalogin>` Database user login.

`-p|--dbapass <dbapassword>` Database user password.

Install BOE Pro Authorization Server

Download and unzip the BOE Pro Authorization self-deployment package.

> The ZIP packages are available in the ProPricer Support Portal.

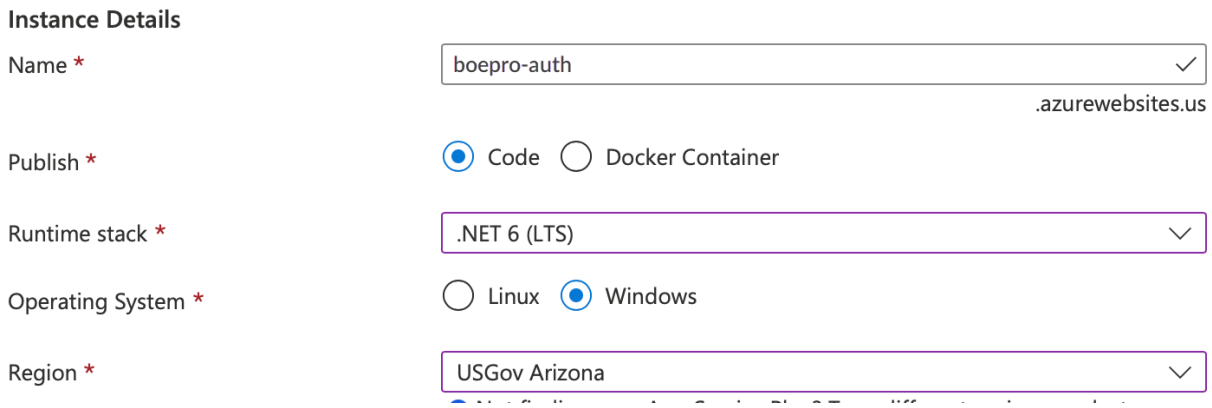
Prerequisites

Configuration

- BOE Pro Web App URL.
- BOE Pro Database Connection String.
- Optional: Application registration's Application ID (client), Directory ID (tenant) to enable Azure AD logins, and certificate private key (.pfx) for client certificate.

App Service

1. In the Azure Portal, add a Web App (App Service).
2. Enter the Web App name.
3. Select the following settings:
    a. **Publish**: Code
    b. **Runtime stack**: .NET 6 (LTS)
    c. **Operating System**: Windows
    d. **Region**: Select the desired region



Recommendation: Make sure you select the same Subscription, Resource group, and Region so you can select the same App Service Plan for all BOE Pro App Services.

4. Click **Next**.
5. On the **Monitoring** tab, make sure **Enable Application Insights** is set to **No**.
6. Click **Next**.
7. On the **Tags** tab, create the desired tags.
8. Click **Next**.
9. Click **Create**.

Register an application with the Microsoft Identity Platform

To enable Azure Active Directory logins, register an app in the Azure Portal so the Microsoft identity platform can provide authentication services for BOE Pro.

1. Sign into the Azure Portal.
2. If you have access to multiple tenants, use the **Directory +** subscription filter in the top menu to switch to the tenant in which you want to register the application.
3. Search for and select **Azure Active Directory**.
4. Under **Manage**, select **App registrations** > **New registration**.
5. Enter a display name for your application. For example, **BOE Pro**.
6. Specify who can use the application, sometimes called its sign-in audience. **Accounts in this organizational directory only** is recommended.
7. In **Redirect URI**, select the **Web** platform, and enter **<your-boepro-auth-server-url>/signin-oidc**. For example, **https://boepro-auth.azurewebsites.us/signin-oidc**
8. Click **Register**. Wait for the application registration creation.
9. Under **Manage**, select **Authentication**.
10. In **Front-channel logout URL**, enter **<your-boepro-auth-server-url>/signout-oidc**. For example, **https://boepro-auth.azurewebsites.us/signout-oidc**
11. Select **ID tokens (used for implicit and hybrid flows)**.
12. Click **Save**.
13. Under **Manage**, select **Token configuration**.
14. Click **Add optional claim.**
15. Under **Token type**, select **ID**.
16. Select **email**.
17. Click **Add**.
18. Go to **API registration** and click **Grant admin consent** for your tenant.

19. Under **Overview**, use **Application (client) ID** and **Directory (tenant) ID** in your Authorization Server configuration settings.

Reference: https://docs.microsoft.com/en-us/azure/active-directory/develop/quickstart-register-app

Add the Client/Desktop platform to the app registration

To enable Azure Active Directory logins when connecting to ProPricer 9, add the Client/Desktop platform to the app registration created in the previous procedure.

BOE Pro requires a client certificate to obtain the token used to log into ProPricer 9 with Azure Active Directory. You must have a valid certificate private key (.pfx).

> Make sure BOE Pro and ProPricer 9 are configured to use the same app registration.

1. Select the app registration created for BOE Pro from **App registrations**.
2. Under **Manage**, select **Authentication**.
3. Select **Add a platform**.
4. Select **Mobile and desktop applications**.
5. Select all redirect URI checkboxes, then click **Configure**.
6. Select **Access tokens (used for implicit flows)**.
7. (Optional) Clear **ID tokens (used for implicit and hybrid flows)** if selected.
8. Click **Save**.
9. Under **Manage**, select **Certificates & secrets**.
10. Under **Certificates**, select **Upload certificate**.
11. Select a certificate file, then click **Add**.

Configuration

There are two methods for configuring BOE Pro Authorization Server:

- Use the Manager tool, or edit **appsettings.json** in the target folder.
- Use the **Configuration** option in the **Settings** section of the App Service in the Azure Portal.

You can use either of these methods or a combination of both. The settings configured in the Azure Portal take precedence over the settings in the **appsettings.json** file.

The recommendation is to use the **Configuration** option in Azure to facilitate the upgrade process when a new version is released. As a minimum, use the Manager tool for most settings, and configure the database connection strings in the Azure Portal so this information is encrypted.

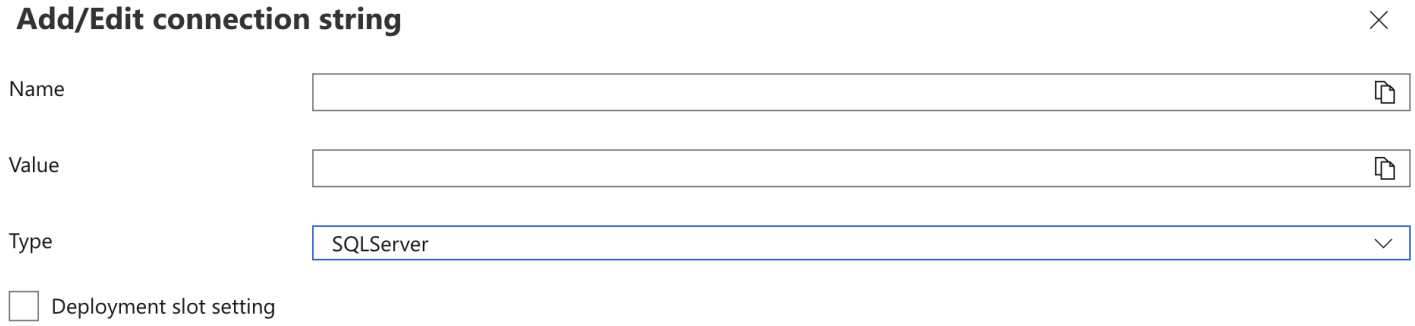
To use the Manager tool:

1. Open the **Command Prompt** window.
2. Go to the folder where BOE Pro Authorization Server is unzipped.
3. At the command prompt, enter: **BOEProAuthorizationServerManager.exe config**
4. When prompted for database information, press **Enter** to use default information.

To use App Service Configuration for BOE Pro Authorization Server:

1. In the App Service in the Azure Portal, go to the **Settings** section and select the **Configuration** option.

2. On the **General Settings** tab, click **New connection string** to create the following connection string:



Name	Value	Example
Name	EstimatorConnection	
Value	<your-database-connection-string>	Server=tcp:my.database.windows.us,1433;Database=boepro001;Persist Security Info=False;User ID=boeprodbo;Password={your-password};MultipleActiveResultSets=True; Encrypt=True; TrustServerCertificate=True; Connection Timeout=30;
Type	SQLServer	
Deployment slot setting	Unselected	

If you are using Azure SQL Database with the Azure Portal, go to the database > **Overview** section > **Show database connection strings**, then select the connection string on the **ADO.NET** tab. Make sure you replace **Initial Catalog** with **Database**.

3. Make sure the SQL Database server and App Service are connected to a Virtual Network (VNet) that allows SQL server traffic. Typically, the easiest is to add a subnet to the SQL Server VNet and link the App Service to the new subnet.

4. If you prefer not to use the Manager tool to store settings in the **appsettings.json** file, create the following application settings:

Name	Value	Example
Clients:0:AllowedCorsOrigins:0	<your-boepro-webapp-url>	https://boepro.azurewebsites.us
Clients:0:AllowedPostLogoutRedirectUrls:0	<your-boepro-webapp-url>/logout	https://boepro.azurewebsites.us/logout
Clients:0:AllowedRedirectUrls:0	<your-boepro-webapp-url>/callback	https://boepro.azurewebsites.us/callback
Clients:0:AllowedRedirectUrls:1	<your-boepro-webapp-url>/silent	https://boepro.azurewebsites.us/silent
Clients:0:AccessTokenLifetime	The access token lifetime in minutes. Logins are only valid within this time frame.	240
Host:Cors:AllowOrigin:0	<your-boepro-webapp-url>	https://boepro.azurewebsites.us
Options:HostDomain	<your-propricer-auth-server-host>	boepro-auth.azurewebsites.us
Options:HostScheme	https	
Options:RedirectUrl	<your-boepro-webapp-url>	https://boepro.azurewebsites.us

5. (Optional) To enable Azure Active Directory logins, click **New application setting** to create the following application settings:

Name	Value	Example
AzureAD:ClientId	<your-azure-application-client-id>	70b2caea-7e70-7212-7c32-a7cf1d47d5e7
AzureAD:TenantId	<your-azure-tenant-id>	80b2caea-8e70-7212-7c32-a7cf1d47d5e8

If you are upgrading from BOE Pro version 3.5.100.5 or earlier, the AzureActiveDirectory setting is now AzureAD.

BOE Pro requires an Application Registration to enable the Azure AD login option. See **Register an application with the Microsoft Identity Platform** to learn more.

6. If you are using a different Azure AD that is not a global service, such as Azure AD for US Government, click **New application setting** to create the following application settings:

Name	Value	Example (Azure AD for US Government)
AzureAD:Instance	<your-azure-AD-endpoint>	https://login.microsoftonline.us
Host:Cors:AllowOrigin:1	<your-azure-AD-endpoint>	https://login.microsoftonline.us

All the national clouds authenticate users separately in each environment and have separate authentication endpoints. See National clouds to learn more.

7. (Optional) To enable Azure AD logins when connecting to ProPricer 9, add the certificate private key (.pfx) of the client certificate configured in the app registration.

    a. Under **Settings**, select **Certificates**.

    b. Select the **Bring your own certificates (.pfx)** tab, then select **Add certificate.**

    c. Select the source of the certificate. Follow the instructions to **Validate** and add the certificate.

    d. Under **Settings**, select **Configuration** > **New application setting** to create the following application settings:

Name	Value	Example
AzureAD:ClientCertificates:0:SourceType	StoreWithThumbprint	StoreWithThumbprint
AzureAD:ClientCertificates:0:CertificateStorePath	CurrentUser/My	CurrentUser/My
AzureAD:ClientCertificates:0:CertificateThumbprint	<certificate thumbprint>	5A6AF846843AFA78FF939AE0F4B29A9BF6A517AF
WEBSITE_LOAD_CERTIFICATES	* or <certificate thumbprint>	5A6AF846843AFA78FF939AE0F4B29A9BF6A517AF

Reference: https://learn.microsoft.com/en-us/azure/active-directory/develop/msal-client-application-configuration

8. On the **General Settings** tab, verify or adjust the following settings:
    a. **Stack**: .NET
    b. **.NET Version**: .NET 6 (LTS)
    c. **Platform**: 64 Bit
    d. **Manage pipeline version**: Integrated
    e. **FTP state**: Disabled
    f. **HTTP version**: 1.1
    g. **Web sockets**: On
    h. **Always on**: On
    i. **ARR affinity**: Off
    j. **HTTPS Only**: On
    k. **Minimum TLS Version**: 1.2
    l. **Remote debugging**: Off
    m. **Client certificate mode**: Ignore
9. Click **Save**.

Application Logging

Enable application logging to collect diagnostic information from this web app. Logging is optional but highly recommended.

To enable application logging in Azure, create the following application settings in the **Configuration** option:

Name	Value
Serilog:WriteTo:1:Name	AzureApp
Serilog:WriteTo:1:Args:outputTemplate	{Message}{NewLine}|[{Url}] {UserAgent}{NewLine}{Exception}

To view logs in a log stream:

1. In the Azure Portal, go to **Monitoring** > **App Service logs**.
2. Enable **Application logging (Filesystem)**.
3. Set **Level** to **Information**.
4. Go to **Log stream** to see the log trace.

To preserve logs to a storage account:

1. In the Azure Portal, go to **Monitoring** > **App Service logs**.
2. Enable **Application logging (Blob)**.
3. Set **Level** to **Information**.
4. Select the **Storage Container** to store the logs.
5. Set the **Retention Period (Days)**.

Web Logging

Enable web server logging to collect diagnostic information from the web server. Logging is optional but highly recommended.

To view logs in a log stream:

1. In the Azure Portal, go to **Monitoring** > **App Service logs**.
2. Set **Web server logging** to **File System**.
3. Enter the **Quote (MB)**.
4. Set the **Retention Period (Days)**.

To preserve web logs:

1. In the Azure Portal, go to **Monitoring** > **App Service logs**.
2. Set **Web server logging** to **Storage**.
3. Select the **Storage** to send the logs to.
4. Set the **Retention Period (Days)**.

Recommended TLS/SSL settings

Azure App Service is created with an SSL certificate by default to provide https and a subdomain, like propricer9webapi.azurewebsites.us.

In the App Service in the Azure Portal, you should verify that the following TLS/SSL settings were selected during configuration:

- **HTTPS Only**: On
- **Minimum TLS Version**: 1.2

Optionally, you can configure your own domain in this section, like mysite.mycompany.com.

Deploy ZIP file using ZipDeployUI

This ZIP file deployment uses the same Kudu service that powers continuous integration-based deployments.

1. Zip the folder. Select all files (not the parent folder), right-click, point to **Send to**, select **Compress (zipped) folder**, then name the ZIP file.

> Deploy the ZIP file downloaded from the ProPricer Support Portal when all the settings are in the Configuration options of the App Service.

2. In the App Service options pane, select **Advanced Tools**, click **Go**, expand the **Tools** menu, then select **Zip Push Deploy**. Alternatively, in your browser, go to https://<app_name>.scm.azurewebsites.us/ZipDeployUI.

3. Upload the ZIP file by dragging it to the file explorer area on the web page.

When deployment is in progress, an icon in the top-right corner shows the progress percentage. The page also shows verbose messages for the operation below the explorer area. When it is finished, the last deployment message should say **Deployment successful**.

> Alternatively, use az webapp deployment source config-zip to deploy the ZIP file using Azure CLI, or the Publish-AzWebApp cmdlet to deploy the ZIP file using PowerShell.

Install BOE Pro Reports Server

Download and unzip the BOE Pro Reports Server self-deployment package.

> The ZIP packages are available in the ProPricer Support Portal.

Prerequisites

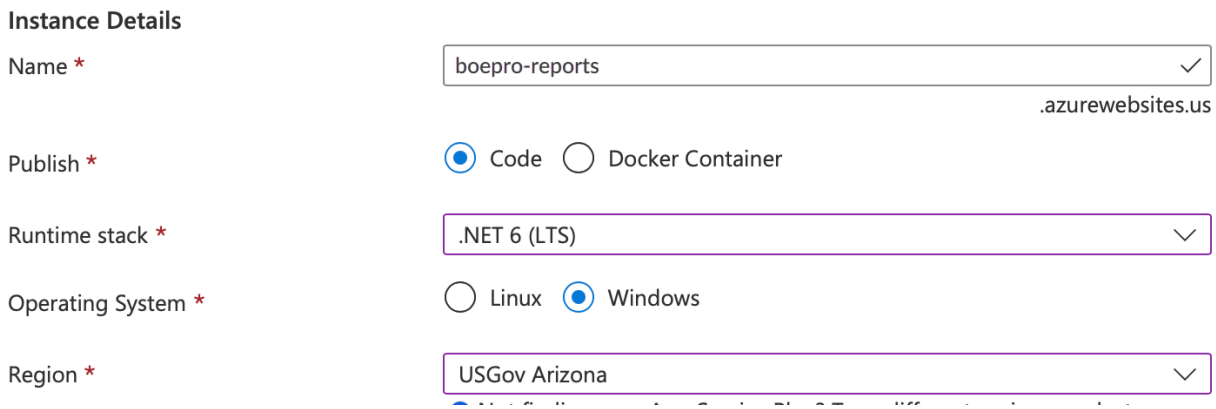
Configuration

- BOE Pro Web API URL.
- BOE Pro Web App URL.

App Service

1. In the Azure Portal, add a Web App (App Service).
2. Enter the Web App name.

3. Select the following settings:
    a. **Publish**: Code
    b. **Runtime stack**: .NET 6 (LTS)
    c. **Operating System**: Windows
    d. **Region**: Select the desired region



> **Recommendation**: Make sure you select the same Subscription, Resource group, and Region so you can select the same App Service Plan for all BOE Pro App Services.

4. Click **Next**.
5. On the **Monitoring** tab, make sure **Enable Application Insights** is set to **No**.
6. Click **Next**.
7. On the **Tags** tab, create the desired tags.
8. Click **Next**.
9. Click **Create**.

Configuration

There are two methods for configuring BOE Pro Reports Server:

- Use the Manager tool, or edit **appsettings.json** in the target folder.
- Use the **Configuration** option in the **Settings** section of the App Service in the Azure Portal.

You can use either of these methods or a combination of both. The settings configured in the Azure Portal take precedence over the settings in the **appsettings.json** file.

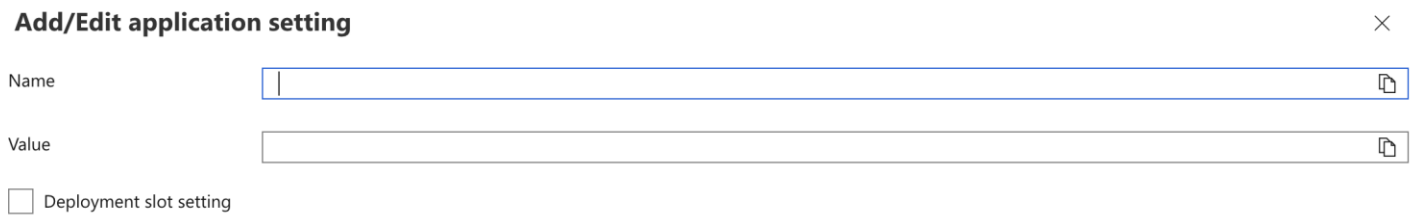
The recommendation is to use **Configuration** option in Azure to facilitate the upgrade process when new version is released.

To use the Manager tool:

1. Open the **Command Prompt** window.
2. Go to the folder where BOE Pro Reports is unzipped.
3. At the command prompt, enter: **BOEProReportsManager.exe config**

If you prefer not to use Manager tool to configure BOE Pro Reports Server:

1. In the App Service in the Azure Portal, go to the **Settings** section and select the **Configuration** option.
2. On the **General Settings** tab, click **New application setting** to create the following settings:



Name	Value	Example
DataSource:WebApiUrl	<your-boepro-webAPI-url>	https://boepro-api.azurewebsites.us
Host:Cors:AllowOrigin:0	<your-boepro-webapp-url>	https://boepro.azurewebsites.us

3. On the **General Settings** tab, verify or adjust the following settings:
    a. **Stack**: .NET
    b. **.NET Version**: .NET 6 (LTS)
    c. **Platform**: 64 Bit
    d. **Manage pipeline version**: Integrated
    e. **FTP state**: Disabled
    f. **HTTP version**: 1.1
    g. **Web sockets**: On
    h. **Always on**: On
    i. **ARR affinity**: Off
    j. **HTTPS Only**: On
    k. **Minimum TLS Version**: 1.2
    l. **Remote debugging**: Off
    m. **Client certificate mode**: Ignore
4. Click **Save**.
5. In the **API** section, select the **CORS** option, then set the following settings:
    a. Select the **Enable Access-Control-Allow-Credentials** option.
    b. In the **Allowed Origins** box, enter: **<your-boepro-webapp-url>**
6. Click **Save**.

Application Logging

Enable application logging to collect diagnostic information from this web app. Logging is optional but highly recommended.

To enable application logging in Azure, create the following application settings in the **Configuration** option:

Name	Value
Serilog:WriteTo:1:Name	AzureApp
Serilog:WriteTo:1:Args:outputTemplate	{Message}{NewLine}|[{Url}] {UserAgent}{NewLine}{Exception}

To view logs in a log stream:

1. In the Azure Portal, go to **Monitoring** > **App Service logs**.
2. Enable **Application logging (Filesystem)**.
3. Set **Level** to **Information**.
4. Go to **Log stream** to see the log trace.

To preserve logs to a storage account:

1. In the Azure Portal, go to **Monitoring** > **App Service logs**.
2. Enable **Application logging (Blob)**.
3. Set **Level** to **Information**.
4. Select the **Storage Container** to store the logs.
5. Set the **Retention Period (Days)**.

Web Logging

Enable web server logging to collect diagnostic information from the web server. Logging is optional but highly recommended.

To view logs in a log stream:

1. In the Azure Portal, go to **Monitoring** > **App Service logs**.
2. Set **Web server logging** to **File System**.
3. Enter the **Quote (MB)**.
4. Set the **Retention Period (Days)**.

To preserve web logs:

1. In the Azure Portal, go to **Monitoring** > **App Service logs**.
2. Set **Web server logging** to **Storage**.
3. Select the **Storage** to send the logs to.
4. Set the **Retention Period (Days)**.

Recommended TLS/SSL settings

Azure App Service is created with an SSL certificate by default to provide https and a subdomain, like propricer9webapi.azurewebsites.us.

In the App Service in the Azure Portal, you should verify that the following TLS/SSL settings were selected during configuration:

- **HTTPS Only**: On
- **Minimum TLS Version**: 1.2

Optionally, you can configure your own domain in this section, like mysite.mycompany.com.

Deploy ZIP file using ZipDeployUI

This ZIP file deployment uses the same Kudu service that powers continuous integration-based deployments.

1. Zip the folder. Select all files (not the parent folder), right-click, point to **Send to**, select **Compress (zipped) folder**, then name the ZIP file.

> Deploy the ZIP file downloaded from the ProPricer Support Portal when all the settings are in the Configuration options of the App Service.

2. In the App Service options pane, select **Advanced Tools**, click **Go**, expand the **Tools** menu, then select **Zip Push Deploy**. Alternatively, in your browser, go to https://<app_name>.scm.azurewebsites.us/ZipDeployUI.

3. Upload the ZIP file by dragging it to the file explorer area on the web page.

When deployment is in progress, an icon in the top-right corner shows the progress percentage. The page also shows verbose messages for the operation below the explorer area. When it is finished, the last deployment message should say **Deployment successful**.

> Alternatively, use az webapp deployment source config-zip to deploy the ZIP file using Azure CLI, or the Publish-AzWebApp cmdlet to deploy the ZIP file using PowerShell.

Install BOE Pro WebAPI Server

Download and unzip the BOE Pro WebAPI self-deployment package.

> The ZIP packages are available in the ProPricer Support Portal.

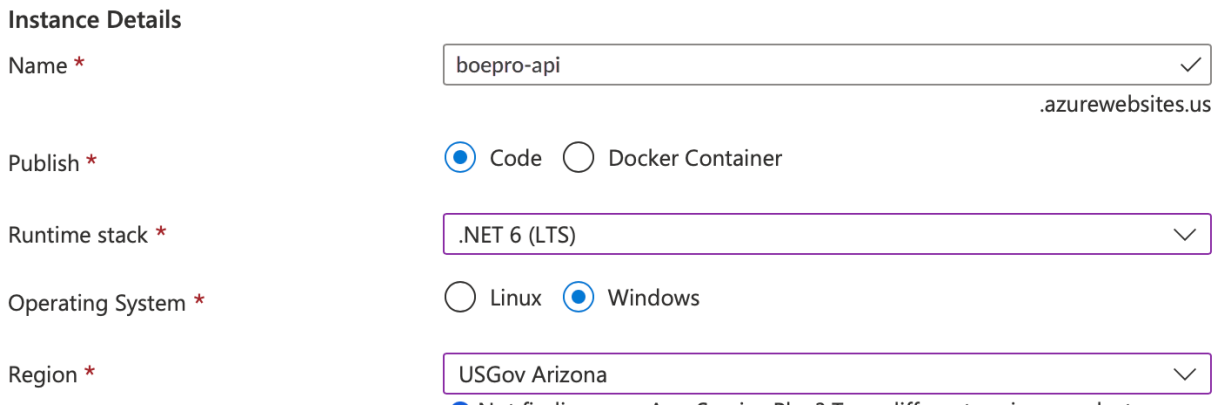
Prerequisites

Configuration

- ProPricer 9 Server host name and port.
- BOE Pro Web App URL.
- BOE Pro Authorization Server URL.
- BOE Pro Database Connection String.

App Service

1. In the Azure Portal, add a Web App (App Service).
2. Enter the Web App name.
3. Select the following settings:
    a. **Publish**: Code
    b. **Runtime stack**: .NET 6 (LTS)
    c. **Operating System**: Windows
    d. **Region**: Select the desired region



> **Recommendation**: Make sure you select the same Subscription, Resource group, and Region so you can select the same App Service Plan for all BOE Pro App Services.

4. Click **Next**.
5. On the **Monitoring** tab, make sure **Enable Application Insights** is set to **No**.
6. Click **Next**.
7. On the **Tags** tab, create the desired tags.
8. Click **Next**.
9. Click **Create**.

Configuration

There are two methods for configuring BOE Pro WebAPI:

- Use the Manager tool, or edit **appsettings.json** in the target folder.
- Use the **Configuration** option in the **Settings** section of the App Service in the Azure Portal.

You can use either of these methods or a combination of both. The settings configured in the Azure Portal take precedence over the settings in the **appsettings.json** file.

The recommendation is to use the **Configuration** option in Azure to facilitate the upgrade process when a new version is released. As a minimum, use the Manager tool for most settings, and configure the Database connection string in the Azure Portal so this information is encrypted.

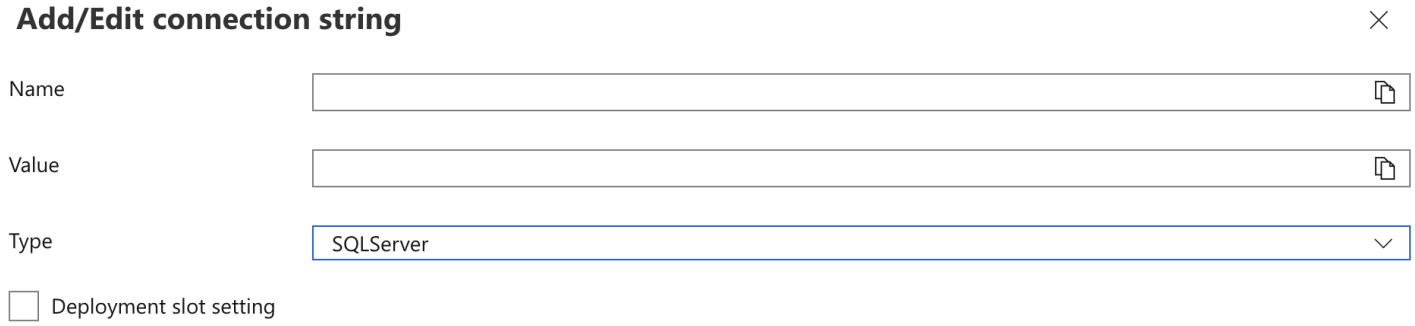
To use the Manager tool:

1. Open the **Command Prompt** window.
2. Go to the folder where BOE Pro WebAPI is installed.
3. At the command prompt, enter: **BOEProWebAPIManager.exe config**
4. When prompted for database information, press **Enter** to use default information.

To finish configuration for BOE Pro WebAPI:

1. In the App Service in the Azure Portal, go to the **Settings** section and select the **Configuration** option.

2. On the **General Settings** tab, click **New connection string** to create the following connection string:



Name	Value	Example
Name	EstimatorConnection	
Value	<your-database-connection-string>	Server=tcp:my.database.windows.us,1433;Database=boepro001;Persist Security Info=False;User ID=boeprodbo;Password={your-password};MultipleActiveResultSets=True;Encrypt=True; TrustServerCertificate=True;Connection Timeout=30;
Type	SQLServer	
Deployment slot setting	Unselected	

If you are using Azure SQL Database with the Azure Portal, go to the database > **Overview** section > **Show database connection strings**, then select the connection string on the **ADO.NET** tab. Make sure you replace **Initial Catalog** with **Database**.

3. Make sure SQL Database server and App Service are connected to a Virtual Network (VNet) that allows SQL server traffic. Typically, the easiest is to add a subnet to the SQL server vnet and link the App Service to the new subnet.

4. If you prefer not to use Manager tool to store settings in **appsettings.json** file, create the following application settings:

Name	Value	Example
Host:Cors:AllowOrigin:0	<your-boepro-webapp-url>	https://boepro.azurewebsites.us
Security:Authority	<your-propricer-boepro-authorization-server-url>	https://boepro-auth.azurewebsites.us
Data:ACLayer:ConnectionNameRegExPattern	Regular expression to filter connection names you want to display in BOE Pro. . (period) is the default value and represents all characters (no filter).	ProPricer_Prod_SQL
PP9Settings:ServerName	Your ProPricer 9 Application Server hostname	propricer9.mycompany.us
PP9Settings:ServerPort	Your ProPricer 9 Application Server port	8092

5. On the **General Settings** tab, verify or adjust the following settings:
    a. **Stack**: .NET
    b. **.NET Version**: .NET 6 (LTS)
    c. **Platform**: 64 Bit
    d. **Manage pipeline version**: Integrated
    e. **FTP state**: Disabled
    f. **HTTP version**: 1.1
    g. **Web sockets**: On
    h. **Always on**: On
    i. **ARR affinity**: Off
    j. **HTTPS Only**: On
    k. **Minimum TLS Version**: 1.2
    l. **Remote debugging**: Off
    m. **Client certificate mode**: Ignore
6. Click **Save**.

Application Logging

Enable application logging to collect diagnostic information from this web app. Logging is optional but highly recommended.

To enable application logging in Azure, create the following application settings in the **Configuration** option:

Name	Value
Serilog:WriteTo:1:Name	AzureApp
Serilog:WriteTo:1:Args:outputTemplate	{Message}{NewLine}|[{Url}] {UserAgent}{NewLine}{Exception}

To view logs in a log stream:

1. In the Azure Portal, go to **Monitoring** > **App Service logs**.
2. Enable **Application logging (Filesystem)**.
3. Set **Level** to **Information**.
4. Go to **Log stream** to see the log trace.

To preserve logs to a storage account:

1. In the Azure Portal, go to **Monitoring** > **App Service logs**.
2. Enable **Application logging (Blob)**.
3. Set **Level** to **Information**.
4. Select the **Storage Container** to store the logs.
5. Set the **Retention Period (Days)**.

Web Logging

Enable web server logging to collect diagnostic information from the web server. Logging is optional but highly recommended.

To view logs in a log stream:

1. In the Azure Portal, go to **Monitoring** > **App Service logs**.
2. Set **Web server logging** to **File System**.
3. Enter the **Quote (MB)**.
4. Set the **Retention Period (Days)**.

To preserve web logs:

1. In the Azure Portal, go to **Monitoring** > **App Service logs**.
2. Set **Web server logging** to **Storage**.
3. Select the **Storage** to send the logs to.
4. Set the **Retention Period (Days)**.

Recommended TLS/SSL settings

Azure App Service is created with an SSL certificate by default to provide https and a subdomain, like propricer9webapi.azurewebsites.us.

In the App Service in the Azure Portal, you should verify that the following TLS/SSL settings were selected during configuration:

- **HTTPS Only**: On
- **Minimum TLS Version**: 1.2

Optionally, you can configure your own domain in this section, like mysite.mycompany.com.

Deploy ZIP file using ZipDeployUI

This ZIP file deployment uses the same Kudu service that powers continuous integration-based deployments.

1. Zip the folder. Select all files (not the parent folder), right-click, point to **Send to**, select **Compress (zipped) folder**, then name the ZIP file.

> Deploy the ZIP file downloaded from the ProPricer Support Portal when all the settings are in the Configuration options of the App Service.

2. In the App Service options pane, select **Advanced Tools**, click **Go**, expand the **Tools** menu, then select **Zip Push Deploy**. Alternatively, in your browser, go to https://<app_name>.scm.azurewebsites.us/ZipDeployUI.

3. Upload the ZIP file by dragging it to the file explorer area on the web page.

When deployment is in progress, an icon in the top-right corner shows the progress percentage. The page also shows verbose messages for the operation below the explorer area. When it is finished, the last deployment message should say **Deployment successful**.

> Alternatively, use az webapp deployment source config-zip to deploy the ZIP file using Azure CLI, or the Publish-AzWebApp cmdlet to deploy the ZIP file using PowerShell.

Install WebSpellChecker (Optional)

Installing WebSpellChecker is completely optional. Before installing, ensure that you have or can obtain a Linux App Service Plan to support it.

The WebSpellChecker docker image is located in the Azure Government Container Registry at https://propricer.azurecr.us.

> Contact Technical Support to obtain the login credentials.

Prerequisites

Configuration

- Linux App Service Plan with a minimum of 7 GB of RAM. Production level as minimum. Isolated level strongly recommended.
- ProPricer Container Registry login credentials.

App Service

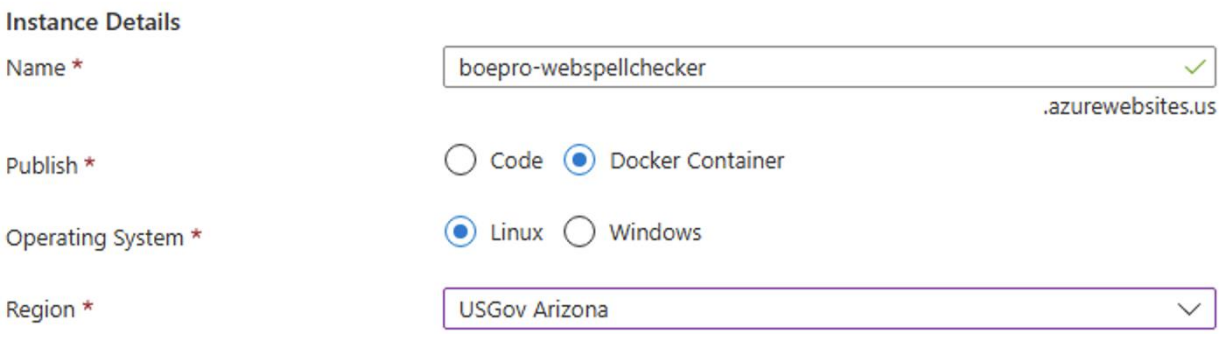
1. In the Azure Portal, add a Web App (App Service).
2. Enter the Web App name.

3. Select the following settings:

    a. **Publish**: Docker Container

    b. **Operating System**: Linux

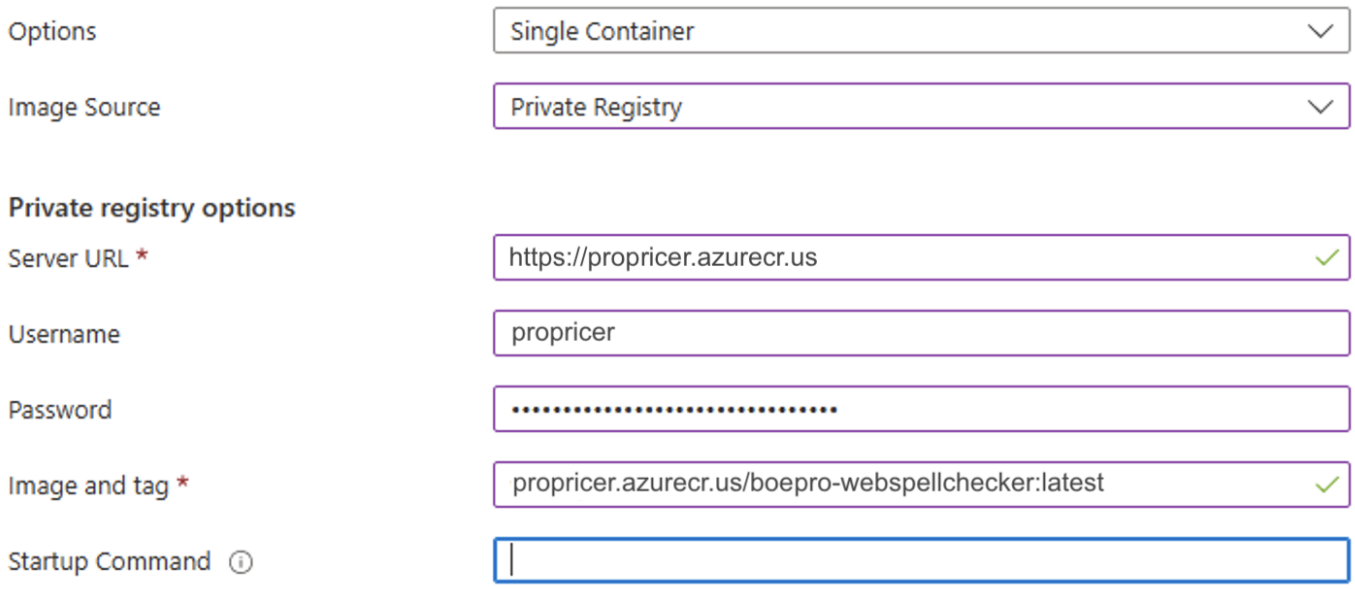
    c. **Region**: Select the desired region



Recommendation: Select the same Region as your BOE Pro App Service Plan.

4. Click **Next**.

5. On the **Docker** tab, enter the following:
    a. **Options**: Single Container
    b. **Image Source**: Private Registry
    c. **Server URL**: https://propricer.azurecr.us
    d. **Username**: propricer
    e. **Password**: Contact ProPricer Technical Support to obtain the password
    f. **Image and tag**: propricer.azurecr.us/boepro-webspellchecker:latest
    g. **Startup Command**: leave blank



6. Click **Next**.
7. On the **Networking** tab, set the appropriate configuration for your environment. Typically the network injection is enabled, and it should allow communication from WebAPI and your users' browsers.
8. Click **Next**.
9. On the **Monitoring** tab, make sure **Enable Application Insights** is set to **No**.
10. Click **Next**.

11. On the **Tags** tab, create the desired tags.

12. Click **Next**.

13. Click **Create**.

Configuration

To use user- and company-level custom dictionaries, your need to share a directory for the dictionaries with the Docker container. You can use the `/home` directory in your custom container file system to persist files across restarts and share them across instances. The `/home` directory is provided to enable your custom container to access persistent storage.

To use App Service Configuration for WebSpellChecker:

1. In the App Service in the Azure Portal, go to the **Settings** section and select the **Configuration** option.
2. On the **Application Settings** tab, make sure the setting **WEBSITES_ENABLE_APP_SERVICE_STORAGE** is set to **true**.
3. On the **General Settings** tab, verify or adjust the following settings:
    a. **Startup Command**: leave blank
    b. **FTP state**: Disabled
    c. **HTTP version**: 1.1
    d. **HTTP 2.0 Proxy**: Off
    e. **Always on**: On
    f. **ARR affinity**: Off
    g. **HTTPS Only**: On
    h. **Minimum TLS Version**: 1.2
    i. **Client certificate mode**: Ignore
4. Click **Save**.

Application Logging

Enable application logging to collect diagnostic information from this web app. Logging is optional but highly recommended.

To enable application logging in Azure:

1. In the Azure Portal, go to **Monitoring** > **App Service logs**.
2. Enable **Application logging (Filesystem)**.
3. Set **Quota (MB)**.
4. Set the **Retention Period (Days)**.

Recommended TLS/SSL settings

Azure App Service is created with an SSL certificate by default to provide https and a subdomain, like boepro-webspellchecker.azurewebsites.us.

In the App Service in the Azure Portal, you should verify that the following TLS/SSL settings were selected during configuration:

- **HTTPS Only**: On
- **Minimum TLS Version**: 1.2

Optionally, you can configure your own domain in this section, like mysite.mycompany.com.

Install BOE Pro Web Application

Download and unzip the BOE Pro Web Application package.

> The ZIP packages are available in the ProPricer Support Portal.

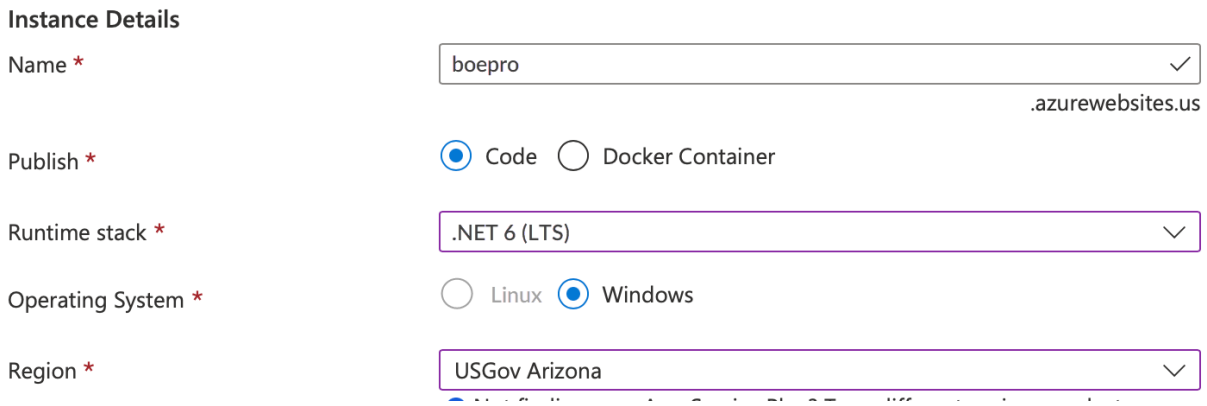
Prerequisites

Configuration

- BOE Pro Web API URL.
- BOE Pro Reports URL.
- BOE Pro Authorization Server URL.

App Service

1. In the Azure Portal, add a Web App (App Service).
2. Enter the Web App name.
3. Select the following settings:
    a. **Publish**: Code
    b. **Runtime stack**: .NET 6 (LTS)
    c. **Operating System**: Windows
    d. **Region**: Select the desired region



> **Recommendation**: Make sure you select the same Subscription, Resource group, and Region so you can select the same App Service Plan for all BOE Pro App Services.

4. Click **Next**.
5. On the **Monitoring** tab, make sure **Enable Application Insights** is set to **No**.
6. Click **Next**.
7. On the **Tags** tab, create the desired tags.
8. Click **Next**.
9. Click **Create**.

Configuration

You can configure BOE Pro Web Application with the Manager tool, which is recommended, or by editing **web.config** in the target folder.

To use the Manager tool:

1. Open the **Command Prompt** window.
2. Go to the folder where BOE Pro Web Application is unzipped.
3. At the command prompt, enter: **BOEProWebAppManager.exe config**

> The Application settings in the Configuration option in Azure are not supported by BOE Pro Web Application.

If you installed WebSpellChecker, to enable in BOE Pro:

- At the command prompt, enter: **BOEProWebAppManager.exe EnableWebSpellChecker**

Microsoft Clarity (Optional)

Microsoft Clarity is a user behavior analytics tool that helps you understand how people are interacting with your website. It uses features such as session replays and heatmaps.

Enabling Microsoft Clarity is completely optional. Be aware that BOE Pro and Executive Business Services, Inc. receive no data from Clarity, and that Microsoft collects all Clarity data. Before enabling, ensure that this is not a security problem for your organization.

To enable Microsoft Clarity in BOE Pro:

1. Sign in to https://clarity.microsoft.com.
2. Create a new Clarity project for BOE Pro.
3. Go to **Settings** > **Overview** and copy the Project ID.
4. At the command prompt, enter: **BOEProWebAppManager.exe EnableClarity**
5. When prompted, enter the Project ID.

To finish configuration for BOE Pro Web Application:

1. In the App Service in the Azure Portal, go to the **Settings** section and select the **Configuration** option.
2. On the **General Settings** tab, verify or adjust the following settings:
    a. **Stack**: .NET
    b. **.NET Version**: .NET 6 (LTS)
    c. **Platform**: 64 Bit
    d. **Manage pipeline version**: Integrated
    e. **FTP state**: Disabled
    f. **HTTP version**: 1.1
    g. **Web sockets**: Off
    h. **Always on**: On
    i. **ARR affinity**: Off
    j. **HTTPS Only**: On
    k. **Minimum TLS Version**: 1.2
    l. **Remote debugging**: Off
    m. **Client certificate mode**: Ignore
3. Click **Save**.

Web Logging

Enable web server logging to collect diagnostic information from the web server. Logging is optional but highly recommended.

> Application logging does not apply to BOE Pro Web Application.

To view logs in a log stream:

1. In the Azure Portal, go to **Monitoring** > **App Service logs**.
2. Set **Web server logging** to **File System**.
3. Enter the **Quote (MB)**.
4. Set the **Retention Period (Days)**.

To preserve web logs:

1. In the Azure Portal, go to **Monitoring** > **App Service logs**.
2. Set **Web server logging** to **Storage**.
3. Select the **Storage** to send the logs to.
4. Set the **Retention Period (Days)**.

Recommended TLS/SSL settings

Azure App Service is created with an SSL certificate by default to provide https and a subdomain, like propricer9webapi.azurewebsites.us.

In the App Service in the Azure Portal, you should verify that the following TLS/SSL settings were selected during configuration:

- **HTTPS Only**: On
- **Minimum TLS Version**: 1.2

Optionally, you can configure your own domain in this section, like mysite.mycompany.com.

Deploy ZIP file using ZipDeployUI

This ZIP file deployment uses the same Kudu service that powers continuous integration-based deployments.

1. Zip the folder. Select all files (not the parent folder), right-click, point to **Send to**, select **Compress (zipped) folder**, then name the ZIP file.

> Deploy the ZIP file downloaded from the ProPricer Support Portal when all the settings are in the Configuration options of the App Service.

2. In the App Service options pane, select **Advanced Tools**, click **Go**, expand the **Tools** menu, then select **Zip Push Deploy**. Alternatively, in your browser, go to https://<app_name>.scm.azurewebsites.us/ZipDeployUI.

3. Upload the ZIP file by dragging it to the file explorer area on the web page.

When deployment is in progress, an icon in the top-right corner shows the progress percentage. The page also shows verbose messages for the operation below the explorer area. When it is finished, the last deployment message should say **Deployment successful**.

> Alternatively, use az webapp deployment source config-zip to deploy the ZIP file using Azure CLI, or the Publish-AzWebApp cmdlet to deploy the ZIP file using PowerShell.

Remove ProPricer 9 WebAPI

BOE Pro v3.5.100.3 and later implement communication with ProPricer 9 Server in the BOE Pro WebAPI. This makes the ProPricer 9 WebAPI web app no longer necessary. If you have BOE Pro v3.5.100.2 or earlier, delete the ProPricer 9 WebAPI web app after installing BOE Pro v3.5.100.3 or later.

> ProPricer 9 Server Host Name and Port settings are now required on BOE Pro WebAPI configuration settings.

Upgrade from 3.5.100.x to 3.5.101.0

If you configure BOE Pro to use Azure AD logins, you need to change all configuration keys starting with **AzureActiveDirectory** to **AzureAD** starting in BOE Pro version 3.5.101.0.

References

- Azure App Service
  https://azure.microsoft.com/en-us/services/app-service
- Deploy your app to Azure App Service with a ZIP or WAR file
  https://docs.microsoft.com/en-us/azure/app-service/deploy-zip
- National clouds
  https://docs.microsoft.com/en-us/azure/active-directory/develop/authentication-national-cloud#azure-ad-authentication-endpoints
- Quickstart: Register an application with the Microsoft identity platform
  https://docs.microsoft.com/en-us/azure/active-directory/develop/quickstart-register-app
- Configure a custom container for Azure App Service
  https://learn.microsoft.com/en-us/azure/app-service/configure-custom-container?tabs=debian&pivots=container-linux
- Microsoft Clarity
  https://clarity.microsoft.com