Deltek ProPricer

BOE Pro Azure Deployment Guide

Contents

Introduction 1
Architecture 1
BOE Pro 2
Minimum requirements 2
Known limitations 2
Deployment package 3
BOE Pro WebAPI 3
BOE Pro Reports 3
BOE Pro WebApp 4
WebSpellChecker (Optional) 4
Create a BOE Pro database 5
Create BOE Pro Reports Server 6
Prerequisites 6
Configuration 6
App Service 6
Configuration 8
Application Logging 10
Web Logging 11

Recommended TLS/SSL settings 11
Deploy ZIP file using ZipDeployUI 12
Create BOE Pro WebAPI Server 13
Prerequisites 13
Configuration 13
App Service 14
Configuration 15
Application Logging 20
Web Logging 21
Recommended TLS/SSL settings 21
Deploy ZIP file using ZipDeployUI 22
Install WebSpellChecker (Optional) 23
Prerequisites 26
Configuration 26
App Service 27
Configuration 29
Application Logging 30
Recommended TLS/SSL settings 30
Create BOE Pro Web Application 31
Prerequisites 31
Configuration 31

App Service 32
Configuration 33
Microsoft Clarity (Optional) 33
Web Logging 35
Recommended TLS/SSL settings 35
Deploy ZIP file using ZipDeployUI 36
Remove BOE Pro Authorization Server 37
Remove ProPricer 9 WebAPI 38
Upgrade from 3.5.100.x 39
Upgrade to 3.5.104.2 (.NET 8) 40
Upgrade to 3.5.105.1 (.NET 8) 41
Migrate users 41
Upgrade BOE Pro 41
Troubleshoot 42
Authorization Server Pro, BOE Pro WebAPI Server, and BOE Pro Reports Server – Fetch error 42
Solution 1 43
Solution 2 43
Solution 3 43
Solution 4 43

Database Setup reference 44

create command 45

upgrade command 46

Upgrade a BOE Pro database 46

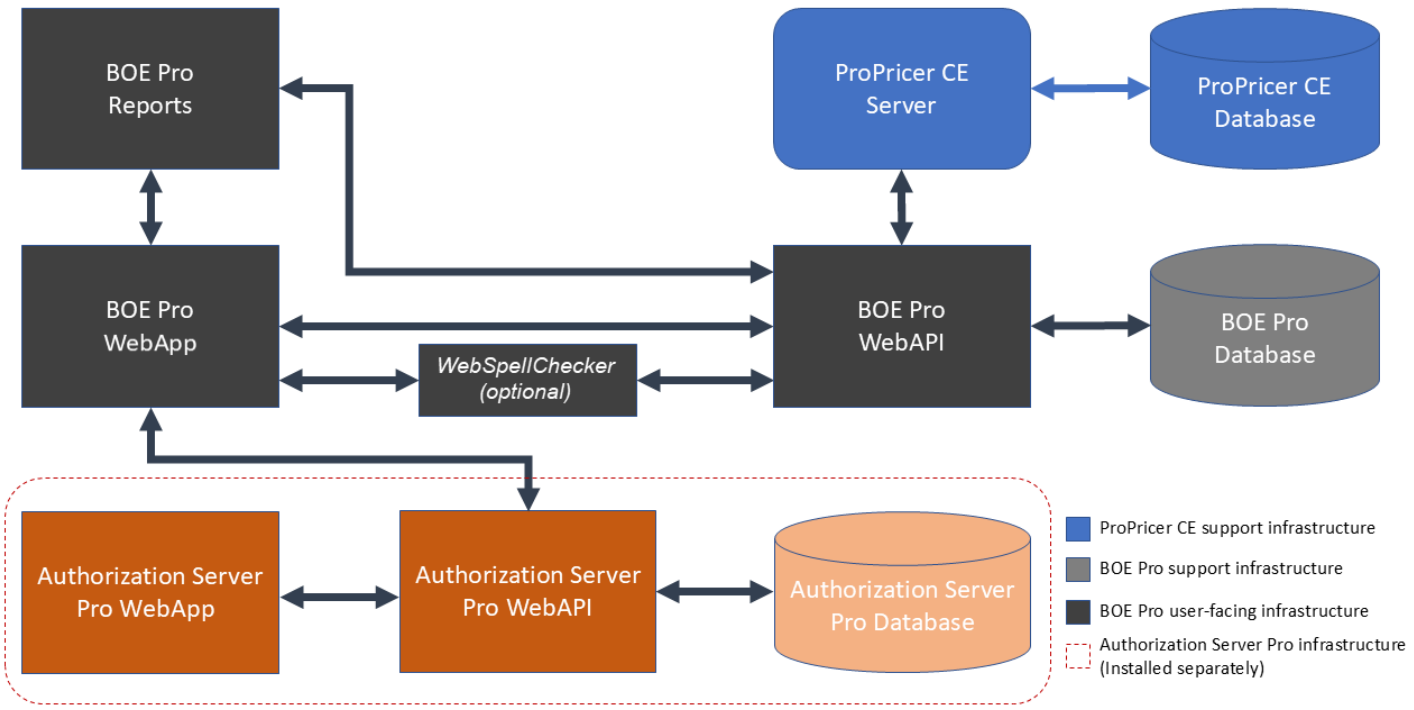
addsysadmin command 48

References 49

Introduction

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

Architecture



BOE Pro

Minimum requirements

- Azure account with required permissions to create App Services.
- Windows App Service Plan with the Premium 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 applications required:

App Name	.NET version (Runtime stack)
Reports	.NET 8
WebAPI	.NET 8
WebApp	.NET 8

- SQL Server (Azure SQL Database or SQL Server 2016 or newer).
- Compatible Authorization Server Pro instance installed.
- Optional: SSL certificate or certificates for custom domains.
- Optional: Azure Application Registration to enable Azure Active Directory login and client certificate.
- Optional: Separate 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.
- BOE Pro v3.5.103.3 WebAPI implements background processing for some processes, but it relies on the memory of WebAPI, so it does not work properly on a multi-instance App Service plan. You can host the other web applications on a multi-instance App Service plan, but not WebAPI.

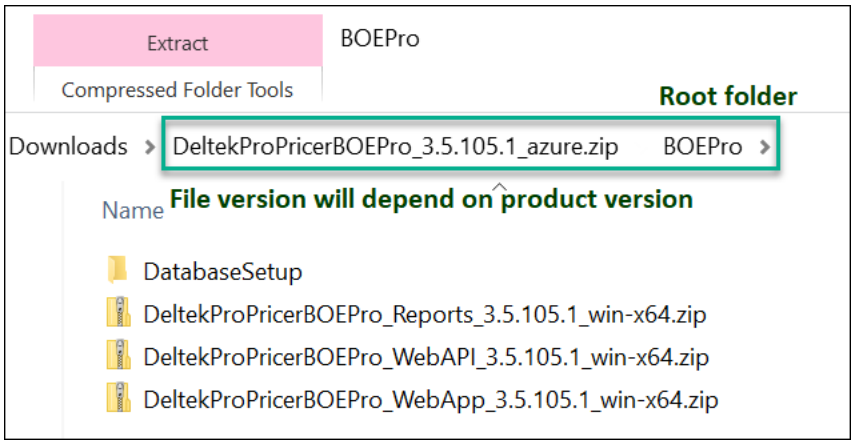
Deployment package

The platform-specific and framework-dependent package include only the application and its dependencies. The .NET runtime is provided by Azure Web App. Deployment packages include a Windows x64 platform-specific executable.

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

The ZIP package is available in the Deltek Software Manager (DSM).

The file name of the ZIP package is **DeltekProPricer_BOEPro_[version]_azure.zip**. Download and extract it to a temporary folder, there you should find the following content:



BOE Pro WebAPI

DeltekProPricerBOEPro_WebAPI_[version]_win-x64.zip (extracted from ZIP package)

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 8 App Service.

BOE Pro Reports

DeltekProPricerBOEPro_Reports_[version]_win-x64.zip (extracted from ZIP package)

BOE Pro report engine. Processes reporting requests from the web application and communicates with the web API to retrieve the information. This component requires a .NET 8 App Service.

BOE Pro WebApp

DeltekProPricerBOEPro_WebApp_[version]_win-x64.zip (extracted from ZIP package)

Front-end BOE Pro web application. This component uses only static files such as JS, HTML, CSS, etc. (the files are included in the folder). It requires a .NET 8 App Service.

WebSpellChecker (Optional)

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

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

Create a BOE Pro database

You must begin setting up BOE Pro by creating your BOE Pro database. BOE Pro Database Setup is a command line tool that allows database administrators to create and upgrade BOE Pro databases.

1. Open a command prompt window and go to the folder where **BOEProDatabaseSetup.exe** was extracted.
2. Run the tool with the `create` command depending on the authentication method.
    - To use SQL authentication over Azure SQL, run the tool with the `create` command and follow the prompts (database must already exist): `BOEProDatabaseSetup create -e`
    - To use SQL authentication, run the tool with the `create` command and follow the prompts: `BOEProDatabaseSetup create`
    - To use Windows authentication, run the tool with the `create` command, add the `-w` parameter, and follow the prompts: `BOEProDatabaseSetup create -w`

See the BOE Pro Database Setup reference section for further guidance.

Create BOE Pro Reports Server

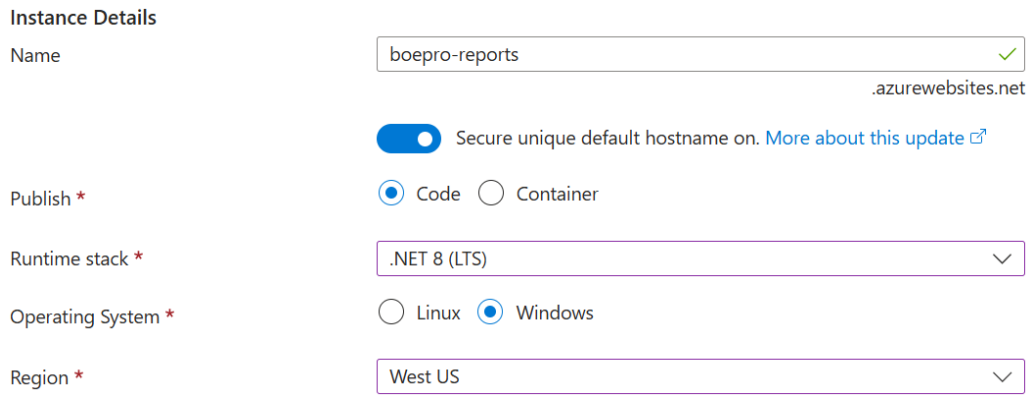
Prerequisites

Configuration

- BOE Pro WebAPI URL.
- BOE Pro WebApp 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:
    - **Publish**: Code
    - **Runtime stack**: .NET 8 (LTS)
    - **Operating System**: Windows
    - **Region**: Select the desired region



***Recommendation**: Select the same subscription, resource group, and region so you can select the same App Service Plan for all BOE Pro App Services.*

4. On the **Deployment** tab, set **Continuous deployment** to **Disable**.
5. On the **Networking** tab, set **Enable public access** to **On**, and **Enable network injection** to **Off**.
6. On the **Monitoring** tab, set **Enable Application Insights** to **No**.
7. On the **Tags** tab, create the desired tags.
8. On the **Review + create** tab, 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.

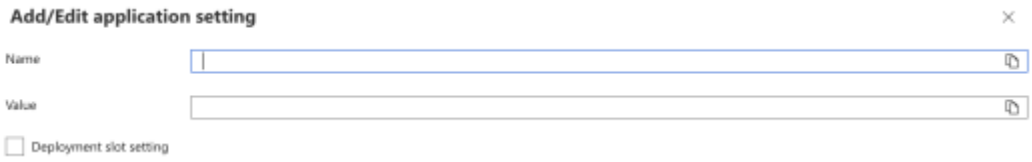
The recommendation is to use the **Configuration** option in Azure to facilitate the upgrade process when a new version is released.

To use the Manager tool:

1. Open a command prompt window.
2. Go to the folder where BOE Pro Reports is extracted.
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:
    - **Stack**: .NET
    - **.NET Version**: .NET 8 (LTS)
    - **Platform**: 64 Bit
    - **Basic Auth Publishing Credentials**: Off
    - **Manage pipeline version**: Integrated
    - **FTP state**: Disabled
    - **HTTP version**: 1.1
    - **Web sockets**: On
    - **Always on**: On
    - **ARR affinity**: Off
    - **HTTPS Only**: On
    - **Minimum TLS Version**: 1.2
    - **Remote debugging**: Off
    - **Client certificate mode**: Ignore
4. Click **Save**.
5. In the **API** section, select the **CORS** option, then set the following settings:
    - Select the **Enable Access-Control-Allow-Credentials** option.
    - 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 application. 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. 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**.
2. Upload the BOE Pro Reports ZIP file from your temporary folder 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.

Create BOE Pro WebAPI Server

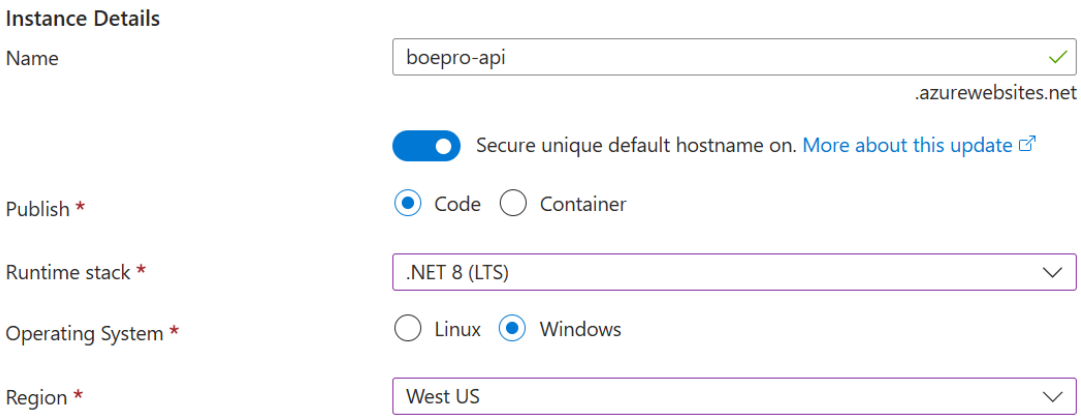
Prerequisites

Configuration

- ProPricer 9 Server host name and port.
- BOE Pro WebApp 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:
   - **Publish**: Code
   - **Runtime stack**: .NET 8 (LTS)
   - **Operating System**: Windows
   - **Region**: Select the desired region



***Recommendation**: Select the same subscription, resource group, and region so you can select the same App Service Plan for all BOE Pro App Services.*

4. On the **Deployment** tab, set **Continuous deployment** to **Disable**.
5. On the **Networking** tab, set **Enable public access** to **On**, and **Enable network injection** to **Off**.
6. On the **Monitoring** tab, set **Enable Application Insights** to **No**.
7. On the **Tags** tab, create the desired tags.
8. On the **Review + create** tab, 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. At 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 a 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 the configuration of 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. Replace **Initial Catalog** with **Database**.*

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

4. Create the following application settings:

Name	Value	Example
Host:Cors:AllowOrigin:0	<your-boepro-webapp-url>	https://boepro.azurewebsites.us
Security:Authority	<your authorization server pro webapi url>	https://asp-api.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. To configure email settings, add the following:

Name	Value	Example
EmailOptions:Login	The login to use to authenticate to the SMTP server	<smtplogin>
EmailOptions:Password	The login password to use to authenticate to the SMTP server	<smtppassword>
EmailOptions:PortNumber	The port number to connect to the SMTP server	587
EmailOptions:Provider	The SMTP server location	email-smtp.us-west-2.amazonaws.com
EmailOptions:RequiresAuthentication	Determines if the SMTP server requires authentication (true/false)	true
EmailOptions:ConnectionSecurity	Security protocol used by your email provider. Values can be: None, Auto, SslOnConnect, StartTls, StartTlsWhenAvailable	StartTls
EmailOptions:Email	The email to use to send emails	noreply@yourcompany.com
EmailOptions:Name	The name to use when sending emails	Your Company BOE Pro Notification

6. On the **General Settings** tab, verify or adjust the following settings:
    - **Stack**: .NET
    - **.NET Version**: .NET 8 (LTS)
    - **Platform**: 64 Bit
    - **Basic Auth Publishing Credentials:** Off
    - **Manage pipeline version**: Integrated
    - **FTP state**: Disabled
    - **HTTP version**: 1.1
    - **Web sockets**: On
    - **Always on**: On
    - **ARR affinity**: Off
    - **HTTPS Only**: On
    - **Minimum TLS Version**: 1.2
    - **Remote debugging**: Off
    - **Client certificate mode**: Ignore
7. Click **Save**.

Application Logging

Enable application logging to collect diagnostic information from this web application. 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. 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**.
2. Upload the BOE Pro WebAPI ZIP file from your temporary folder 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.

Enable AI configuration (Optional)

BOE Pro includes artificial intelligence (AI) capabilities that enhance the proposal estimation workflow. When enabled, the following AI-powered features become available:

- BOE Narrative Scoring: Measures the quality and completeness of text in BOE sections, providing a score and improvement suggestions to help users write more effective narratives.

- Formula Generation: Assists users in creating cost estimation formulas by analyzing BOE context and generating formula expressions using natural language processing.

Both features use the same AI service configuration. Enabling AI activates all AI-powered features simultaneously.

Warning: If enabled, BOE text and estimation context data are sent to the AI service you specify. For confidentiality and security reasons, a local or private service is highly recommended. AI-generated results are provided subject to the ProPricer Product terms, and Deltek does not independently verify the accuracy of each result. Users should independently review and verify the accuracy of all AI-generated scores and formulas.

There are two methods for configuring AI:

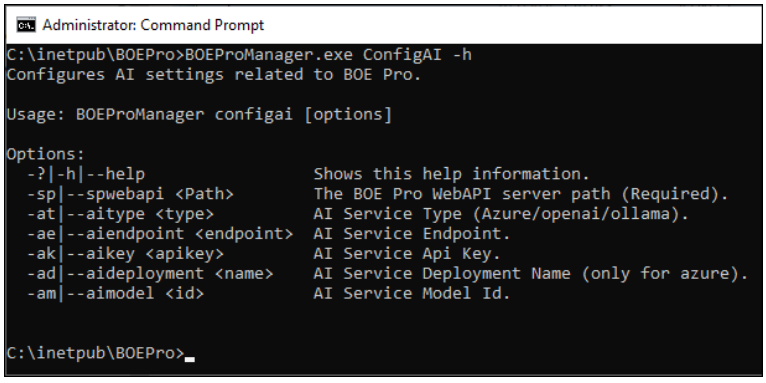
- Use the Manager tool before deploying the ZIP file.

- 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.

Method 1: Manager Tool

1. Open a command prompt window.
2. Go to the folder where BOE Pro WebAPI is extracted.
3. At the command prompt, enter: `BOEProManager.exe configai`
4. Enter values in the parameters that appear. Some parameters depend on the AI type you enter.



Method 2: Azure Portal Application Settings (Recommended)

1. In the App Service for BOE Pro WebAPI in the Azure Portal, go to the Settings section and select the Configuration option.

2. On the Application Settings tab, create the following settings:

Name	Value	Example
AI:Enabled	true	true
AI:LLM:Type	The AI service type: azure, openai, or ollama	azure
AI:LLM:Endpoint	The AI service endpoint URL (see notes below)	https://my-resource.openai.azure.com/
AI:LLM:ApiKey	The AI service API key	(your API key)
AI:LLM:DeploymentName	The AI deployment name. Only required for the Azure AI type.	gpt-4o
AI:LLM:ModelId	The AI model ID. Only required for the OpenAI and Ollama AI types.	gpt-4o

3. Click Save.

AI Endpoint notes by provider:

- Azure: Required. Your Azure OpenAI resource endpoint (for example, https://my-resource.openai.azure.com/).

- OpenAI: Optional. If left empty, the default OpenAI API endpoint (https://api.openai.com/v1) is used automatically. Specify a custom endpoint only if you are using an OpenAI-compatible service hosted at a different URL.

- Ollama: Required. Your local Ollama service endpoint (for example, http://localhost:11434).

Required settings by AI type:

Setting	Azure	OpenAI	Ollama
AI:Enabled	Yes	Yes	Yes
AI:LLM:Type	Yes	Yes	Yes
AI:LLM:Endpoint	Required	Optional	Required
AI:LLM:ApiKey	Required	Required	Not required
AI:LLM:DeploymentName	Required	Not required	Not required
AI:LLM:ModelId	Not required	Required	Required

After configuration, verify AI is working:

1. BOE Narrative Scoring: Open a proposal, navigate to a BOE section with text, and check that the scoring indicator appears.

2. Formula Generation: Open a proposal, navigate to a task's BOE formulas, and verify the AI-assisted formula generation option is available.

Note: AI configuration applies globally to all BOE Pro users. There is no option to enable individual AI features separately — all AI features are enabled or disabled together.

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 through the Deltek Support Center to obtain the login credentials.

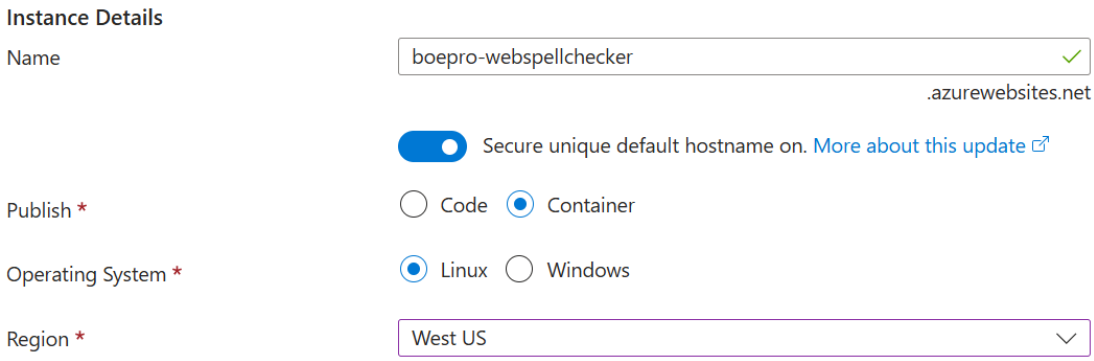
Prerequisites

Configuration

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

App Service

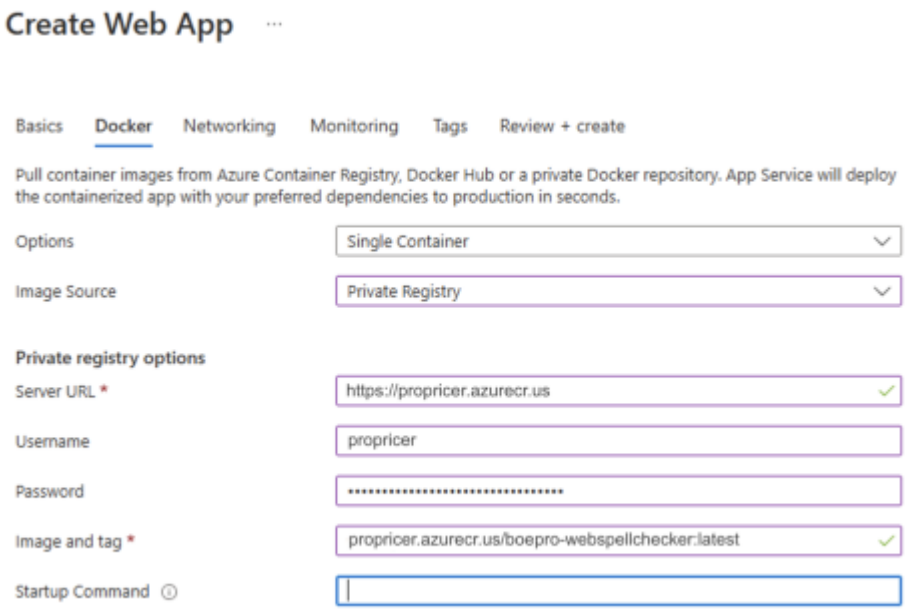
1. In the Azure Portal, add a Web App (App Service).
2. Enter the Web App name.
3. Select the following settings:
   - **Publish**: Docker Container
   - **Operating System**: Linux
   - **Region**: Select the desired region



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

4. Click **Next**.

5. On the **Docker** tab, enter the following:
    - **Options**: Single Container
    - **Image Source**: Private Registry
    - **Server URL**: https://propricer.azurecr.us
    - **Username**: propricer
    - **Password**: Contact Deltek Customer Support to obtain the password
    - **Image and tag**: propricer.azurecr.us/boepro-webspellchecker:latest
    - **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, set **WEBSITES_ENABLE_APP_SERVICE_STORAGE** to **true**.
3. On the **General Settings** tab, verify or adjust the following settings:
    - **Startup Command**: leave blank
    - **FTP state**: Disabled
    - **HTTP version**: 1.1
    - **HTTP 2.0 Proxy**: Off
    - **Always on**: On
    - **ARR affinity**: Off
    - **HTTPS Only**: On
    - **Minimum TLS Version**: 1.2
    - **Client certificate mode**: Ignore
4. Click **Save**.

Application Logging

Enable application logging to collect diagnostic information from this web application. 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.

Create BOE Pro Web Application

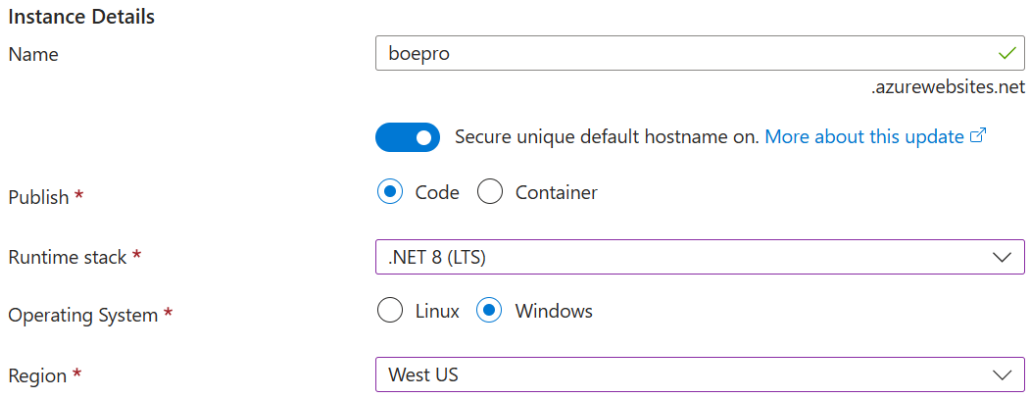
Prerequisites

Configuration

- BOE Pro WebAPI 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:
   - **Publish**: Code
   - **Runtime stack**: .NET 8 (LTS)
   - **Operating System**: Windows
   - **Region**: Select the desired region



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

4. On the **Deployment** tab, set **Continuous deployment** to **Disable**.
5. On the **Networking** tab, set **Enable public access** to **On**, and **Enable network injection** to **Off**.
6. On the **Monitoring** tab, set **Enable Application Insights** to **No**.
7. On the **Tags** tab, create the desired tags.
8. On the **Review + create** tab, click **Create**.

Configuration

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

To use the Manager tool:

1. Open a command prompt window.
2. Go to the folder where BOE Pro is extracted and extract **DeltekProPricerBOEPro_WebApp_[version]_win-x64.zip** to a temporary folder.
3. At the command prompt in your temporary folder, enter: `BOEProWebAppManager.exe config`

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

---

4. (Optional) If you installed WebSpellChecker and want to enable it in BOE Pro:
    - At the command prompt, enter: `BOEProWebAppManager.exe EnableWebSpellChecker`
5. In Windows Explorer, select all contents in your temporary folder and zip it into a new file.

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 Deltek, 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 WebApp after enabling or skipping Clarity:

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:
    - **Stack**: .NET
    - **.NET Version**: .NET 8 (LTS)
    - **Platform**: 64 Bit
    - **Basic Auth Publishing Credentials**: Off
    - **Manage pipeline version**: Integrated
    - **FTP state**: Disabled
    - **HTTP version**: 1.1
    - **Web sockets**: Off
    - **Always on**: On
    - **ARR affinity**: Off
    - **HTTPS Only**: On
    - **Minimum TLS Version**: 1.2
    - **Remote debugging**: Off
    - **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 WebApp.

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. 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**.
2. Upload the BOE Pro WebApp ZIP file from your temporary folder 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 BOE Pro Authorization Server

BOE Pro version 3.5.105.1 and later use the new Authorization Server Pro website to authenticate user logins. This makes the BOE Pro Authorization Server website unnecessary. If you have BOE Pro version 3.5.105.0 or earlier, delete the BOE Pro Authorization Server website, application pool, and the website's folder in **C:\inetpub**. (For example, **C:\inetpub\BOEPro\AuthorizationServer**).

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 application no longer necessary. If you have BOE Pro v3.5.100.2 or earlier, delete the ProPricer 9 WebAPI web application after installing BOE Pro v3.5.100.3 or later.

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

Upgrade from 3.5.100.x

If you configure BOE Pro to use Azure AD logins, you must change all configuration keys that begin with **AzureActiveDirectory** to **AzureAD**. This change is required if you are upgrading to BOE Pro version 3.5.101.0 or newer.

Upgrade to 3.5.104.2 (.NET 8)

Earlier versions of BOE Pro required .NET 6. Make sure the runtime stack is updated to .NET 8 (LTS) for all your web application configurations.

Also, make sure Connection Strings in Authorization Server and WebAPI include:
`TrustServerCertificate=True;`

Upgrade to 3.5.105.1 (.NET 8)

Earlier versions of BOE Pro use BOE Pro's authorization server for authentication. Starting with version 3.5.105.1, BOE Pro uses the Authorization Server Pro application for authentication instead. Therefore, a migration of existing users is required to prepare for the 3.5.105.1 upgrade process.

Before starting the upgrade process, log in to BOE Pro as an Administrator user and deactivate the license.

Migrate users

To migrate users from an earlier BOE Pro version:

1. Unzip **DeltekProPricerBOEPro_[version]_azure.zip.**
2. Open a command prompt window.
3. Go to the folder where **BOEProDatabaseSetup**.exe is extracted.
4. At the command prompt, enter: `BOEProDatabaseSetup.exe migrate-users`
5. Enter the information requested.

Upgrade BOE Pro

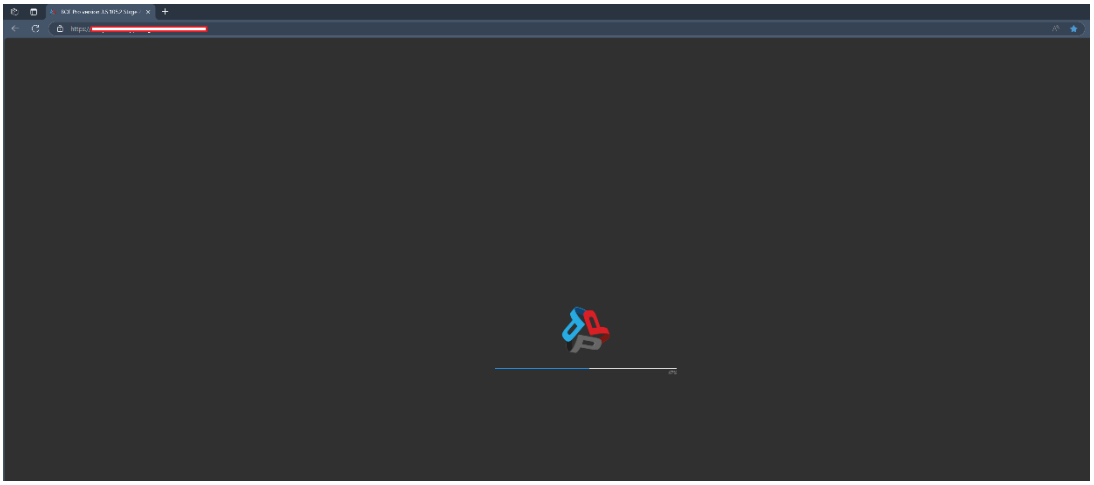
After migrating all existing BOE Pro users:

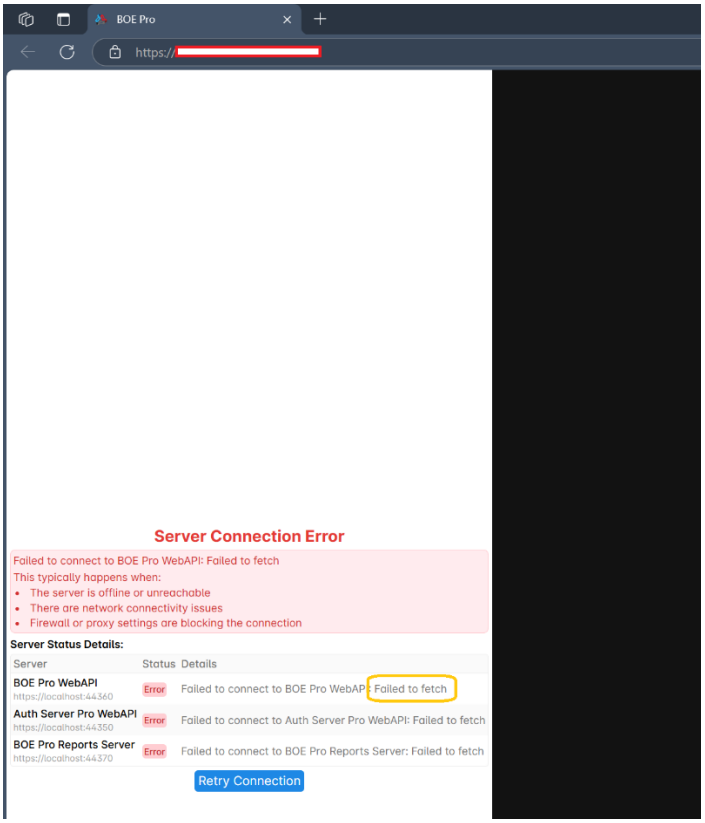
1. At the command prompt, enter: `BOEProDatabaseSetup.exe upgrade`
2. Enter the information requested.
3. Log in to Authorization Server Pro to activate your BOE Pro license and assign it to users.

Troubleshoot

Authorization Server Pro, BOE Pro WebAPI Server, and BOE Pro Reports Server – Fetch error

At BOE Pro WebApp startup, there will be a status checkup for all required servers. You might receive the **Failed to fetch** error message after entering the WebApp URL in your browser (for example, **https://boepro.mycompany.com**).





Solution 1

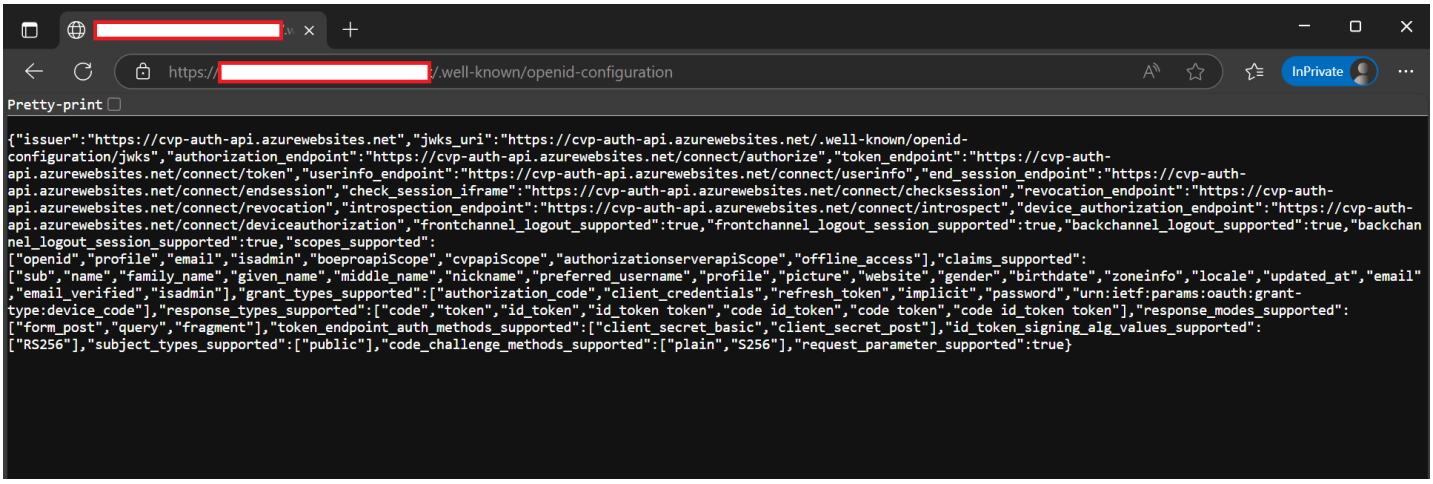
Check your networking. Verify that your firewall is not blocking the authorization server port.

Solution 2

Restart the server website.

Solution 3

1. For Authorization Server Pro, in your browser enter your server URL followed by **/.well-known/openid-configuration** (for example, **https:// authserverpro.mycompany.com/.well-known/openid-configuration**). The resulting .json file should look like:



2. If you receive an error instead of the .json file, go to the **Authorization Server Pro WebAPI** folder and find the file **appsettings.json**. Verify that it is a valid .json file containing the correct configuration.

Solution 4

1. Enable logging in server.
2. Try logging in to BOE Pro again and check the log for the error.

If there is no log file, check Azure Advanced Tools in the Authorization Server Pro web application.

Database Setup reference

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

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` 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 a 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` 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.

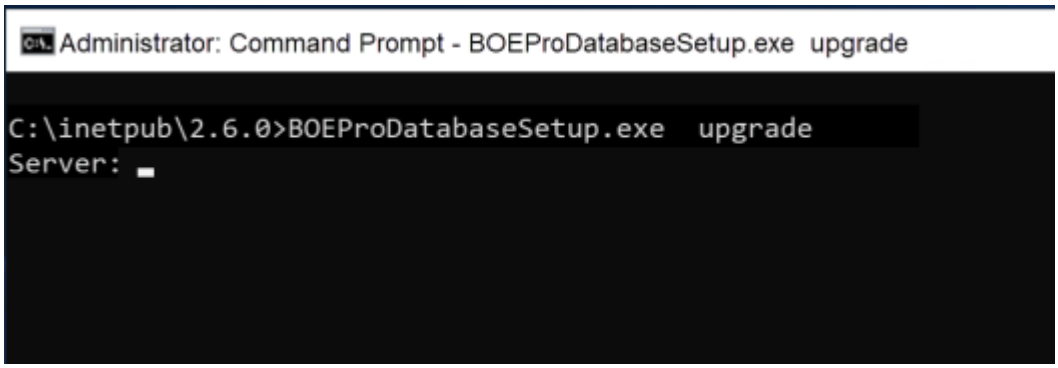
`-p|--dbapass <dbapassword>` Database user password.

`-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. Extract **BOEProDatabaseSetup.exe**.
3. Open a command prompt window.
4. Go to the folder where **BOEProDatabaseSetup.exe** is extracted.
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.

Examples

Upgrade a database named **BOEPro** on a SQL server on a 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

`addsysadmin` Create an administrator-type user in a previously created BOE Pro database. The user will have the Admin-only role. There can be only one user with the Admin-only role.

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.

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