Hosting Blazor Boilerplate on Microsoft Azure

Contents

Preamble

Please be aware that hosting on Microsoft Azure will cost money.

This tutorial assumes that you do already have a working azure subscription.

In this example, Blazor Boilerplate is being hosted using App Services and a managed SQL database.

The certificate for Identity Server has been stored in Azure Key Vault.

Please ensure that for all resources created in this tutorial, you choose the same subscription, region and resource group!

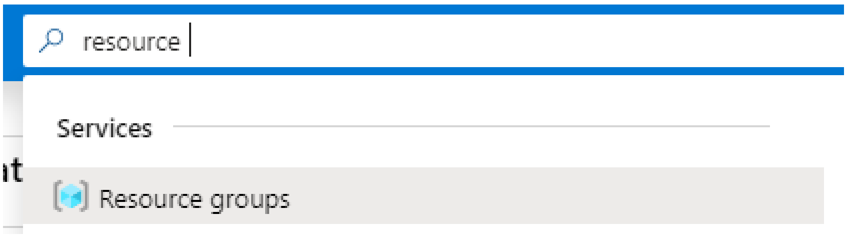
All URIs, resources, user names and passwords shown have been deleted prior to publishing.

Create Resource Group

Open up Microsoft Azure.

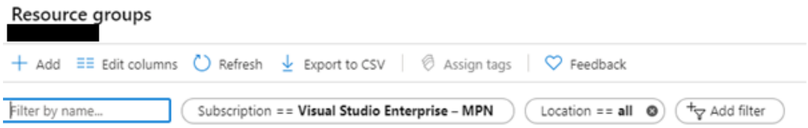
If you do not already have a resource group, create a new one.

Search for Resource Group in the search field



resource

Select Resource groups



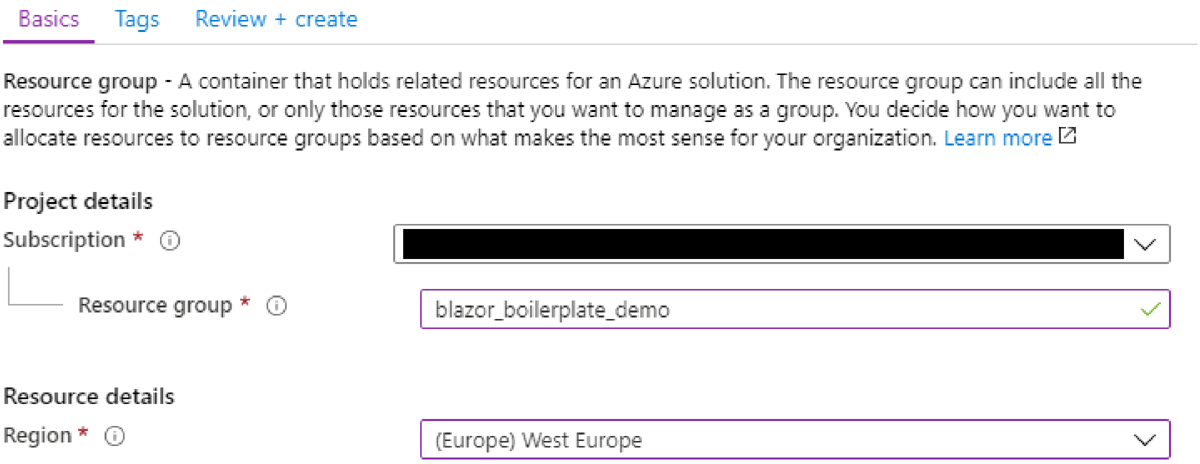
Resource groups

[redacted]

Subscription == **Visual Studio Enterprise – MPN**

Location == **all**

Click Add



Resource group - A container that holds related resources for an Azure solution. The resource group can include all the resources for the solution, or only those resources that you want to manage as a group. You decide how you want to allocate resources to resource groups based on what makes the most sense for your organization. Learn more

Project details

Subscription * [redacted]

Resource group * blazor_boilerplate_demo

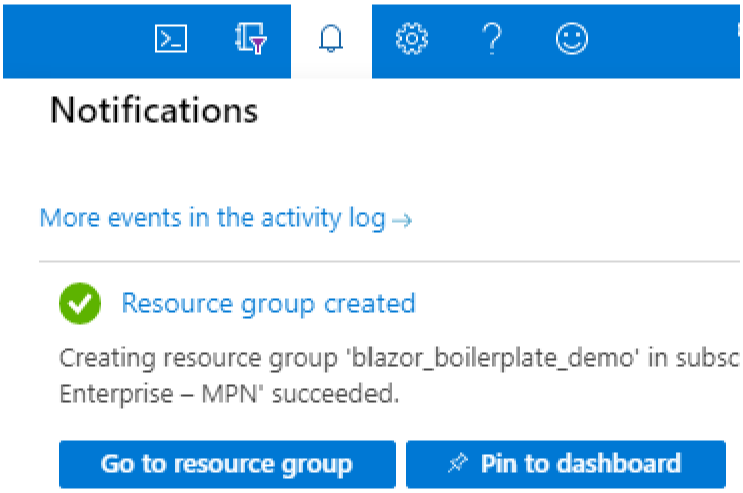
Resource details

Region * (Europe) West Europe

Choose your subscription, a name and a region

Click Review + Create

Click Create



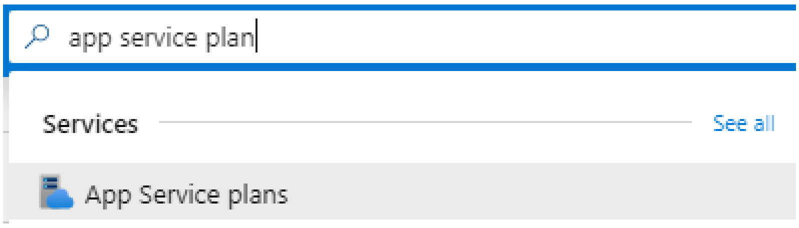
Resource group created

Creating resource group 'blazor_boilerplate_demo' in subsc
Enterprise – MPN' succeeded.

Wait for the deployment to finish

Create App Service Plan

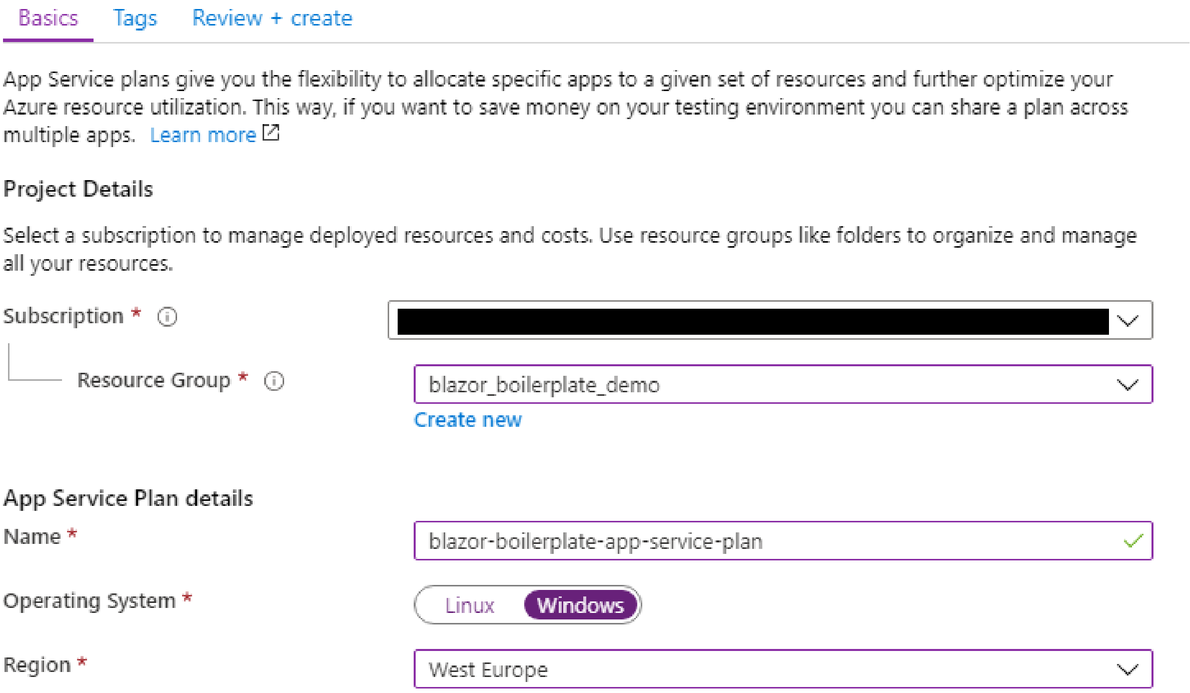
Next, we need an App Service Plan to host our application



app service plan

Search for “App Service Plans”

Click “Create app service plan”



App Service plans give you the flexibility to allocate specific apps to a given set of resources and further optimize your Azure resource utilization. This way, if you want to save money on your testing environment you can share a plan across multiple apps. Learn more

Project Details

Select a subscription to manage deployed resources and costs. Use resource groups like folders to organize and manage all your resources.

Subscription * [redacted]

Resource Group * blazor_boilerplate_demo

Create new

App Service Plan details

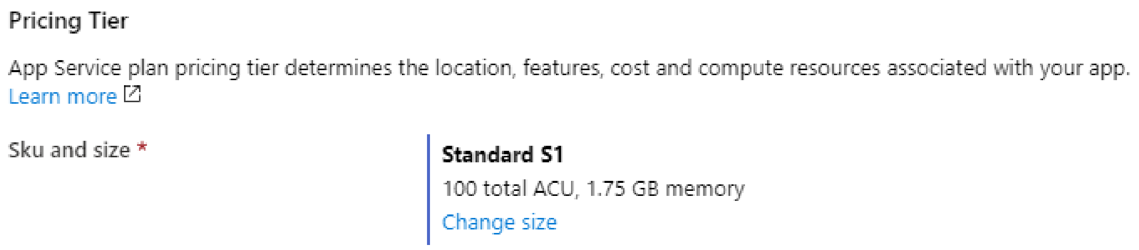
Name * blazor-boilerplate-app-service-plan ✓

Operating System * Linux [✓ Windows]

Region * West Europe

Choose your subscription, resource group, a name for the plan and a region to host it.

Select Windows as the operating system



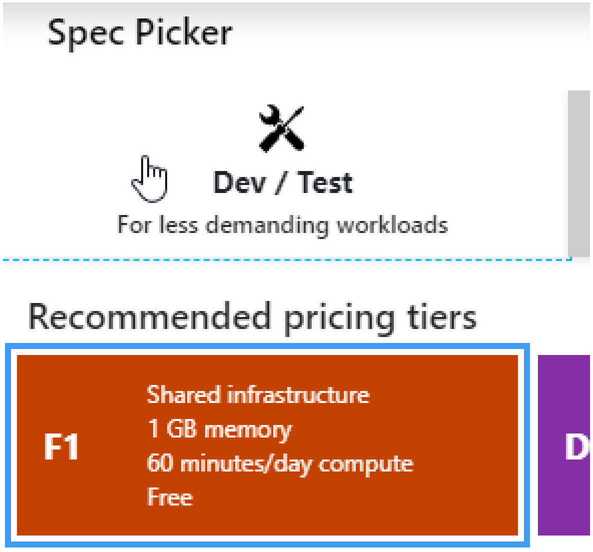
Pricing Tier

App Service plan pricing tier determines the location, features, cost and compute resources associated with your app.
Learn more

Sku and size *

Standard S1
100 total ACU, 1.75 GB memory
Change size

Click on „Change size“ to select the free tier for app service plans



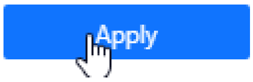
Spec Picker

Dev / Test
For less demanding workloads

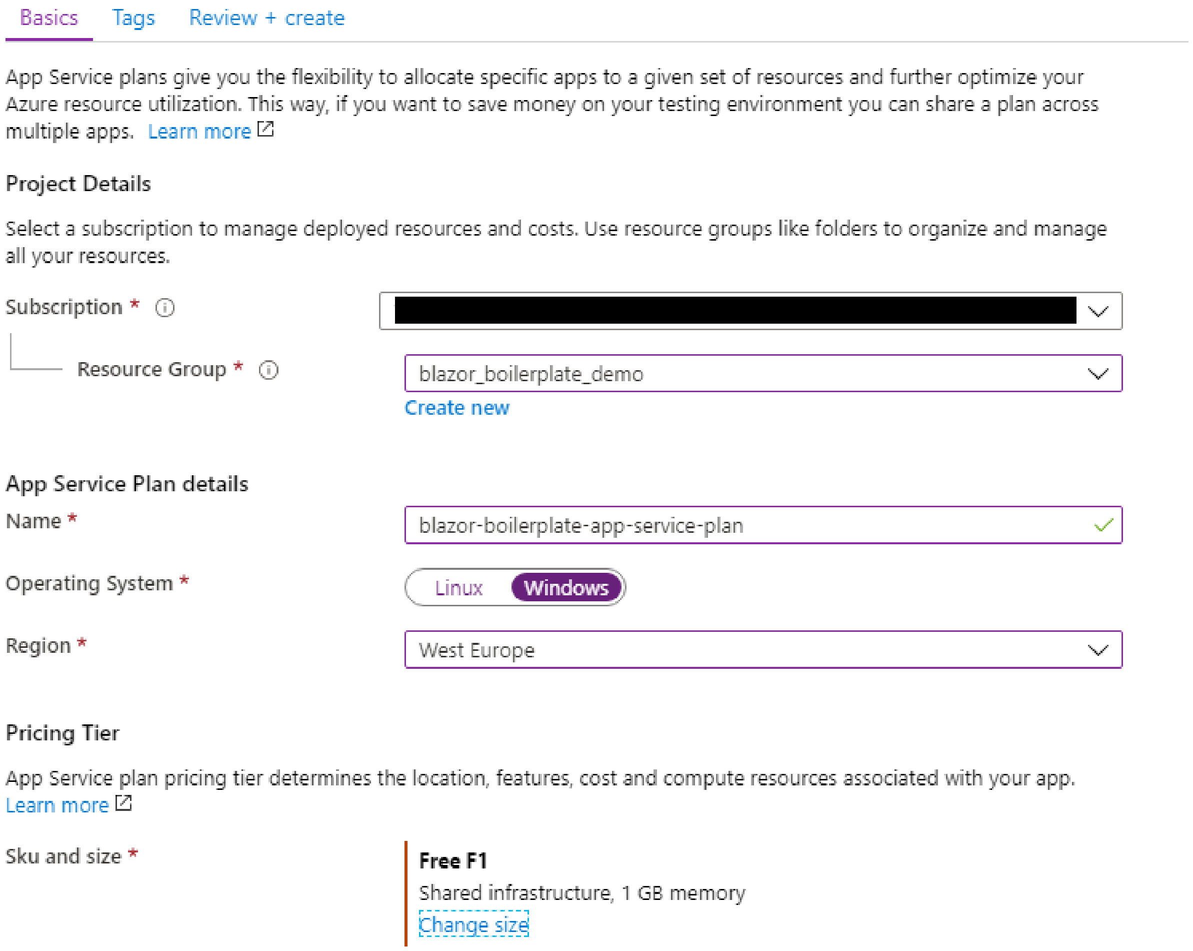
Recommended pricing tiers

F1
Shared infrastructure
1 GB memory
60 minutes/day compute
Free

Click „Dev / Test“ and select “Share infrastructure”



Click „Apply“



App Service plans give you the flexibility to allocate specific apps to a given set of resources and further optimize your Azure resource utilization. This way, if you want to save money on your testing environment you can share a plan across multiple apps. Learn more

Project Details

Select a subscription to manage deployed resources and costs. Use resource groups like folders to organize and manage all your resources.

Subscription * [redacted]

Resource Group * blazor_boilerplate_demo

App Service Plan details

Name * blazor-boilerplate-app-service-plan

Operating System * Linux [✓ Windows]

Region * West Europe

Pricing Tier

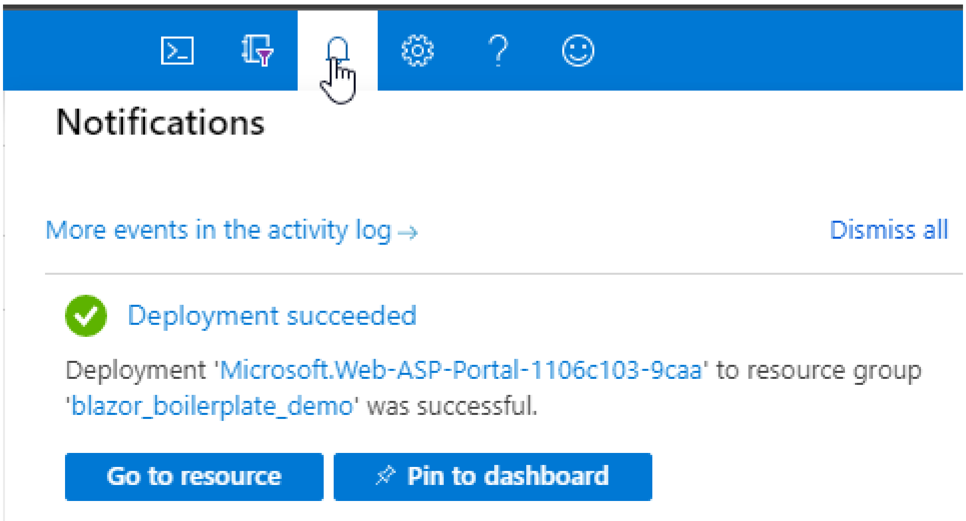
App Service plan pricing tier determines the location, features, cost and compute resources associated with your app. Learn more

Sku and size * **Free F1**
Shared infrastructure, 1 GB memory

Review your choices.

Click „Review + create“

Click „Create“



Notifications

More events in the activity log →

Dismiss all

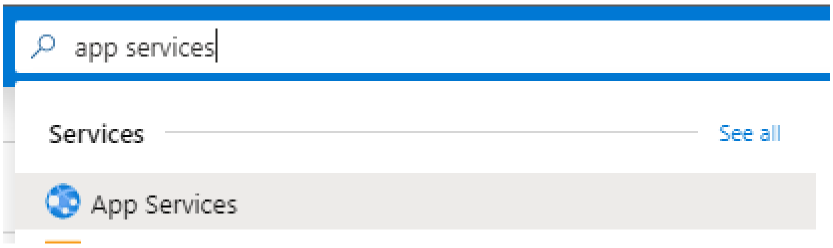
Deployment succeeded

Deployment 'Microsoft.Web-ASP-Portal-1106c103-9caa' to resource group 'blazor_boilerplate_demo' was successful.

Wait for the deployment to finish

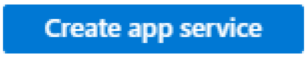
Create App Service

Search for “App Services”

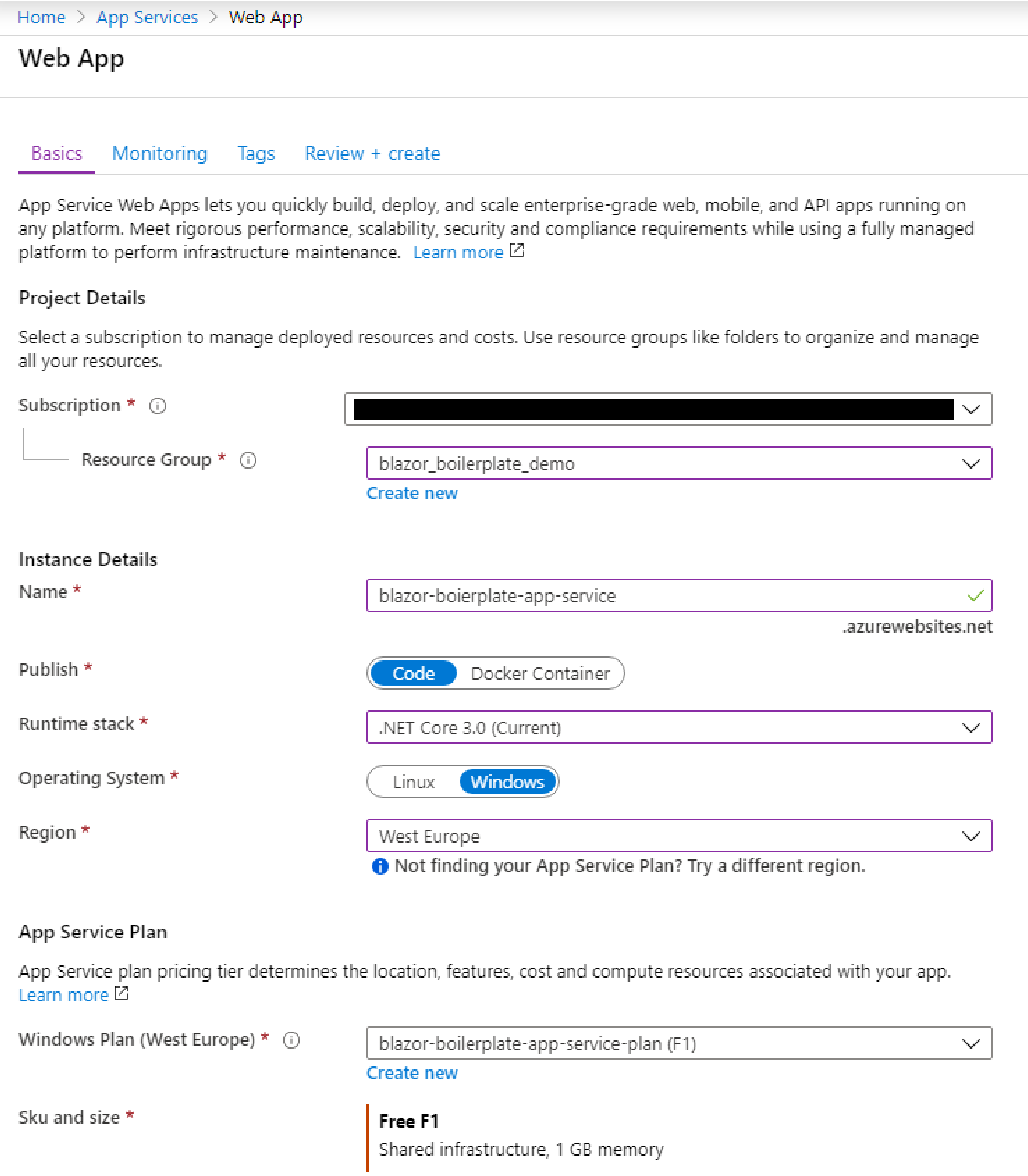


app services

Choose „App Services“



Click „Create app service“



Home > App Services > Web App

Web App

Basics | Monitoring | Tags | Review + create

App Service Web Apps lets you quickly build, deploy, and scale enterprise-grade web, mobile, and API apps running on any platform. Meet rigorous performance, scalability, security and compliance requirements while using a fully managed platform to perform infrastructure maintenance. Learn more

Project Details

Select a subscription to manage deployed resources and costs. Use resource groups like folders to organize and manage all your resources.

Subscription * [redacted]

Resource Group * blazor_boilerplate_demo

Instance Details

Name * blazor-boierplate-app-service .azurewebsites.net

Publish * [✓ Code] Docker Container

Runtime stack * .NET Core 3.0 (Current)

Operating System * Linux [✓ Windows]

Region * West Europe

Not finding your App Service Plan? Try a different region.

App Service Plan

App Service plan pricing tier determines the location, features, cost and compute resources associated with your app. Learn more

Windows Plan (West Europe) * blazor-boilerplate-app-service-plan (F1)

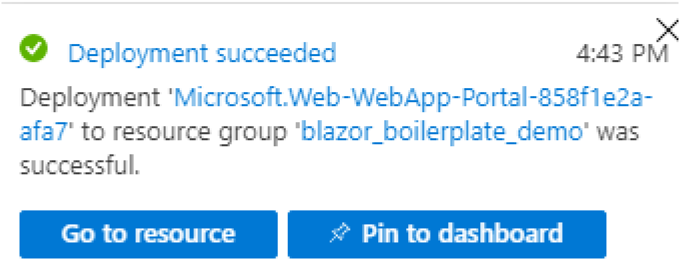
Sku and size * **Free F1** Shared infrastructure, 1 GB memory

Choose your subscription, resource group, a name for the app, publish code, .Net Core Runtime, Windows and region.

Please note: Although Blazor Boilerplate currently targets .Net 3.1, .Net Core 3.0 has been selected as choosing 3.1 will switch the OS from Windows to Linux. For debugging purposes, Windows is being preferred. The app will still work.

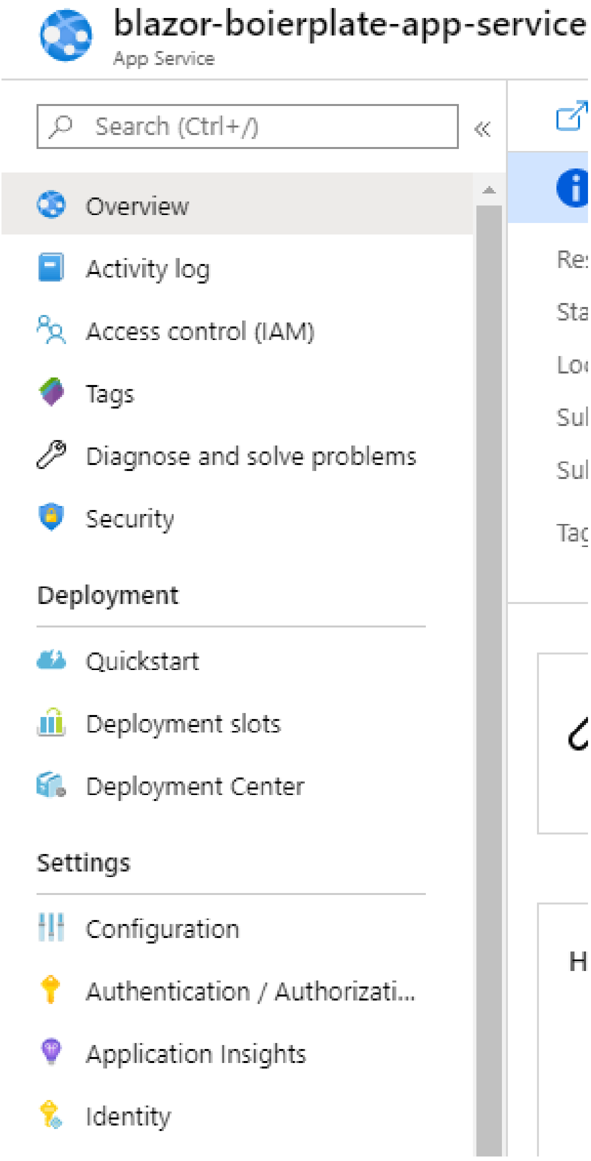
Click “Review + Create”

Click “Create”

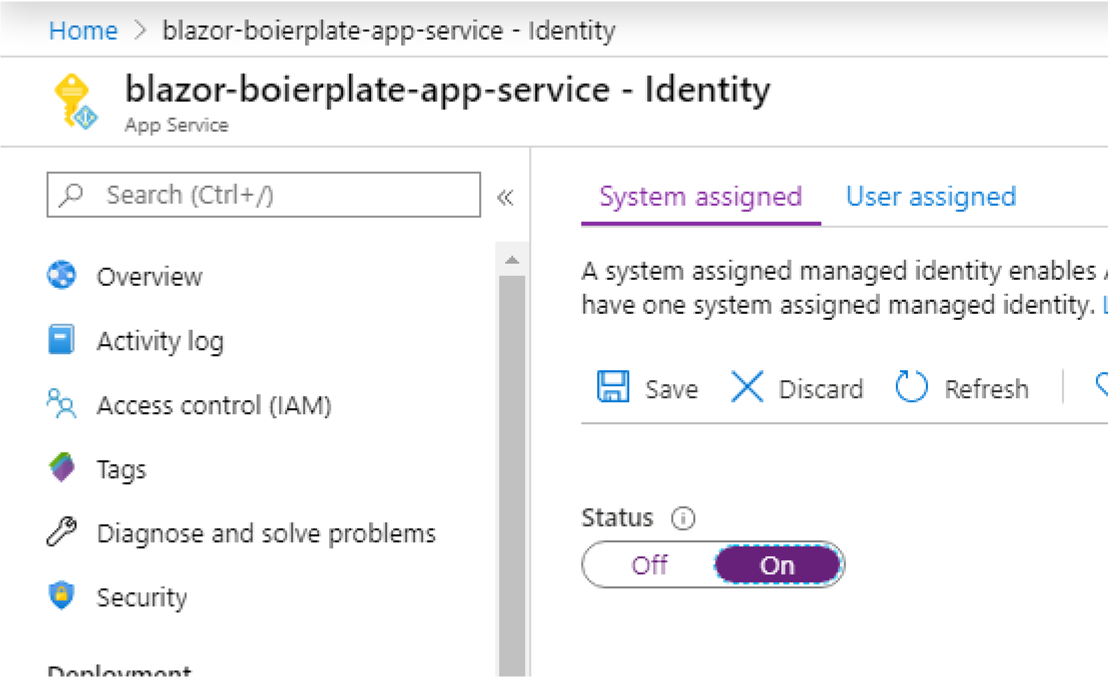

Deployment succeeded 4:43 PM

Deployment 'Microsoft.Web-WebApp-Portal-858f1e2a-afa7' to resource group 'blazor_boilerplate_demo' was successful.

After the deployment has finished, select “Go to resource”

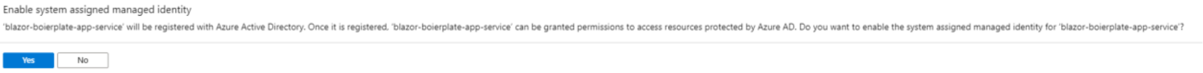


Select “Identity” on the left hand side.



A system assigned managed identity enables
have one system assigned managed identity. L

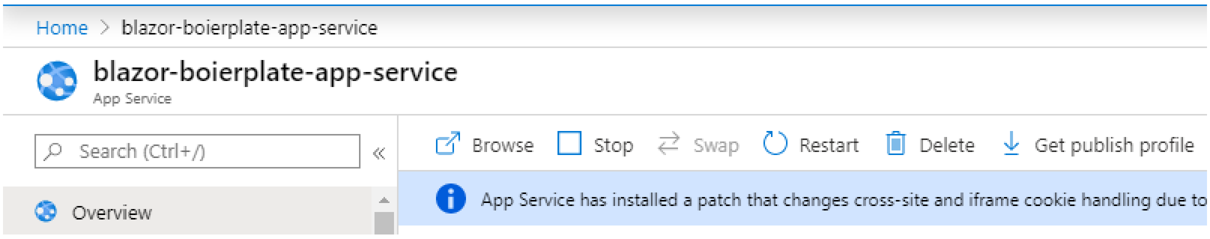
Note that “Status” for System assigned identities is switched to “Off” by default. Enable it and click on “Save”



Enable system assigned managed identity

'blazor-boierplate-app-service' will be registered with Azure Active Directory. Once it is registered, 'blazor-boierplate-app-service' can be granted permissions to access resources protected by Azure AD. Do you want to enable the system assigned managed identity for 'blazor-boierplate-app-service'?

Accept to restart the app.



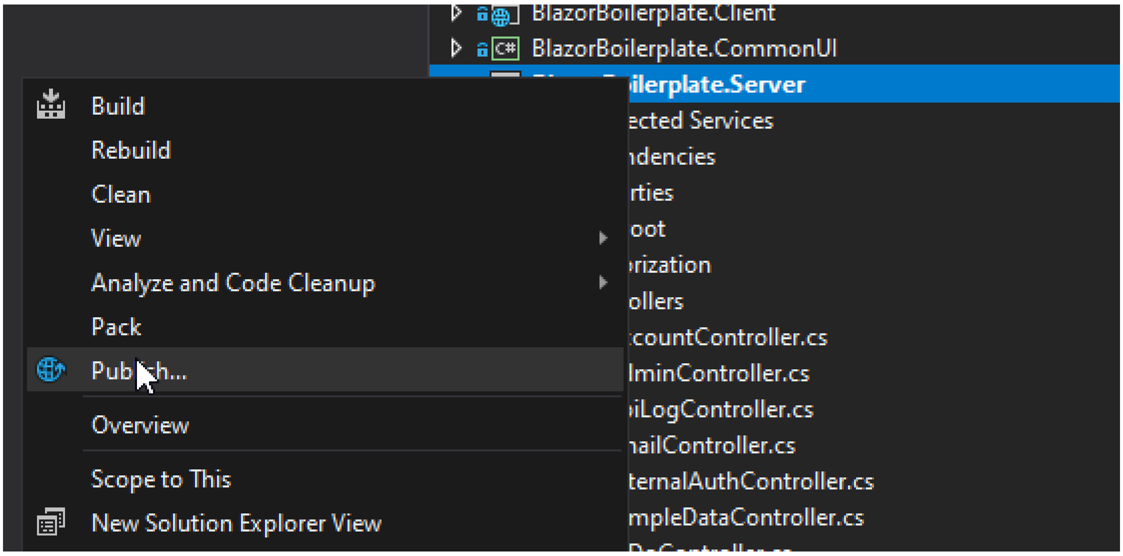
App Service has installed a patch that changes cross-site and iframe cookie handling due to

Click on “Get publish profile” to download the publish profile.

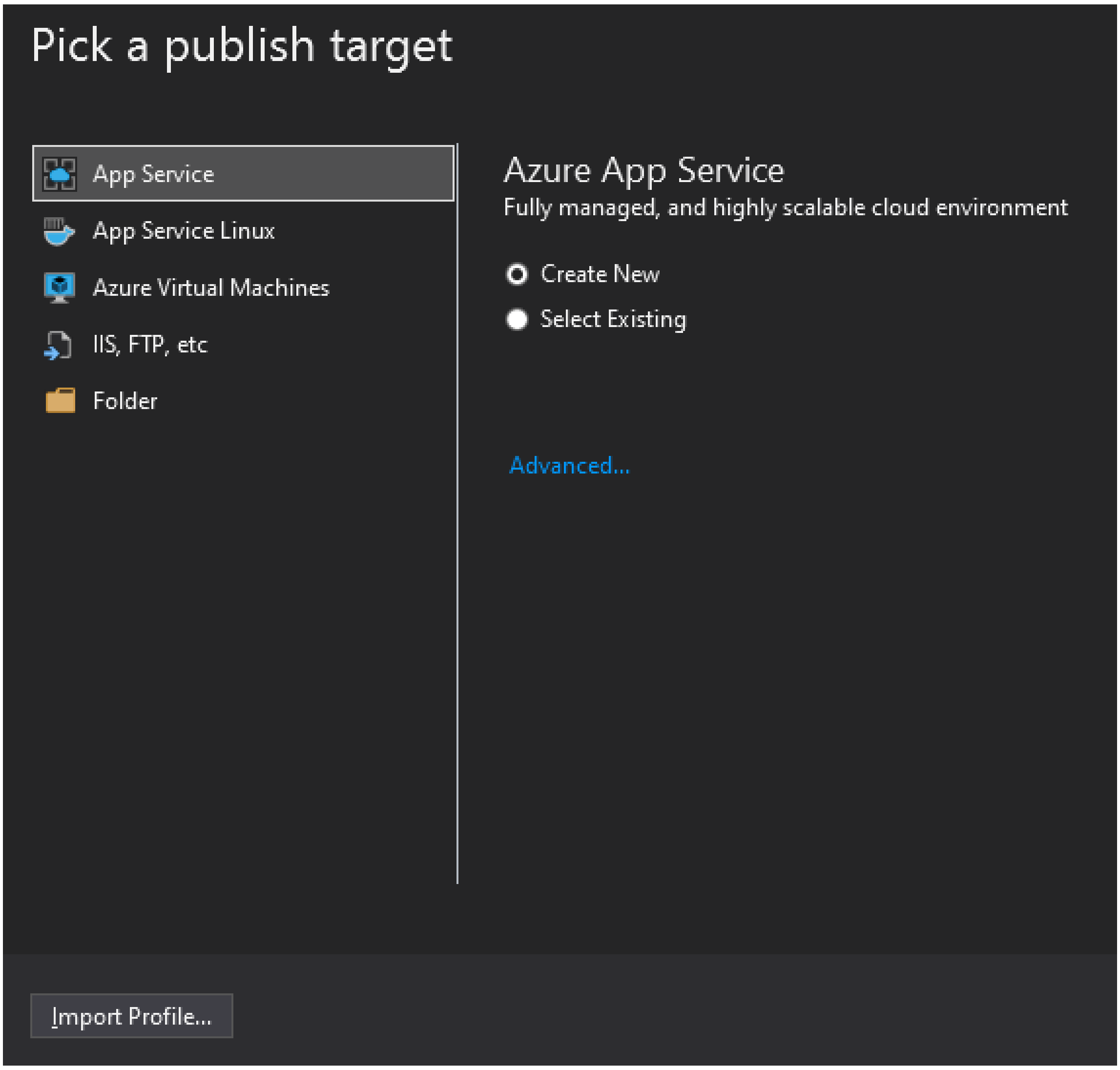
Import the Publish Profile to Visual Studio

In Visual Studio, open the Blazor Boilerplate Solution.

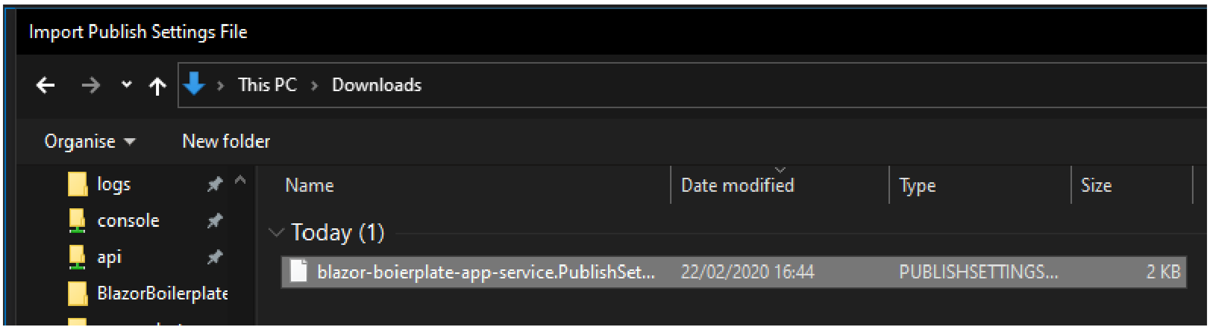
Right-Click on BlazorBoilerplate.Server and select “Publish…”



Click on “Start”



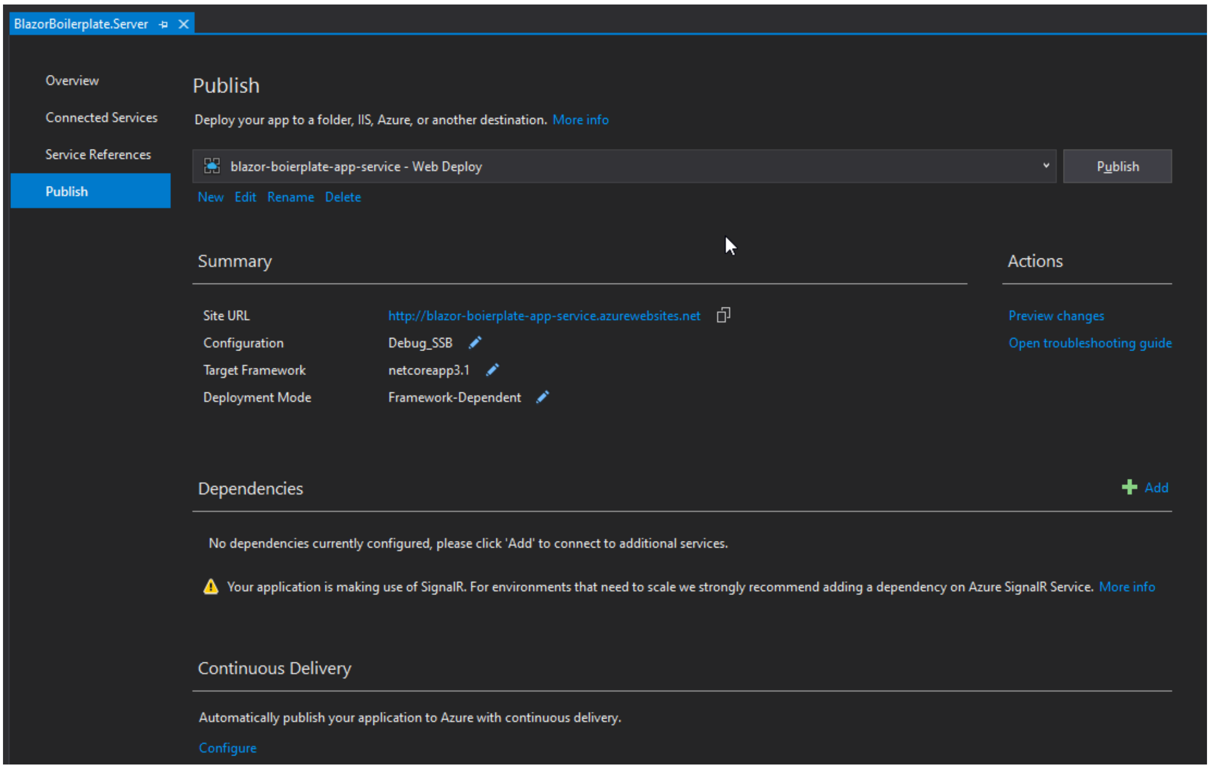
Choose "Import Profile..."



Name	Date modified	Type	Size
blazor-boierplate-app-service.PublishSet...	22/02/2020 16:44	PUBLISHSETTINGS...	2 KB

Select the downloaded profile.

Click "Open"



Publish

Deploy your app to a folder, IIS, Azure, or another destination. More info

blazor-boierplate-app-service - Web Deploy

Summary

Site URL	http://blazor-boierplate-app-service.azurewebsites.net
Configuration	Debug_SSB
Target Framework	netcoreapp3.1
Deployment Mode	Framework-Dependent

Actions

Dependencies

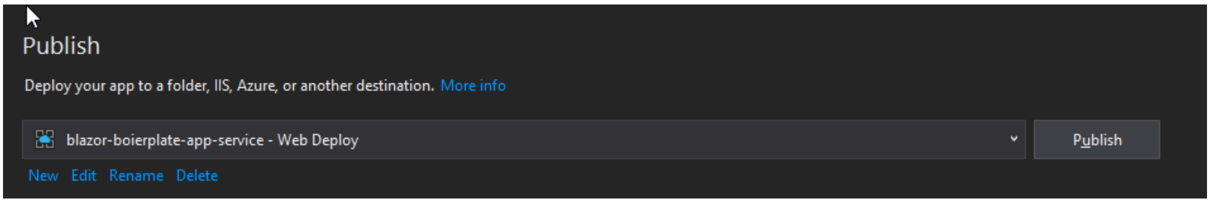
No dependencies currently configured, please click 'Add' to connect to additional services.

Your application is making use of SignalR. For environments that need to scale we strongly recommend adding a dependency on Azure SignalR Service. More info

Continuous Delivery

Automatically publish your application to Azure with continuous delivery.

Three profiles will be imported. We will use the “Web Deploy” profile in this tutorial.



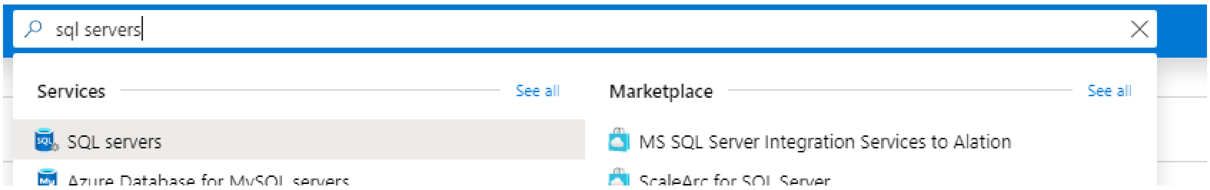
Publish

Deploy your app to a folder, IIS, Azure, or another destination. More info

blazor-boierplate-app-service - Web Deploy

Create SQL Server

Back in Azure, search for “SQL servers”

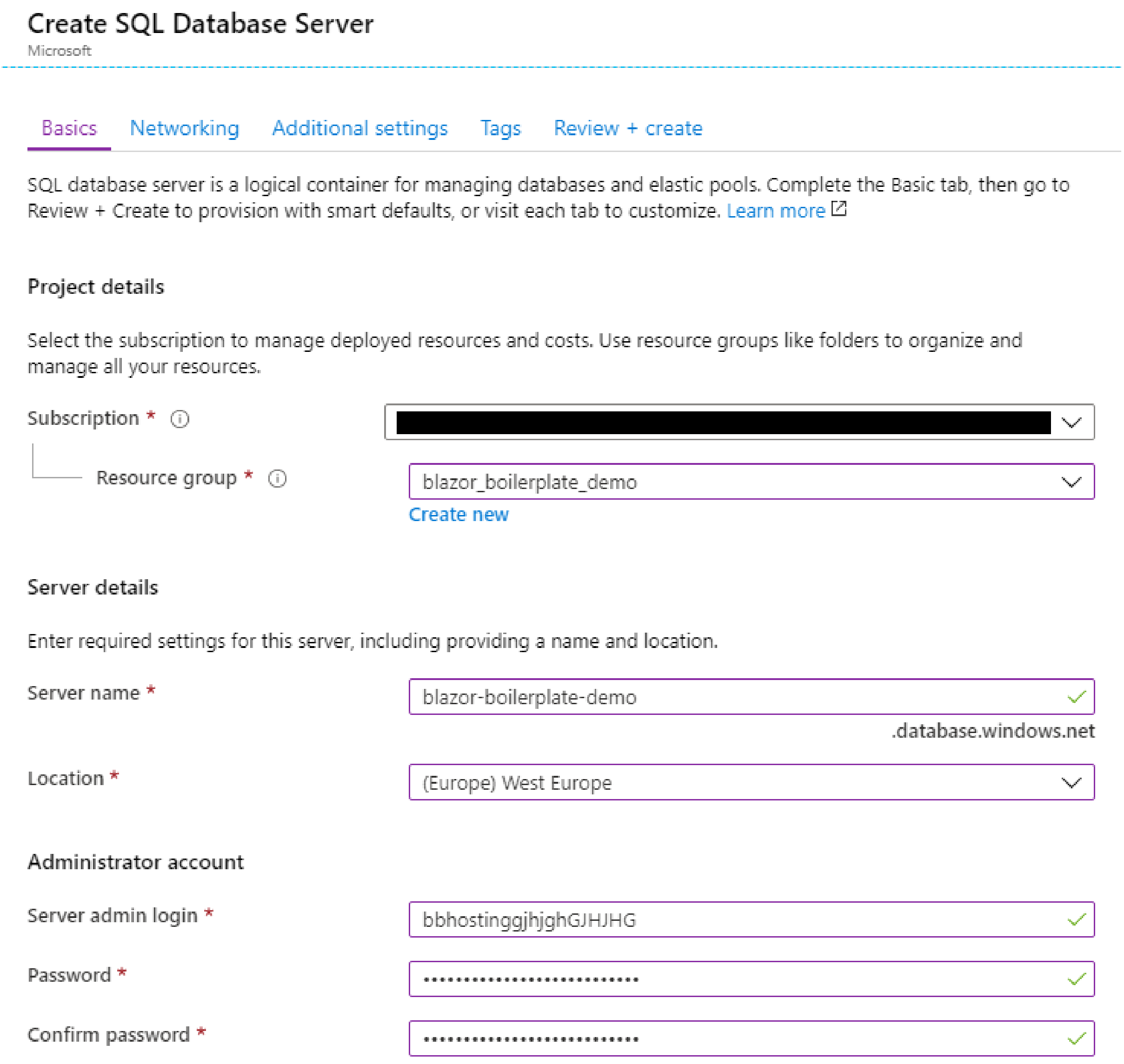


sql servers

Services
See all
SQL servers
Azure Database for MySQL servers
Marketplace
See all
MS SQL Server Integration Services to Alation
ScaleArc for SQL Server

Choose “SQL servers”

Click “Add”



Create SQL Database Server

Microsoft

SQL database server is a logical container for managing databases and elastic pools. Complete the Basic tab, then go to Review + Create to provision with smart defaults, or visit each tab to customize. Learn more

Project details

Select the subscription to manage deployed resources and costs. Use resource groups like folders to organize and manage all your resources.

Subscription * [redacted]

Resource group * blazor_boilerplate_demo

Create new

Server details

Enter required settings for this server, including providing a name and location.

Server name * blazor-boilerplate-demo .database.windows.net

Location * (Europe) West Europe

Administrator account

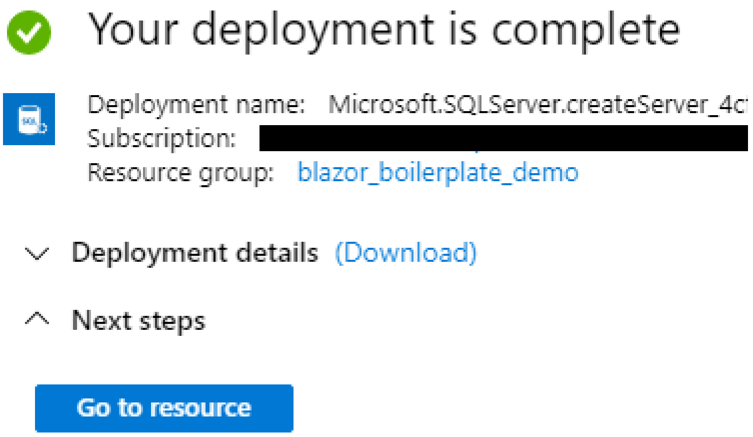
Server admin login * bbhostinggjhjghGJHJHG

Password * ••••••••••••••••••••••••••

Confirm password * ••••••••••••••••••••••••••

Choose your Subscription, Resource group, Server name, Location, Server admin login and Password. Also, confirm the password.

Click Create



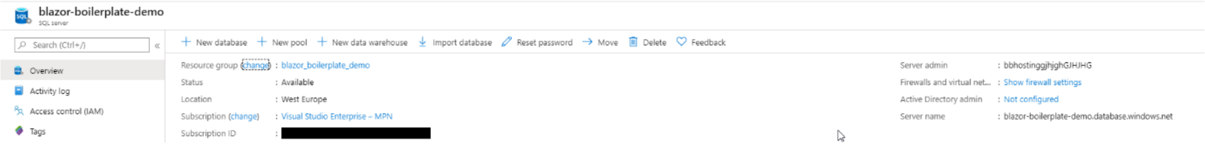
Your deployment is complete

Deployment name: Microsoft.SQLServer.createServer_4c

Subscription: [redacted]

Resource group: blazor_boilerplate_demo

Once the deployment is finished, select “Go to resource”



blazor-boilerplate-demo

SQL server

Resource group (change) : blazor_boilerplate_demo

Status : Available

Location : West Europe

Subscription (change) : Visual Studio Enterprise – MPN

Subscription ID : [redacted]

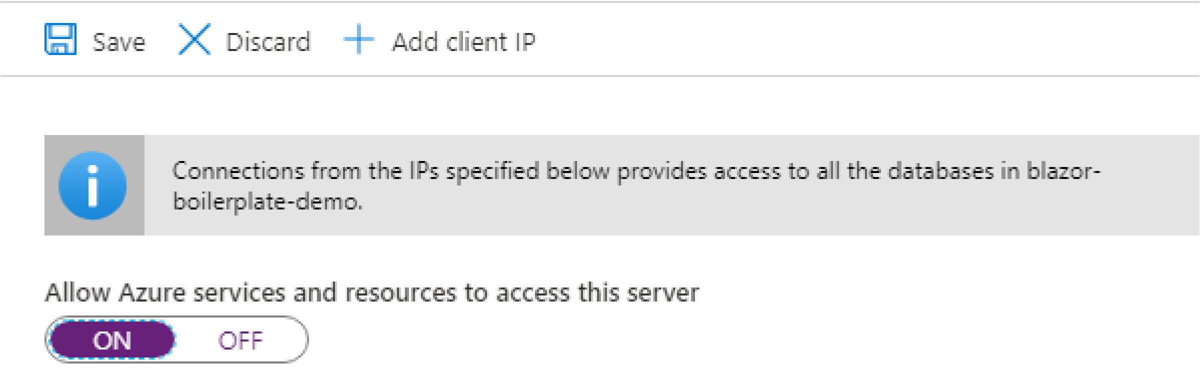
Server admin : bbhostinggjhjghGJHJHG

Firewalls and virtual net... : Show firewall settings

Active Directory admin : Not configured

Server name : blazor-boilerplate-demo.database.windows.net

Click on “Show firewall settings” on the right hand side.

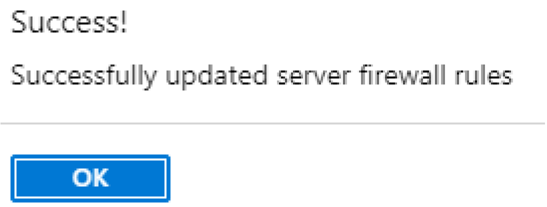


Connections from the IPs specified below provides access to all the databases in blazor-boilerplate-demo.

Allow Azure services and resources to access this server

Switch this setting to on.

Click Save.



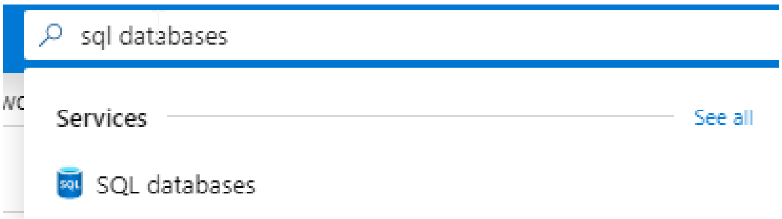
Success!

Successfully updated server firewall rules

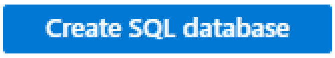
Click okay.

Create SQL Database

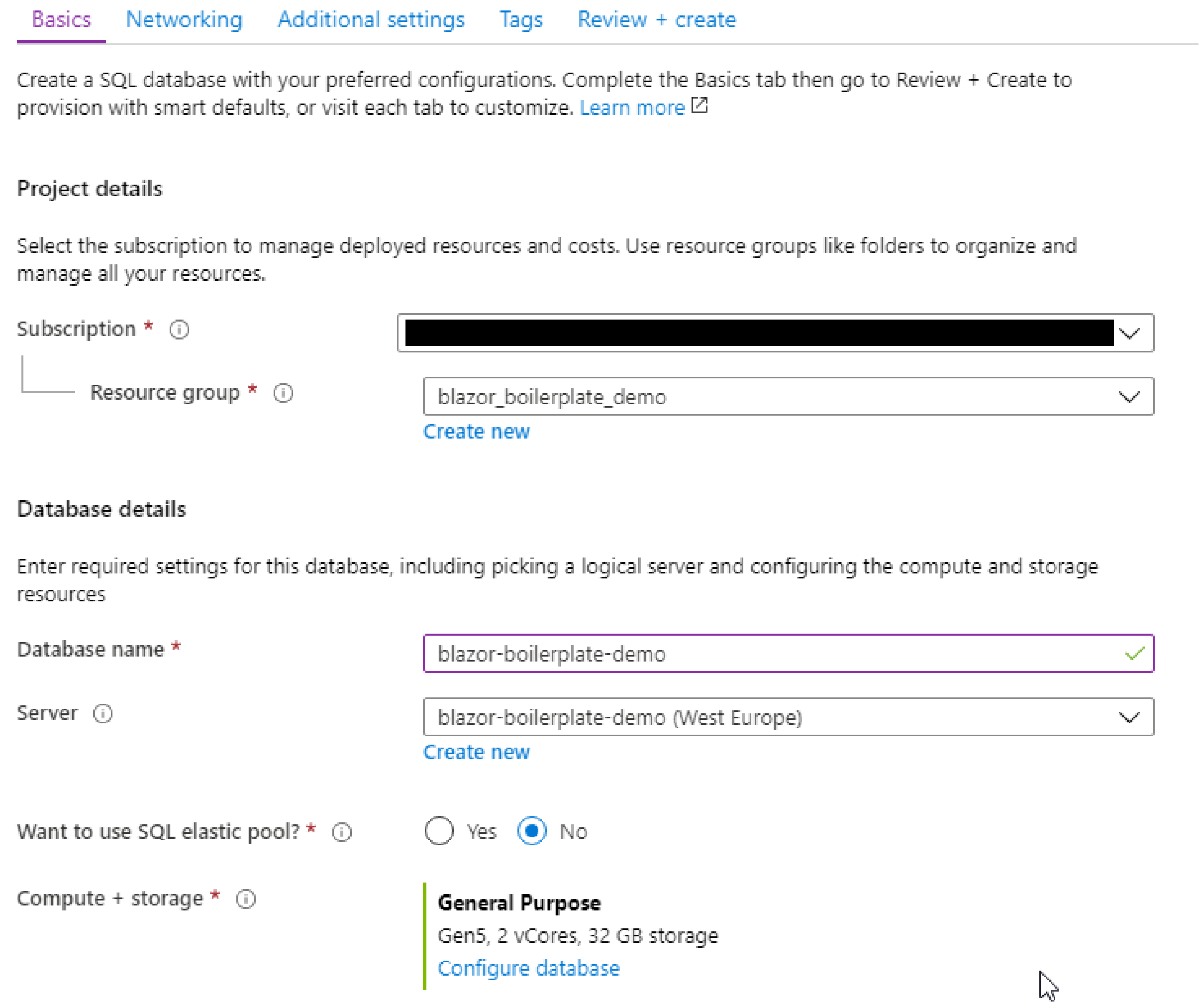
In Azure, search for “SQL databases”



Select “SQL databases”



Click “Create SQL database”



Create a SQL database with your preferred configurations. Complete the Basics tab then go to Review + Create to provision with smart defaults, or visit each tab to customize. Learn more

Project details

Select the subscription to manage deployed resources and costs. Use resource groups like folders to organize and manage all your resources.

Subscription * [redacted]

Resource group * blazor_boilerplate_demo

Create new

Database details

Enter required settings for this database, including picking a logical server and configuring the compute and storage resources

Database name * blazor-boilerplate-demo ✓

Server blazor-boilerplate-demo (West Europe)

Create new

Want to use SQL elastic pool? * ○ Yes ● No

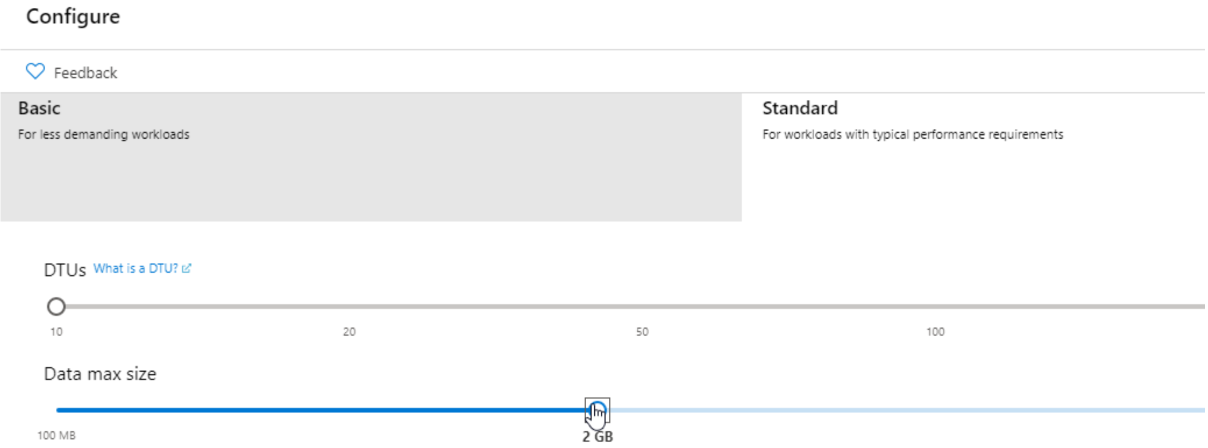
Compute + storage *

General Purpose

Gen5, 2 vCores, 32 GB storage

Configure database

Choose your Subscription, Resource group, Database name and Server. No Elastic pool is needed. Click “Configure Database”



Configure

Basic
For less demanding workloads

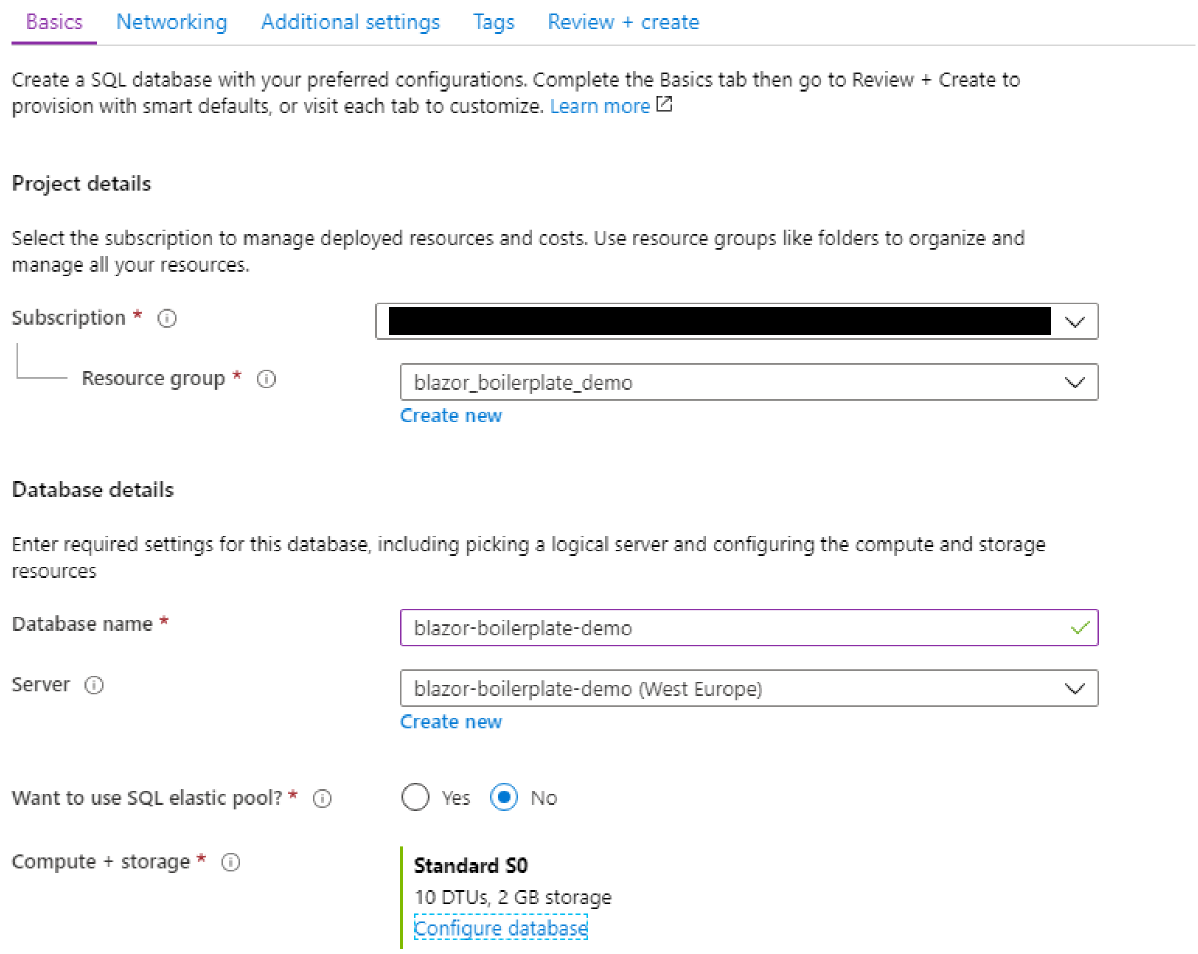
Standard
For workloads with typical performance requirements

DTUs

Data max size: 2 GB

Choose a database size that suites you. For this tutorial, we use Basic with 10 DTUs and a max size of 2GB

Click Apply



Create a SQL database with your preferred configurations. Complete the Basics tab then go to Review + Create to provision with smart defaults, or visit each tab to customize. Learn more

Project details

Select the subscription to manage deployed resources and costs. Use resource groups like folders to organize and manage all your resources.

Subscription * ⓘ [redacted]

Resource group * ⓘ blazor_boilerplate_demo

Create new

Database details

Enter required settings for this database, including picking a logical server and configuring the compute and storage resources

Database name * blazor-boilerplate-demo ✓

Server ⓘ blazor-boilerplate-demo (West Europe)

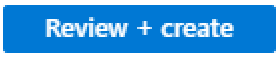
Create new

Want to use SQL elastic pool? * ⓘ ○ Yes ● No

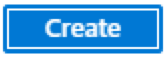
Compute + storage * ⓘ **Standard S0**

10 DTUs, 2 GB storage

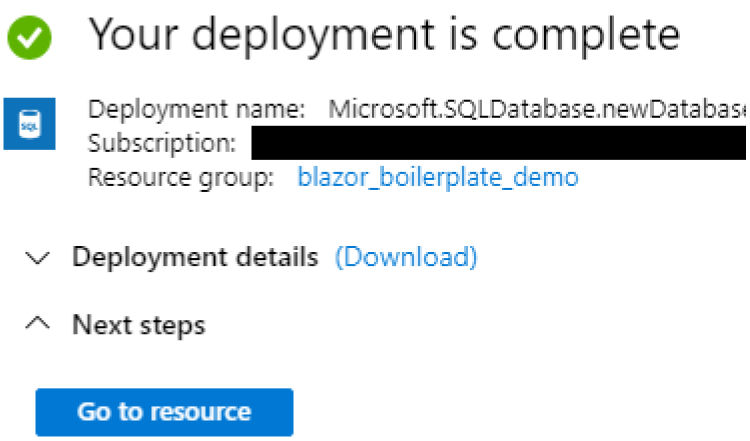
Configure database



Click “Review + create”



Click “Create”



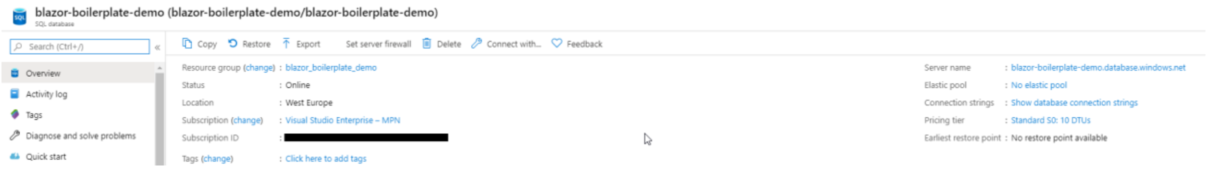
Your deployment is complete

Deployment name: Microsoft.SQLDatabase.newDatabas

Subscription: [redacted]

Resource group: blazor_boilerplate_demo

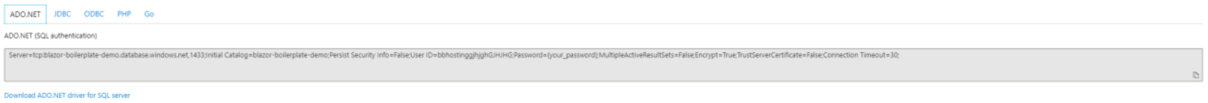
Wait for the deployment to complete and select “Go to resource”.



blazor-boilerplate-demo (blazor-boilerplate-demo/blazor-boilerplate-demo)
SQL database

Resource group (change)	: blazor_boilerplate_demo	Server name	: blazor-boilerplate-demo.database.windows.net
Status	: Online	Elastic pool	: No elastic pool
Location	: West Europe	Connection strings	: Show database connection strings
Subscription (change)	: Visual Studio Enterprise – MPN	Pricing tier	: Standard S0: 10 DTUs
Subscription ID	: [redacted]	Earliest restore point	: No restore point available
Tags (change)	: Click here to add tags		

Select “Show database connection strings” on the right hand side

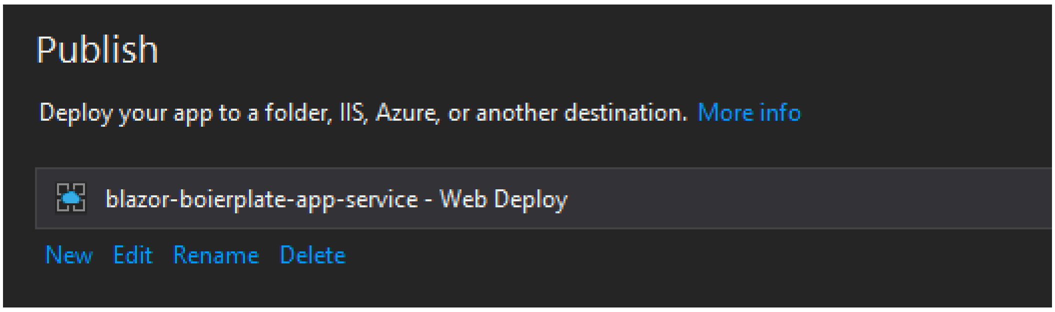


ADO.NET (SQL authentication)

```
Server=tcp:blazor-boilerplate-demo.database.windows.net,1433;Initial Catalog=blazor-boilerplate-demo;Persist Security Info=False;User ID=bbhostinggjhjghGJHJHG;Password={your_password};MultipleActiveResultSets=False;Encrypt=True;TrustServerCertificate=False;Connection Timeout=30;
```

Copy the connection string.

Back in Visual Studio, edit the publish profile

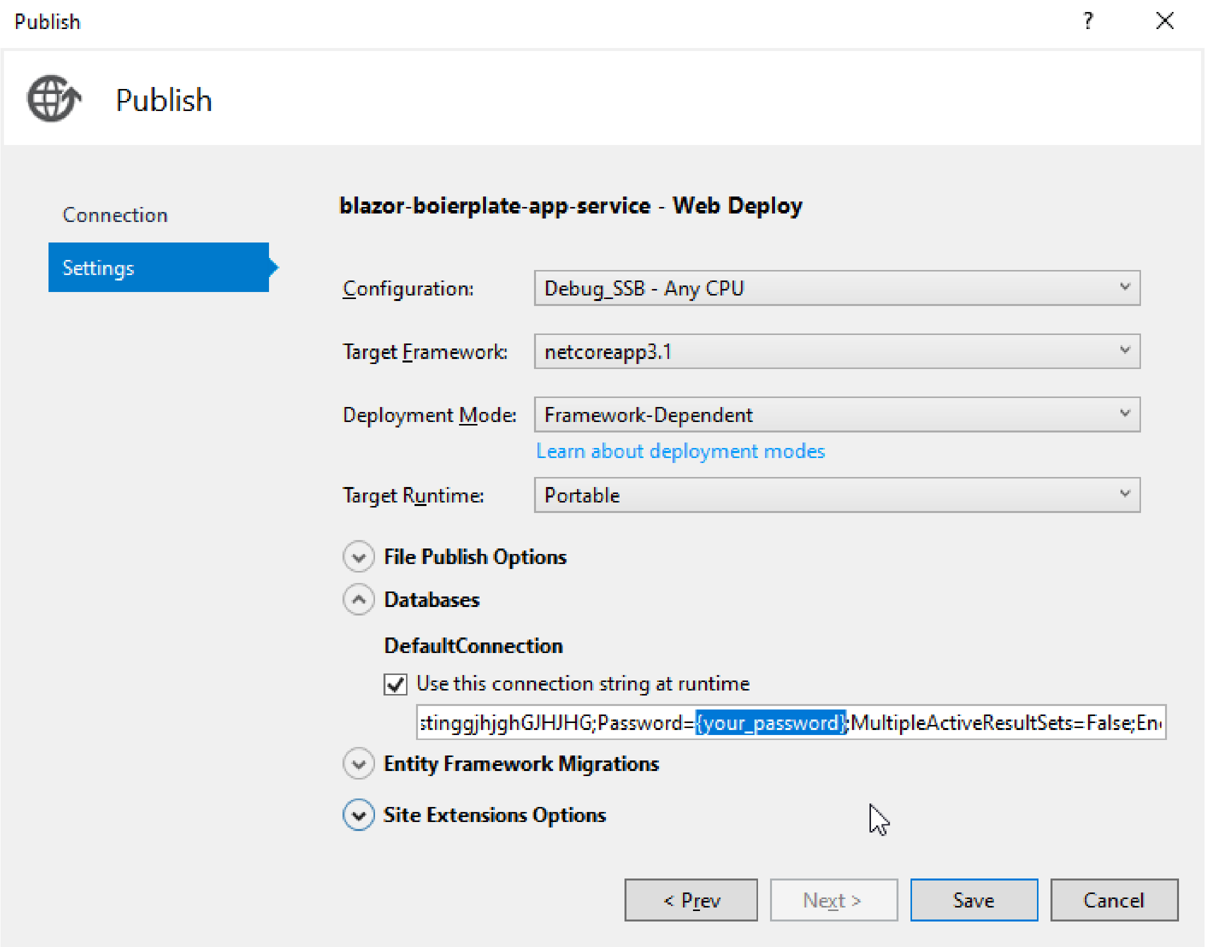


Publish

Deploy your app to a folder, IIS, Azure, or another destination. More info

blazor-boierplate-app-service - Web Deploy

Select the profile and click “Edit”



Publish

blazor-boierplate-app-service - Web Deploy

Configuration: Debug_SSB - Any CPU

Target Framework: netcoreapp3.1

Deployment Mode: Framework-Dependent

Learn about deployment modes

Target Runtime: Portable

File Publish Options

Databases

DefaultConnection

☑ Use this connection string at runtime

:tinggjhjghGJHJHG;Password={your_password};MultipleActiveResultSets=False;En

Entity Framework Migrations

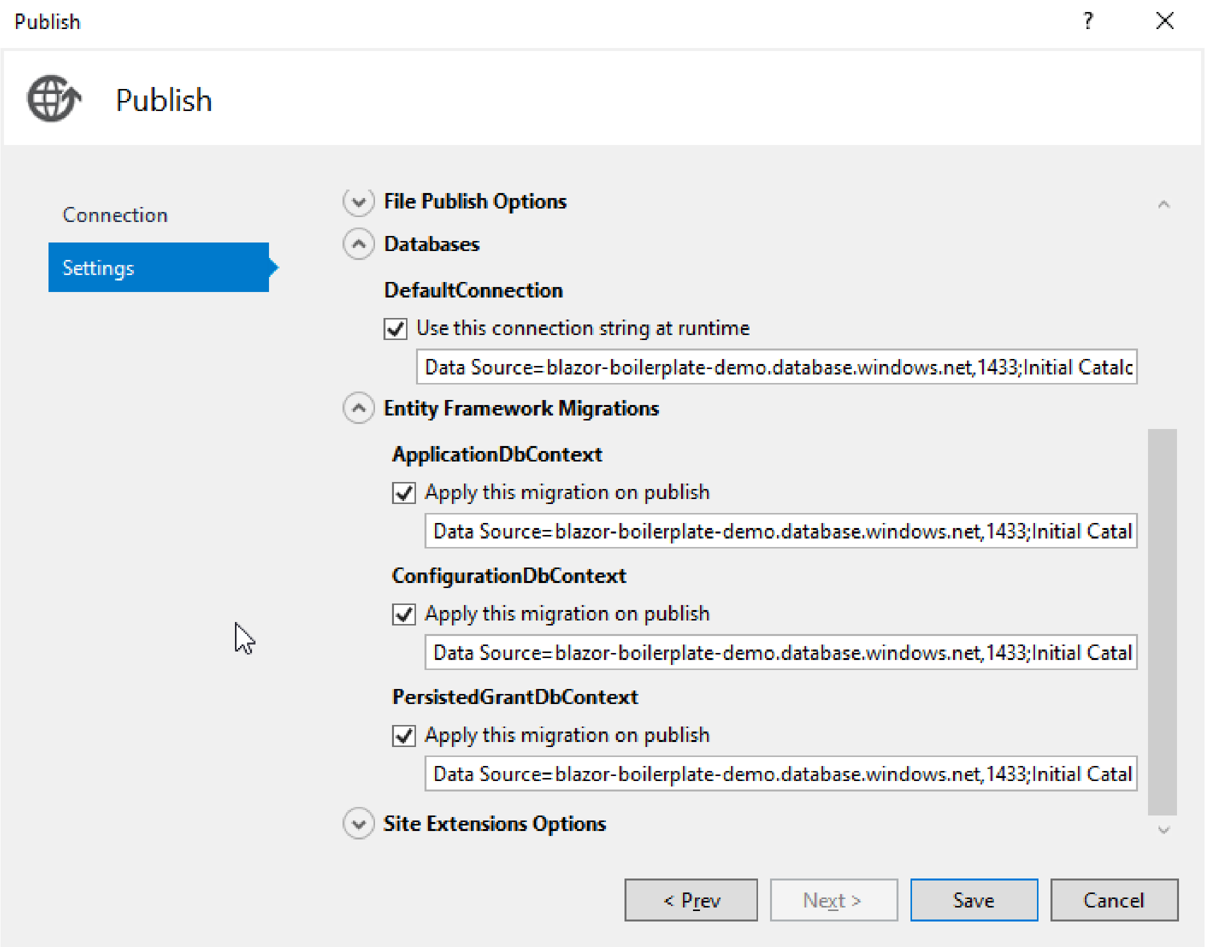
Site Extensions Options

Paste the Connection String into Settings -> Databases -> “DefaultConnection”. Insert your password.

Make sure that at the beginning of the connection string, you replace “Server=tcp:” with “Data Source=”

The final connection string should be something like this:

Data Source=blazor-boilerplate-demo.database.windows.net,1433;Initial Catalog=blazor_boilerplate;Persist Security Info=False;User ID=bbhostinggjhjghGJHJHG;Password=dshkuifsdfkhj//()(/&(/565zt;MultipleActiveResultSets=False;Encrypt=True;TrustServerCertificate=False;Connection Timeout=30;



DefaultConnection

☑ Use this connection string at runtime

Data Source=blazor-boilerplate-demo.database.windows.net,1433;Initial Catalc

Entity Framework Migrations

ApplicationDbContext

☑ Apply this migration on publish

Data Source=blazor-boilerplate-demo.database.windows.net,1433;Initial Catal

ConfigurationDbContext

☑ Apply this migration on publish

Data Source=blazor-boilerplate-demo.database.windows.net,1433;Initial Catal

PersistedGrantDbContext

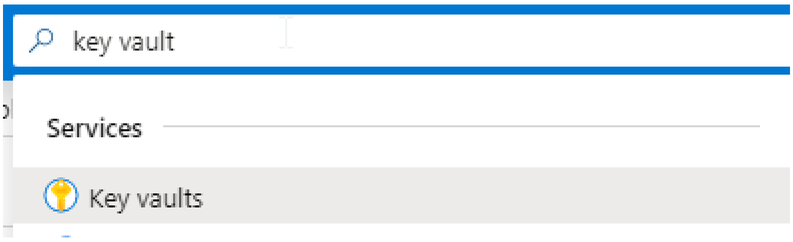
☑ Apply this migration on publish

Data Source=blazor-boilerplate-demo.database.windows.net,1433;Initial Catal

Copy and paste your resulting connection string into Settings -> Entity Framework Migrations for all connection strings

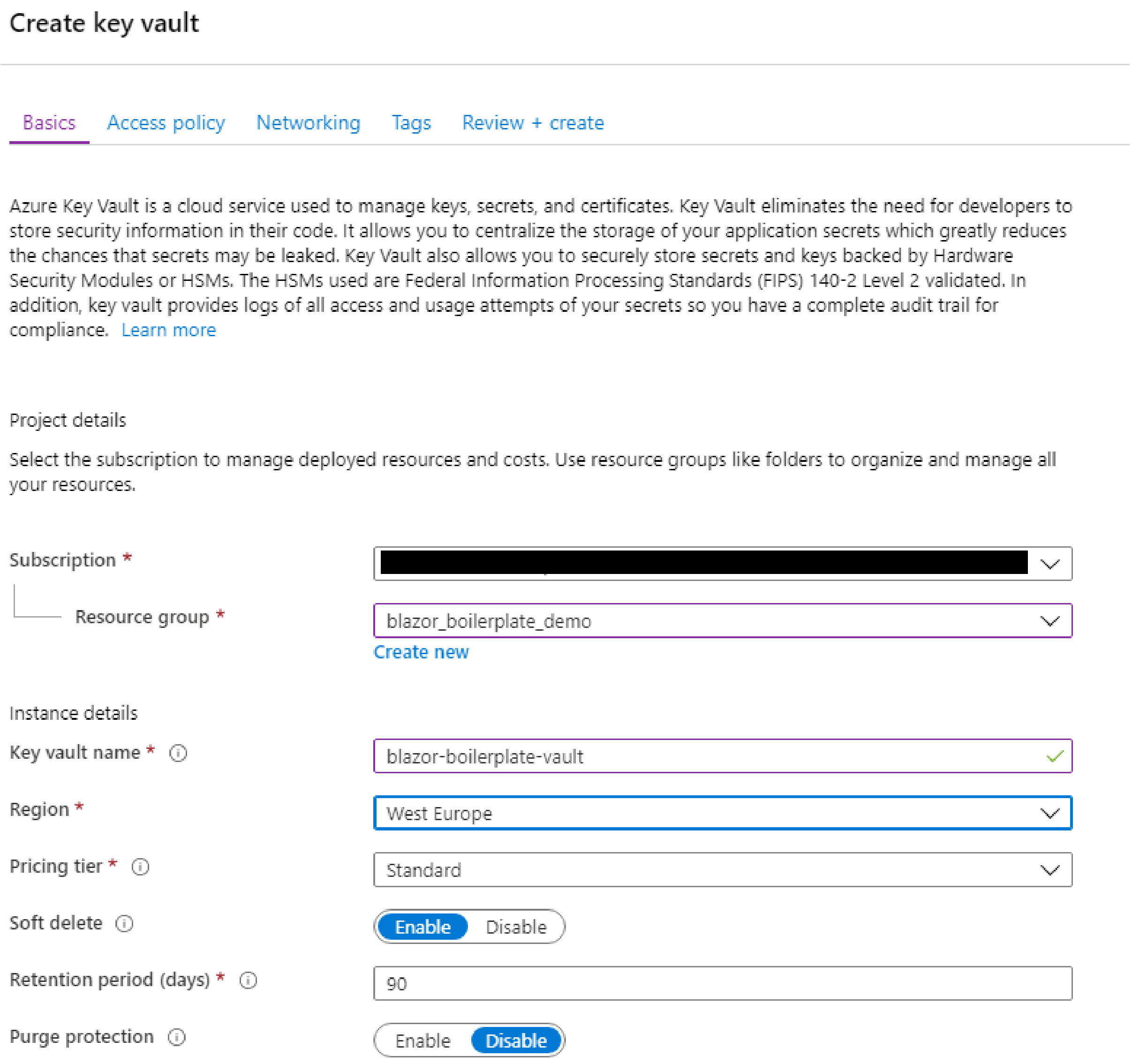
Create Key Vault

Switch to Azure. Search for “Key vaults”



Choose „Key vaults“

Click „Create key vault“



Create key vault

Azure Key Vault is a cloud service used to manage keys, secrets, and certificates. Key Vault eliminates the need for developers to store security information in their code. It allows you to centralize the storage of your application secrets which greatly reduces the chances that secrets may be leaked. Key Vault also allows you to securely store secrets and keys backed by Hardware Security Modules or HSMs. The HSMs used are Federal Information Processing Standards (FIPS) 140-2 Level 2 validated. In addition, key vault provides logs of all access and usage attempts of your secrets so you have a complete audit trail for compliance. Learn more

Project details

Select the subscription to manage deployed resources and costs. Use resource groups like folders to organize and manage all your resources.

Subscription * [redacted]

Resource group * blazor_boilerplate_demo

Create new

Instance details

Key vault name * blazor-boilerplate-vault ✓

Region * West Europe

Pricing tier * Standard

Soft delete [✓ Enable] Disable

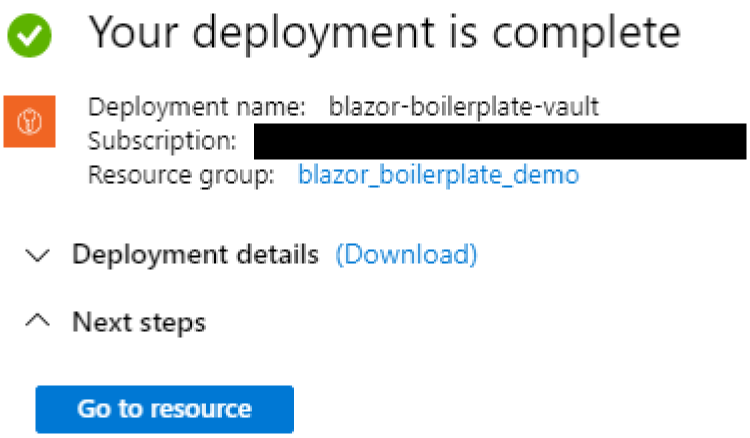
Retention period (days) * 90

Purge protection Enable [✓ Disable]

Select your subscription and Resource group. Choose a name, Region and Pricing tier.

Click “Review + create”

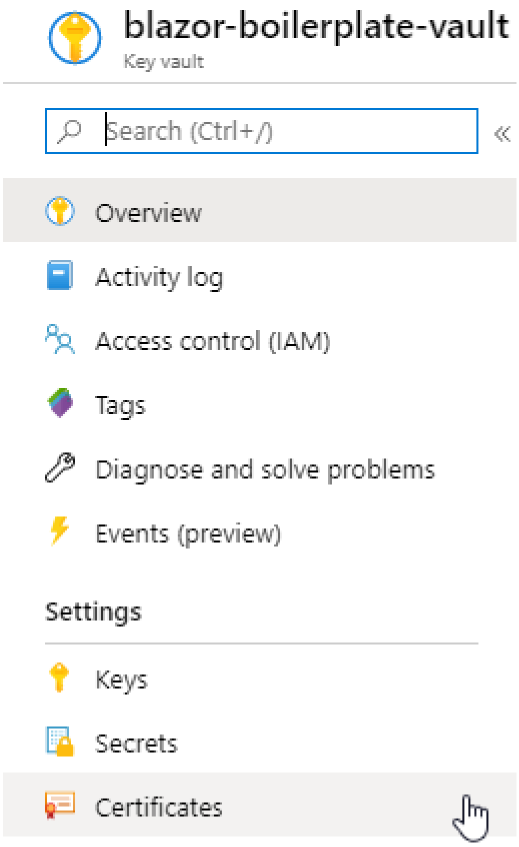
Click “Create”



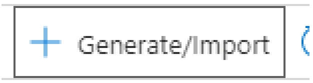
Your deployment is complete

Deployment name: blazor-boilerplate-vault
Subscription: [redacted]
Resource group: blazor_boilerplate_demo

After the deployment finished, select “Go to resource”.

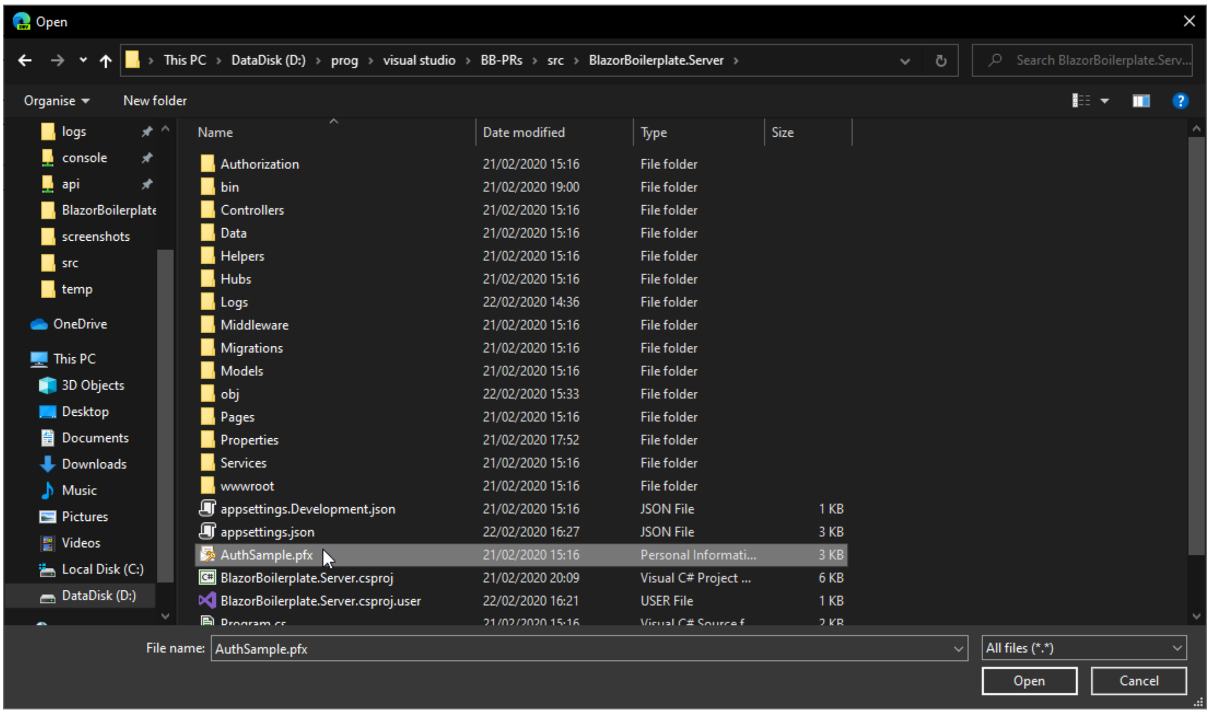


In the Key vault, select “Certificates”



Click „Generate/Import“.

Choose to Import a certificate.



Name	Date modified	Type	Size
Authorization	21/02/2020 15:16	File folder	
bin	21/02/2020 19:00	File folder	
Controllers	21/02/2020 15:16	File folder	
Data	21/02/2020 15:16	File folder	
Helpers	21/02/2020 15:16	File folder	
Hubs	21/02/2020 15:16	File folder	
Logs	22/02/2020 14:36	File folder	
Middleware	21/02/2020 15:16	File folder	
Migrations	21/02/2020 15:16	File folder	
Models	21/02/2020 15:16	File folder	
obj	22/02/2020 15:33	File folder	
Pages	21/02/2020 15:16	File folder	
Properties	21/02/2020 17:52	File folder	
Services	21/02/2020 15:16	File folder	
wwwroot	21/02/2020 15:16	File folder	
appsettings.Development.json	21/02/2020 15:16	JSON File	1 KB
appsettings.json	22/02/2020 16:27	JSON File	3 KB
AuthSample.pfx	21/02/2020 15:16	Personal Informati...	3 KB
BlazorBoilerplate.Server.csproj	21/02/2020 20:09	Visual C# Project ...	6 KB
BlazorBoilerplate.Server.csproj.user	22/02/2020 16:21	USER File	1 KB
Program.cs	21/02/2020 15:16	Visual C# Source f...	2 KB

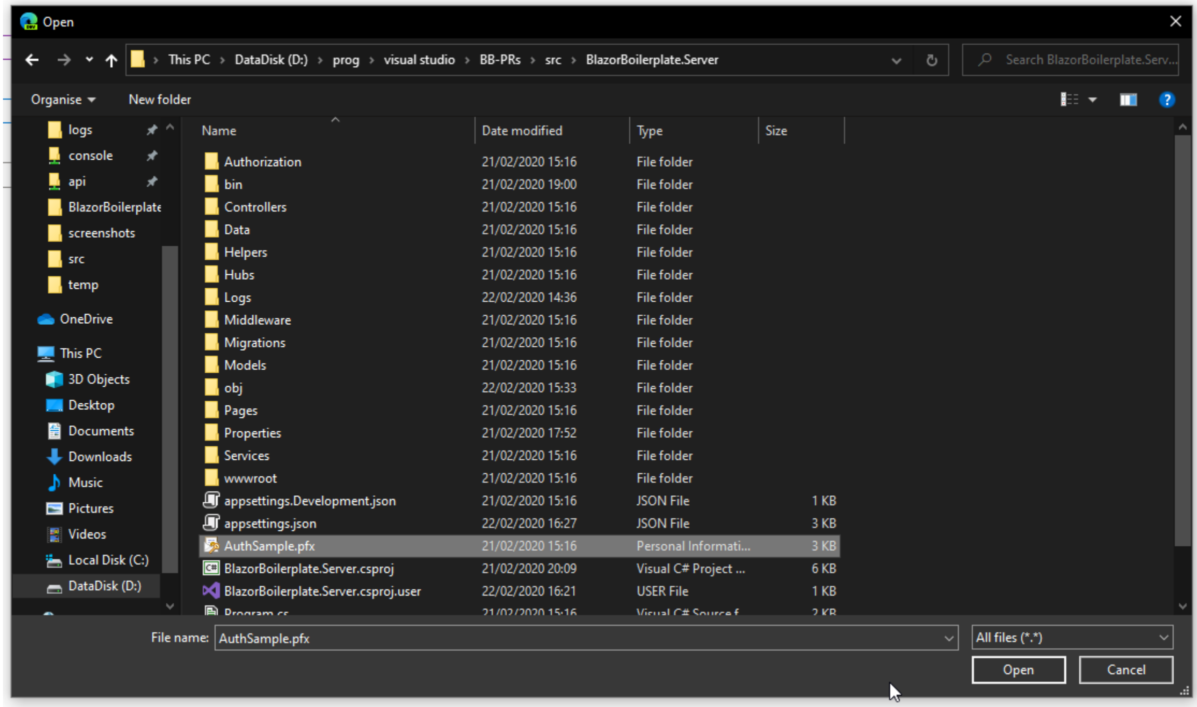
File name: AuthSample.pfx

All files (*.*)

Select your Certificate for Identity Server to use.

You can use the default “AuthSample.pfx”.

The certificate needs to have a private key marked as exportable.

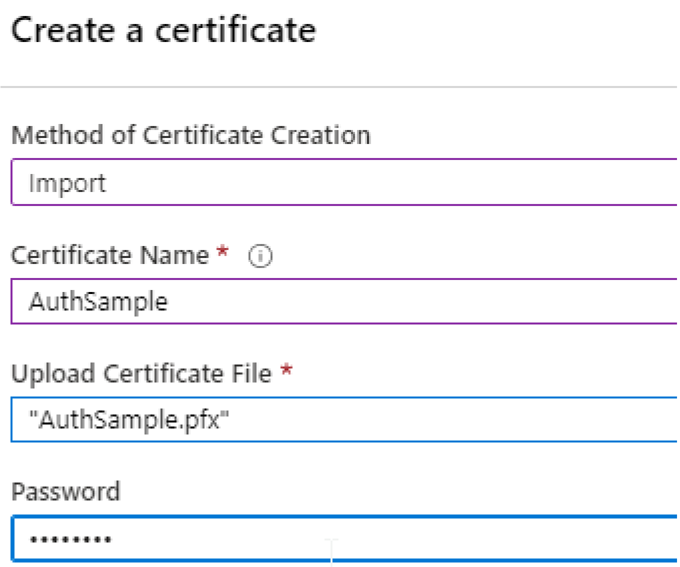

Open
This PC › DataDisk (D:) › prog › visual studio › BB-PRs › src › BlazorBoilerplate.Server

Name	Date modified	Type	Size
Authorization	21/02/2020 15:16	File folder	
bin	21/02/2020 19:00	File folder	
Controllers	21/02/2020 15:16	File folder	
Data	21/02/2020 15:16	File folder	
Helpers	21/02/2020 15:16	File folder	
Hubs	21/02/2020 15:16	File folder	
Logs	22/02/2020 14:36	File folder	
Middleware	21/02/2020 15:16	File folder	
Migrations	21/02/2020 15:16	File folder	
Models	21/02/2020 15:16	File folder	
obj	22/02/2020 15:33	File folder	
Pages	21/02/2020 15:16	File folder	
Properties	21/02/2020 17:52	File folder	
Services	21/02/2020 15:16	File folder	
wwwroot	21/02/2020 15:16	File folder	
appsettings.Development.json	21/02/2020 15:16	JSON File	1 KB
appsettings.json	22/02/2020 16:27	JSON File	3 KB
AuthSample.pfx	21/02/2020 15:16	Personal Informati...	3 KB
BlazorBoilerplate.Server.csproj	21/02/2020 20:09	Visual C# Project ...	6 KB
BlazorBoilerplate.Server.csproj.user	22/02/2020 16:21	USER File	1 KB
Program.cs	21/02/2020 15:16	Visual C# Source f...	2 KB

File name: AuthSample.pfx

All files (*.*)

Select your certificate and click “Open”.



Create a certificate

Method of Certificate Creation: Import

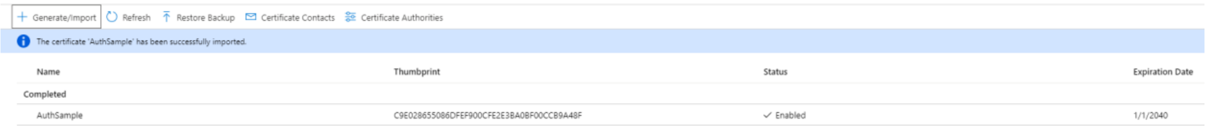
Certificate Name *: AuthSample

Upload Certificate File *: "AuthSample.pfx"

Password: ••••••••

Choose a name for the certificate and enter the password. For AuthSample.pfx, it is “Admin123”.

Click „Create“

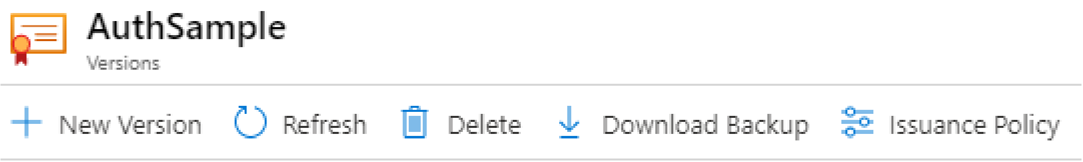


The certificate 'AuthSample' has been successfully imported.

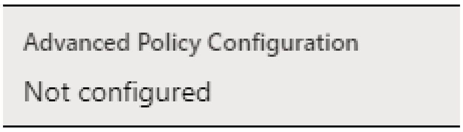
Name	Thumbprint	Status	Expiration Date
Completed			
AuthSample	C9E028655086DFEF900CFE2E3BA0BF00CCB9A48F	✓ Enabled	1/1/2040

Verify that the certificate has been imported correctly.

Click on the certificate in the list.

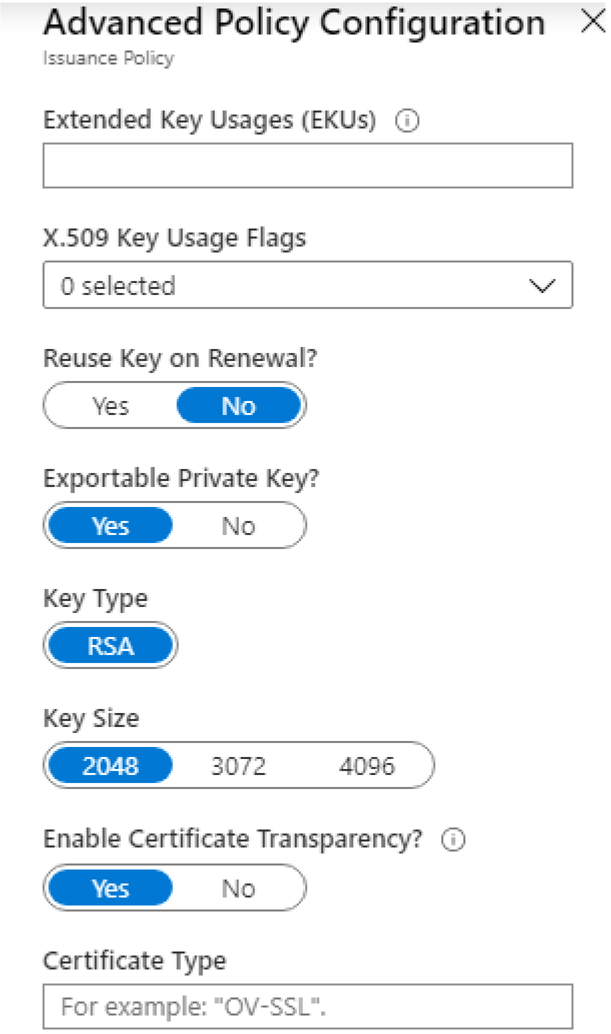


Click “Issuance Policy”



Advanced Policy Configuration: Not configured

Click “Advanced Policy Configuration”



Advanced Policy Configuration

Issuance Policy

Extended Key Usages (EKUs)

X.509 Key Usage Flags

0 selected

Reuse Key on Renewal?

Yes [✓ No]

Exportable Private Key?

[✓ Yes] No

Key Type

[✓ RSA]

Key Size

[✓ 2048] 3072 4096

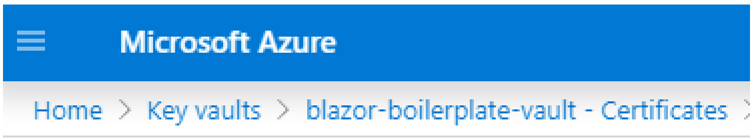
Enable Certificate Transparency?

[✓ Yes] No

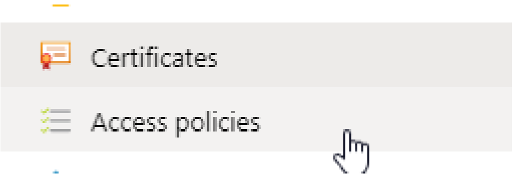
Certificate Type

For example: "OV-SSL".

Verify that the private key is marked as exportable.



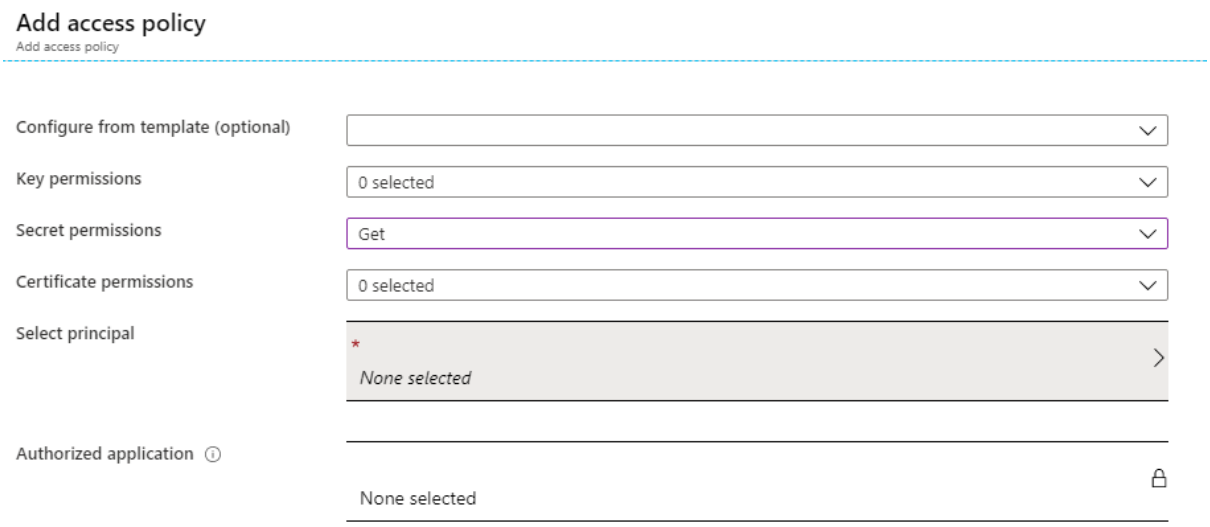
Click on your key vault on the top left.



On the menu at the left hand side, choose “Access policies”

+ Add Access Policy

Click „Add Access Policy“

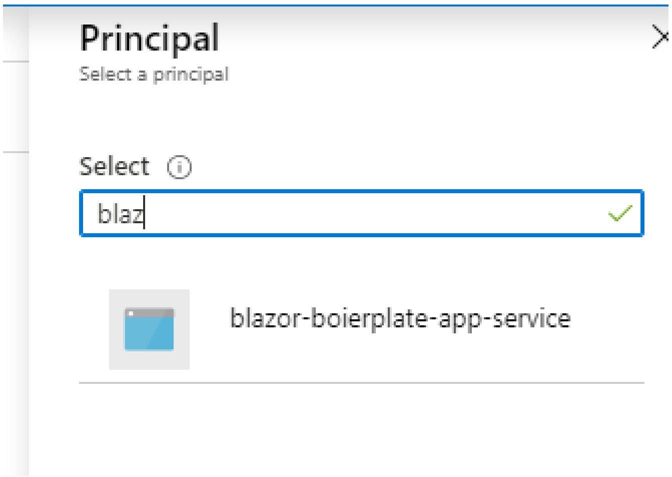

Add access policy

Add access policy

Configure from template (optional)	
Key permissions	0 selected
Secret permissions	Get
Certificate permissions	0 selected
Select principal	* None selected
Authorized application ⓘ	None selected

Select the „Get“ permission on “Secret permissions”. As Identity Server needs access to the private key, we will import the certificate as a secret – not as a certificate.

Click on “Select principal”


Principal

Select a principal

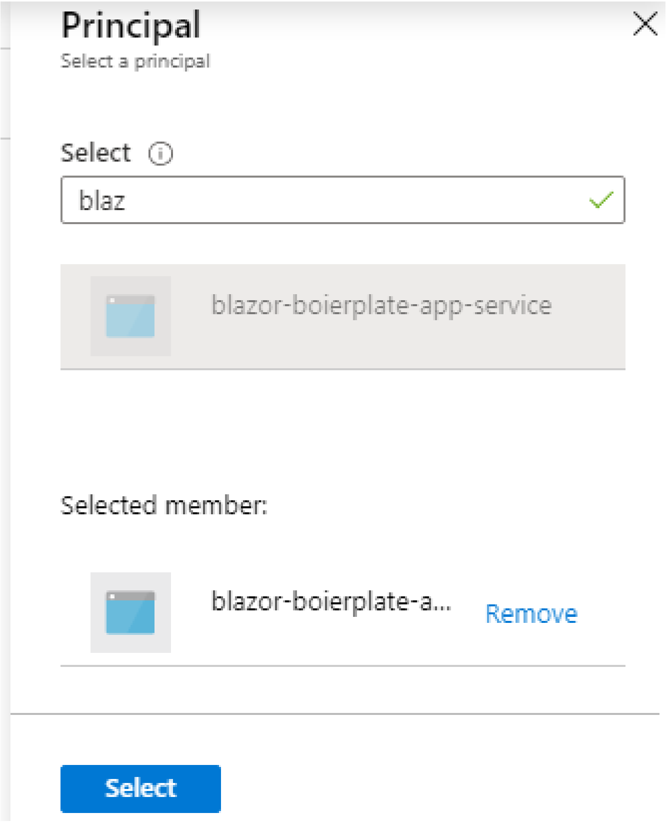
Select ⓘ

blaz

blazor-boierplate-app-service

Search for your app service by entering it’s name.

Click on the App Service.



Principal

Select a principal

Select ⓘ

blaz ✓

blazor-boierplate-app-service

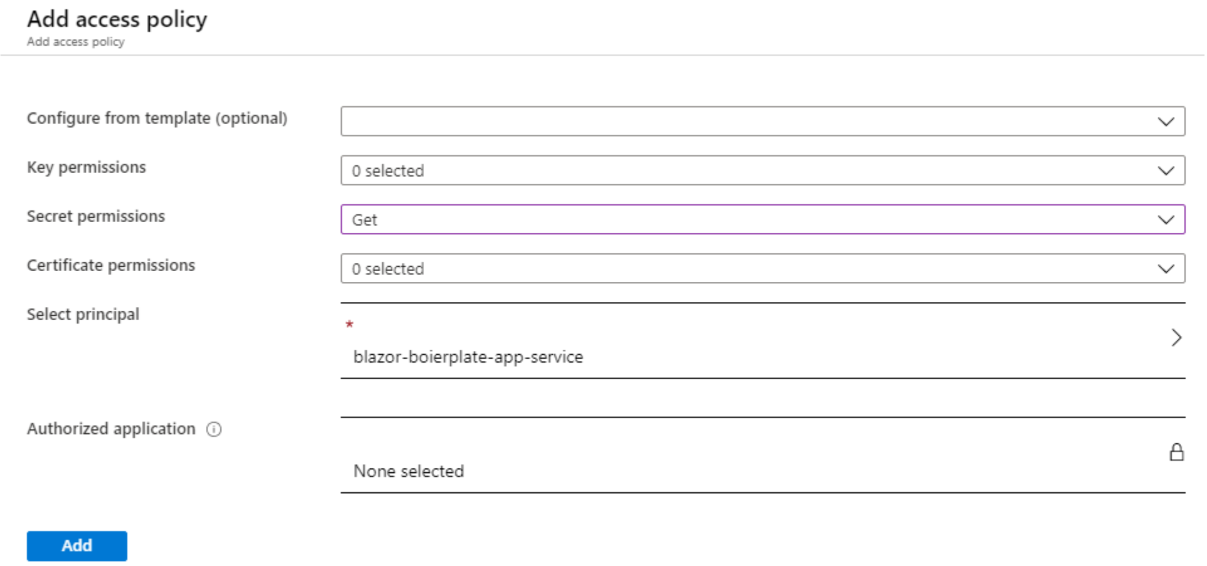
Selected member:

blazor-boierplate-a... Remove

Select

It will now appear on the bottom list.

Click “Select”.



Add access policy

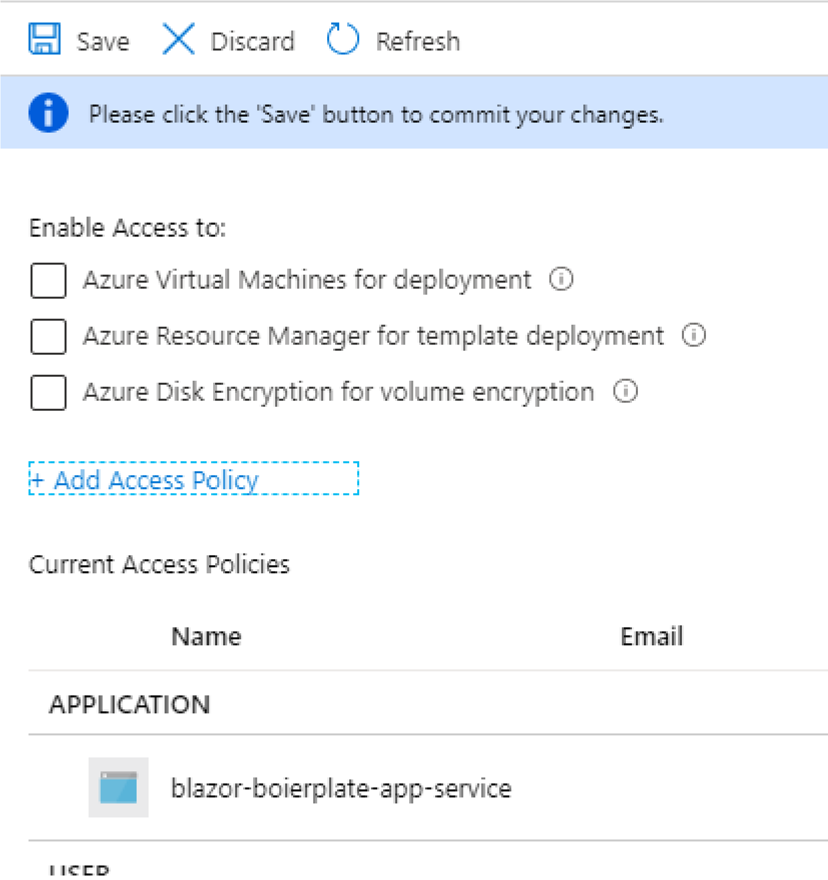
Add access policy

Configure from template (optional)	
Key permissions	0 selected
Secret permissions	Get
Certificate permissions	0 selected
Select principal	* blazor-boierplate-app-service
Authorized application ⓘ	None selected

Add

Verify that your app service now appears in “Select principal”.

Click “Add”.



Please click the 'Save' button to commit your changes.

Enable Access to:

☐ Azure Virtual Machines for deployment ⓘ

☐ Azure Resource Manager for template deployment ⓘ

☐ Azure Disk Encryption for volume encryption ⓘ

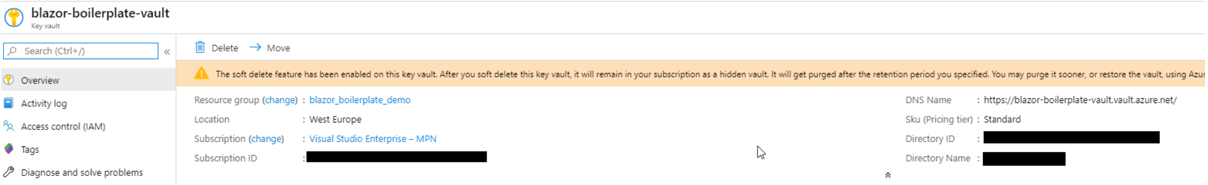
Current Access Policies

Name	Email
APPLICATION	
blazor-boierplate-app-service	
USER	

Click „Save“.

Use Key Vault in appsettings.json

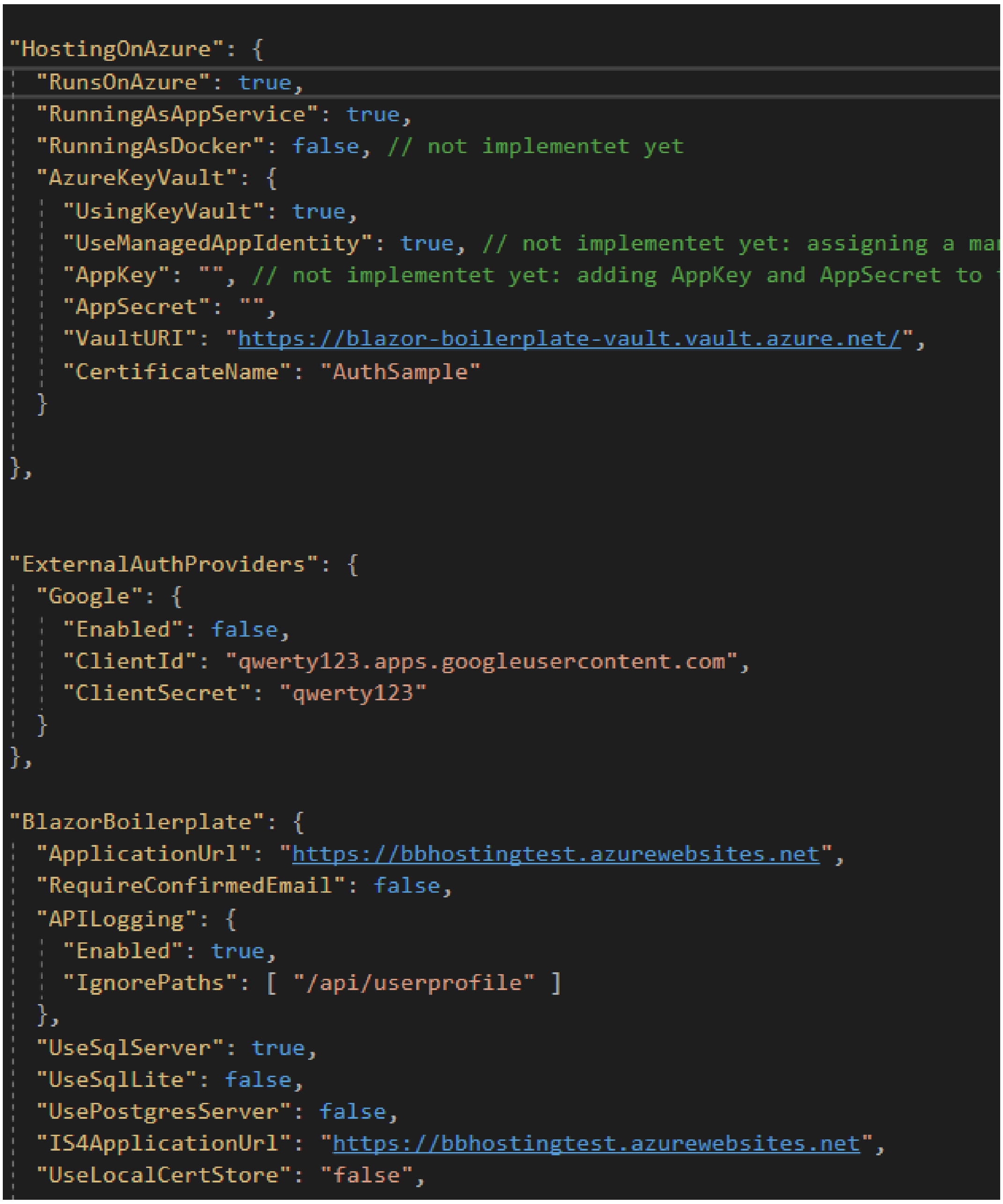
In the key vault, copy the DNS Name displayed on the right hand side.



The soft delete feature has been enabled on this key vault. After you soft delete this key vault, it will remain in your subscription as a hidden vault. It will get purged after the retention period you specified. You may purge it sooner, or restore the vault, using Azur

Resource group (change) : blazor_boilerplate_demo
Location : West Europe
Subscription (change) : Visual Studio Enterprise – MPN
Subscription ID : [redacted]

DNS Name : https://blazor-boilerplate-vault.vault.azure.net/
Sku (Pricing tier) : Standard
Directory ID : [redacted]
Directory Name : [redacted]



```
"HostingOnAzure": {
  "RunsOnAzure": true,
  "RunningAsAppService": true,
  "RunningAsDocker": false, // not implementet yet
  "AzureKeyVault": {
    "UsingKeyVault": true,
    "UseManagedAppIdentity": true, // not implementet yet: assigning a mar
    "AppKey": "", // not implementet yet: adding AppKey and AppSecret to
    "AppSecret": "",
    "VaultURI": "https://blazor-boilerplate-vault.vault.azure.net/",
    "CertificateName": "AuthSample"
  }

},


"ExternalAuthProviders": {
  "Google": {
    "Enabled": false,
    "ClientId": "qwerty123.apps.googleusercontent.com",
    "ClientSecret": "qwerty123"
  }
},

"BlazorBoilerplate": {
  "ApplicationUrl": "https://bbhostingtest.azurewebsites.net",
  "RequireConfirmedEmail": false,
  "APILogging": {
    "Enabled": true,
    "IgnorePaths": [ "/api/userprofile" ]
  },
  "UseSqlServer": true,
  "UseSqlLite": false,
  "UsePostgresServer": false,
  "IS4ApplicationUrl": "https://bbhostingtest.azurewebsites.net",
  "UseLocalCertStore": "false",
```

Back in Visual Studio, open appsettings.json, change “RunsOnAzure” to true and paste the Vault URL at “Vault URI”. Also fill in your Certificate name.
Change “UseLocalCertStore” to false.

Publish

Now publish your application. Once publishing is complete, it should automatically be launched inside a browser.

Troubleshooting

Using Kudu

For troubleshooting you can use an Azure Service called Kudu, to access your application’s log files.

Using Visual Studio

You can also access log files through Visual Studio integrated tools.