Publishing to Azure

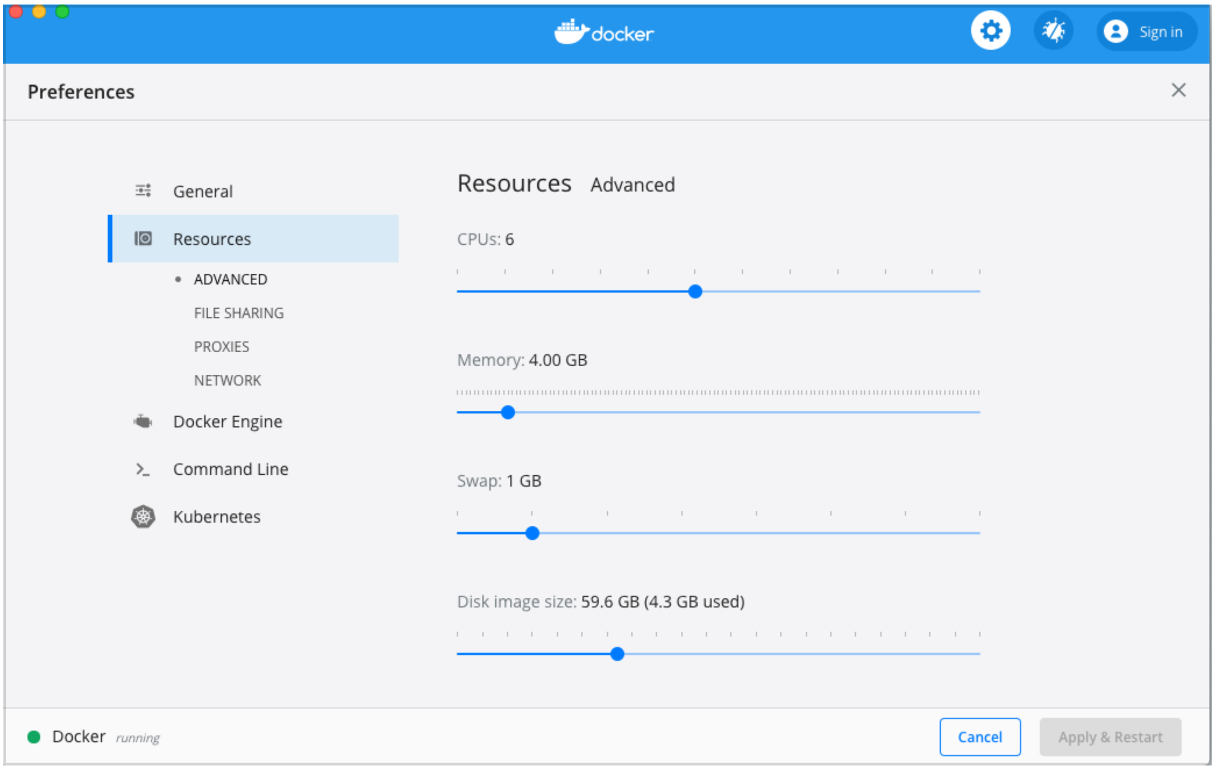
First of all we want to make sure our app runs without issue on Sql Server 2019. For windows you can just install this directly, but for Mac/Linux then you can get a docker image of SQL as Microsoft now has a Linux version of SQL. If you are on Windows but do not have SQL installed then so long as you have Docker then you can go ahead and do the same as me.

Setting up Sql Server for Development

Since SQL is a bit of a big install I'm going to download the files to my computer by running the following command:

```
docker pull microsoft/mssql-server-linux:latest
```

SQL Server requires a bit more memory than other DBs so I am also going to increase the memory for Docker in the preferences to 4GB:



Resources Advanced

CPUs: 6

Memory: 4.00 GB

Swap: 1 GB

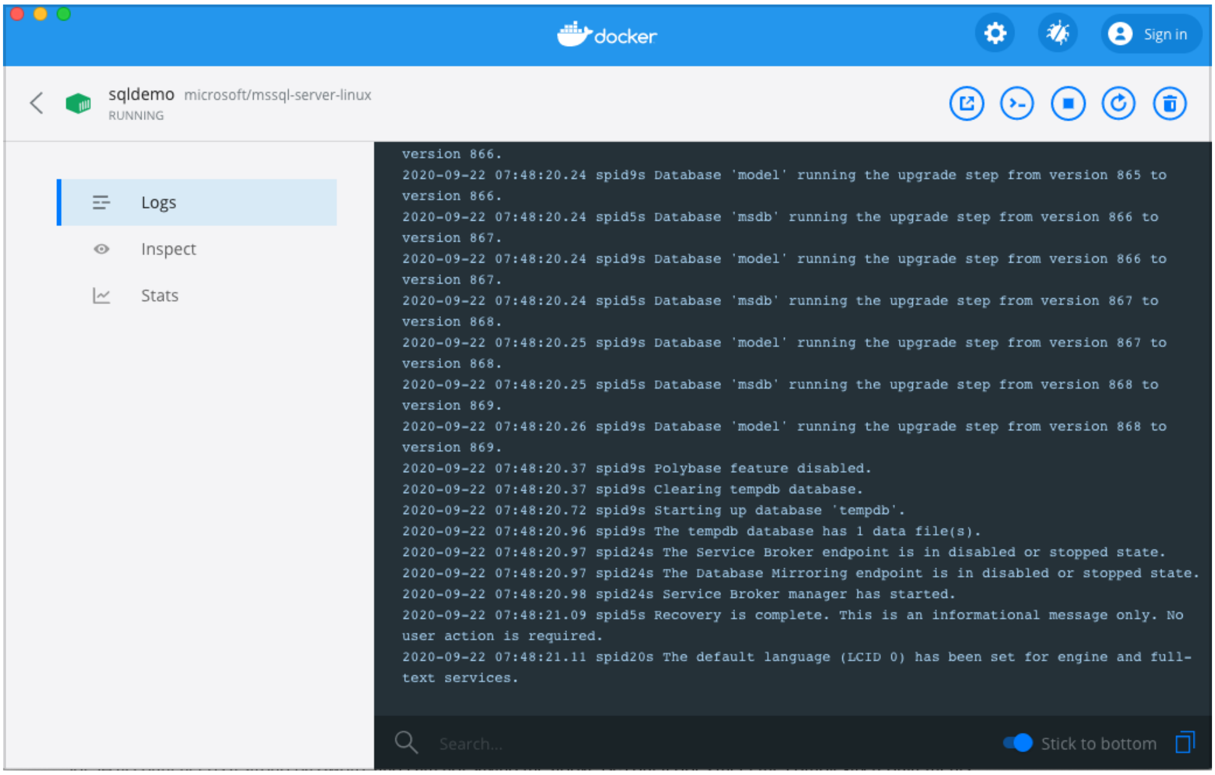
Disk image size: 59.6 GB (4.3 GB used)

Then we can run the following command to run the SQL Server:

```
docker run -d --name sqldemo -e 'ACCEPT_EULA=Y' -e 'SA_PASSWORD=Password1!' -p
1433:1433 microsoft/mssql-server-linux
```

Sql SA account needs a strong password, and I am not saying the above is(!) but it does meet the complexity requirements.

Should see this now in the docker dashboard:



```
version 866.
2020-09-22 07:48:20.24 spid9s Database 'model' running the upgrade step from version 865 to
version 866.
2020-09-22 07:48:20.24 spid5s Database 'msdb' running the upgrade step from version 866 to
version 867.
2020-09-22 07:48:20.24 spid9s Database 'model' running the upgrade step from version 866 to
version 867.
2020-09-22 07:48:20.24 spid5s Database 'msdb' running the upgrade step from version 867 to
version 868.
2020-09-22 07:48:20.25 spid9s Database 'model' running the upgrade step from version 867 to
version 868.
2020-09-22 07:48:20.25 spid5s Database 'msdb' running the upgrade step from version 868 to
version 869.
2020-09-22 07:48:20.26 spid9s Database 'model' running the upgrade step from version 868 to
version 869.
2020-09-22 07:48:20.37 spid9s Polybase feature disabled.
2020-09-22 07:48:20.37 spid9s Clearing tempdb database.
2020-09-22 07:48:20.72 spid9s Starting up database 'tempdb'.
2020-09-22 07:48:20.96 spid9s The tempdb database has 1 data file(s).
2020-09-22 07:48:20.97 spid24s The Service Broker endpoint is in disabled or stopped state.
2020-09-22 07:48:20.97 spid24s The Database Mirroring endpoint is in disabled or stopped state.
2020-09-22 07:48:20.98 spid24s Service Broker manager has started.
2020-09-22 07:48:21.09 spid5s Recovery is complete. This is an informational message only. No
user action is required.
2020-09-22 07:48:21.11 spid20s The default language (LCID 0) has been set for engine and full-
text services.
```

Switching to use SQL Server for dev.

I'm going to create a new branch so that I do not interfere with the master branch.   Run the following:

```
git checkout -b AzurePublish
```

Add the following Sql Server provider via Nuget:

Microsoft.EntityFrameworkCore.SqlServer

You can remove the package for Sqlite and Postgres if you still have them installed - we only need SqlServer for this

Ensure you pick the same version as your runtime.

Open the **appsettings.development.json** and change the default connection string to the following:

```
"ConnectionStrings" : {
    "DefaultConnection": "Server=localhost; User Id=sa; Password=Password1!;
Database=datingappdb"
  },
```

Update the ApplicationServiceExtensions to use this:

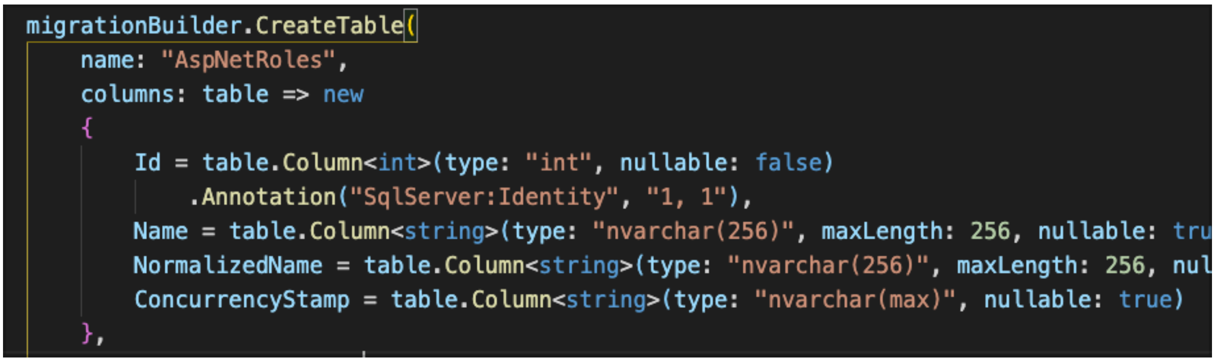
```
            services.AddDbContext<DataContext>(options =>
            {

options.UseSqlServer(config.GetConnectionString("DefaultConnection"));
            });
```

Delete the migrations folder from Data/Migrations and create a new migration for the Sql Server provider:

```
dotnet ef migrations add SqlInitial -o Data/Migrations
```

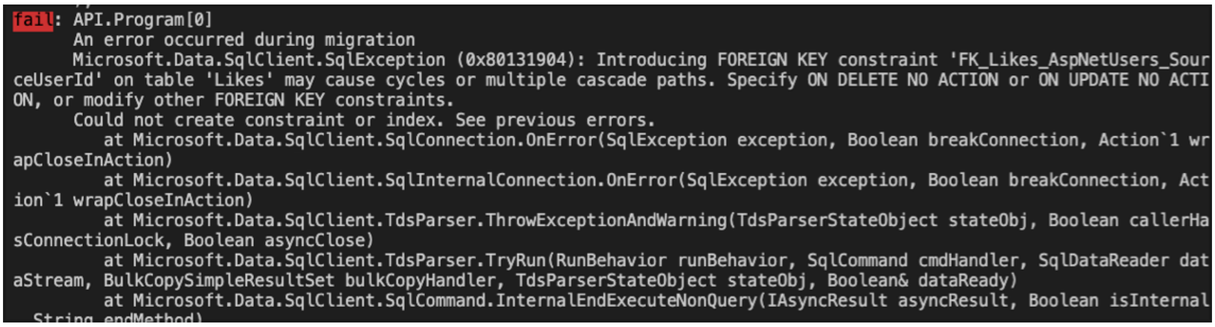
Check the migration and ensure you can see Sql server specific annotations in there:



```
migrationBuilder.CreateTable(
    name: "AspNetRoles",
    columns: table => new
    {
        Id = table.Column<int>(type: "int", nullable: false)
            .Annotation("SqlServer:Identity", "1, 1"),
        Name = table.Column<string>(type: "nvarchar(256)", maxLength: 256, nullable: tru
        NormalizedName = table.Column<string>(type: "nvarchar(256)", maxLength: 256, nul
        ConcurrencyStamp = table.Column<string>(type: "nvarchar(max)", nullable: true)
    },
```

Restart the app and make sure everything works!

Well, it doesn't because Sql server is special:



```
fail: API.Program[0]
      An error occurred during migration
      Microsoft.Data.SqlClient.SqlException (0x80131904): Introducing FOREIGN KEY constraint 'FK_Likes_AspNetUsers_Sour
ceUserId' on table 'Likes' may cause cycles or multiple cascade paths. Specify ON DELETE NO ACTION or ON UPDATE NO ACTI
ON, or modify other FOREIGN KEY constraints.
      Could not create constraint or index. See previous errors.
         at Microsoft.Data.SqlClient.SqlConnection.OnError(SqlException exception, Boolean breakConnection, Action`1 wr
apCloseInAction)
         at Microsoft.Data.SqlClient.SqlInternalConnection.OnError(SqlException exception, Boolean breakConnection, Act
ion`1 wrapCloseInAction)
         at Microsoft.Data.SqlClient.TdsParser.ThrowExceptionAndWarning(TdsParserStateObject stateObj, Boolean callerHa
sConnectionLock, Boolean asyncClose)
         at Microsoft.Data.SqlClient.TdsParser.TryRun(RunBehavior runBehavior, SqlCommand cmdHandler, SqlDataReader dat
aStream, BulkCopySimpleResultSet bulkCopyHandler, TdsParserStateObject stateObj, Boolean& dataReady)
         at Microsoft.Data.SqlClient.SqlCommand.InternalEndExecuteNonQuery(IAsyncResult asyncResult, Boolean isInternal
   String endMethod)
```

So we need to add an extra bit of configuration here to the DataContext.cs class and make sure one of the 'UserLike' specifies no action for the delete behaviour:

```
            builder.Entity<UserLike>()
                .HasOne(s => s.SourceUser)
                .WithMany(l => l.LikedUsers)
                .HasForeignKey(s => s.SourceUserId)
                .OnDelete(DeleteBehavior.NoAction);

            builder.Entity<UserLike>()
                .HasOne(s => s.LikedUser)
                .WithMany(l => l.LikedByUsers)
                .HasForeignKey(s => s.LikedUserId)
                .OnDelete(DeleteBehavior.Cascade);
```

Delete the migrations folder and recreate the migration

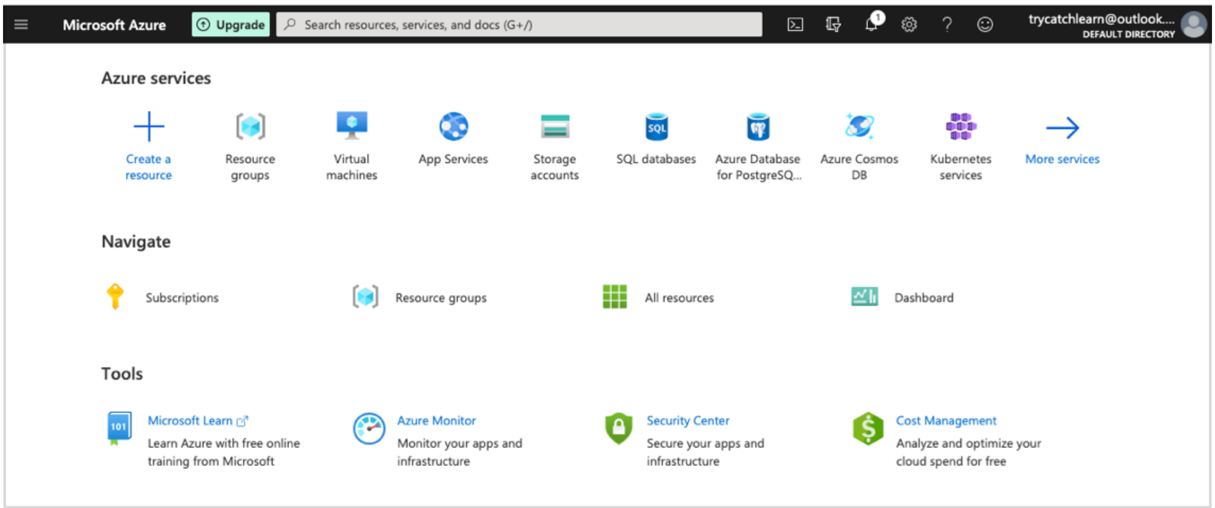
```
dotnet ef migrations add SqlInitial -o Data/Migrations
```

Restart the app again and make sure we have success!

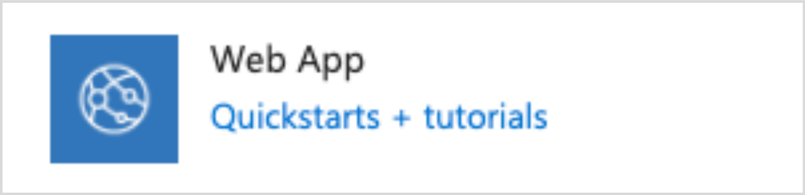
This time everything goes smoothly. Restart the angular app and ensure we can operate the application without any errors on localhost 4200. It should work fine.

Creating an app service in Azure

Now is time to go and create an app in Azure. If you have never used Azure before then you will be able to use $150 worth of resources for the first 12 months. You will still need to provide credit card info though which is why this is not part of the main course. Once you are in you should see something like this:

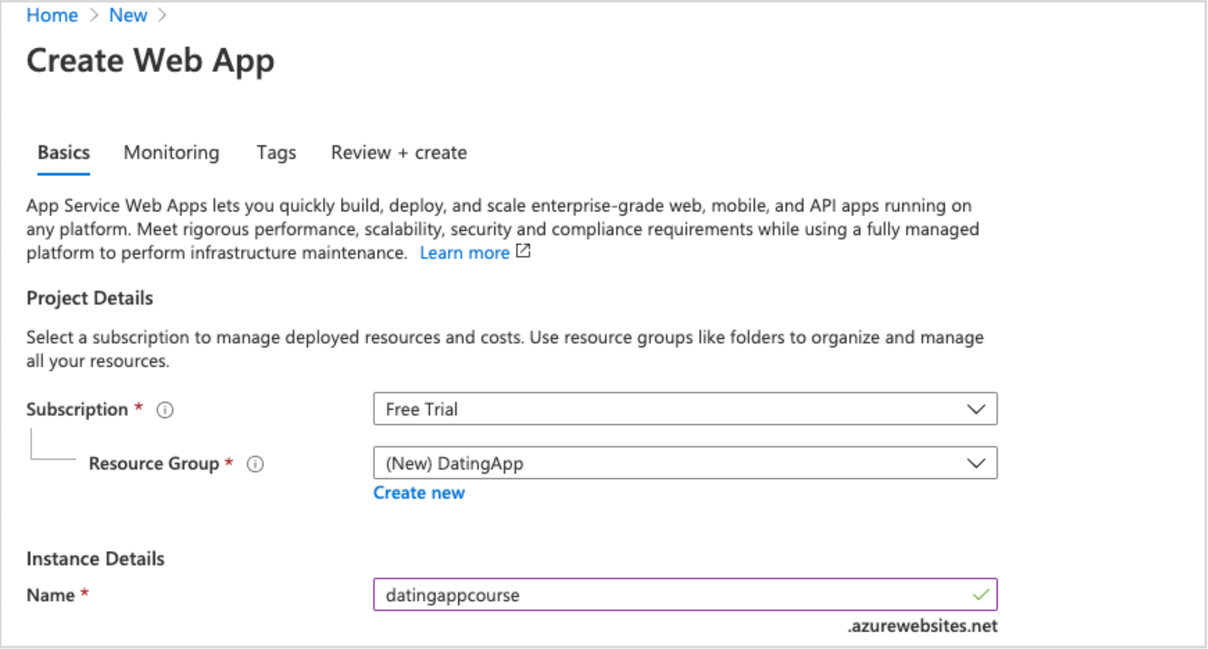


From here we want to create a new resource, so hit the 'Create resource' button, then select 'Web App' from the list:



We then need to complete a form (yay!). You will need to create a resource group which is just a container for… resources.

These are the options I've selected - the goal here is to select the absolute cheapest or free options available so you will need to be careful as obviously Microsoft will select some expensive defaults for you if you just accept them without changing.



Create Web App

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.

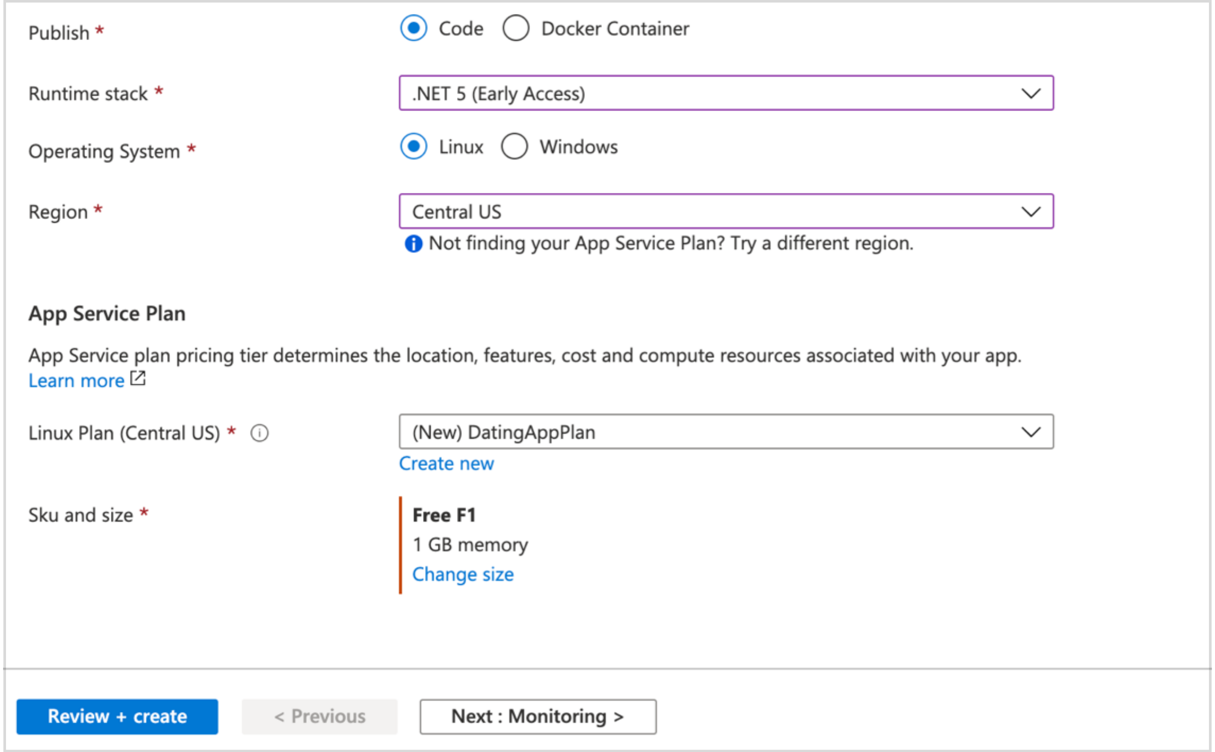
Subscription * : Free Trial

Resource Group * : (New) DatingApp

Create new

Instance Details

Name * : datingappcourse ✓ .azurewebsites.net



Publish * : ● Code ○ Docker Container

Runtime stack * : .NET 5 (Early Access)

Operating System * : ● Linux ○ Windows

Region * : Central US

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

Linux Plan (Central US) * : (New) DatingAppPlan

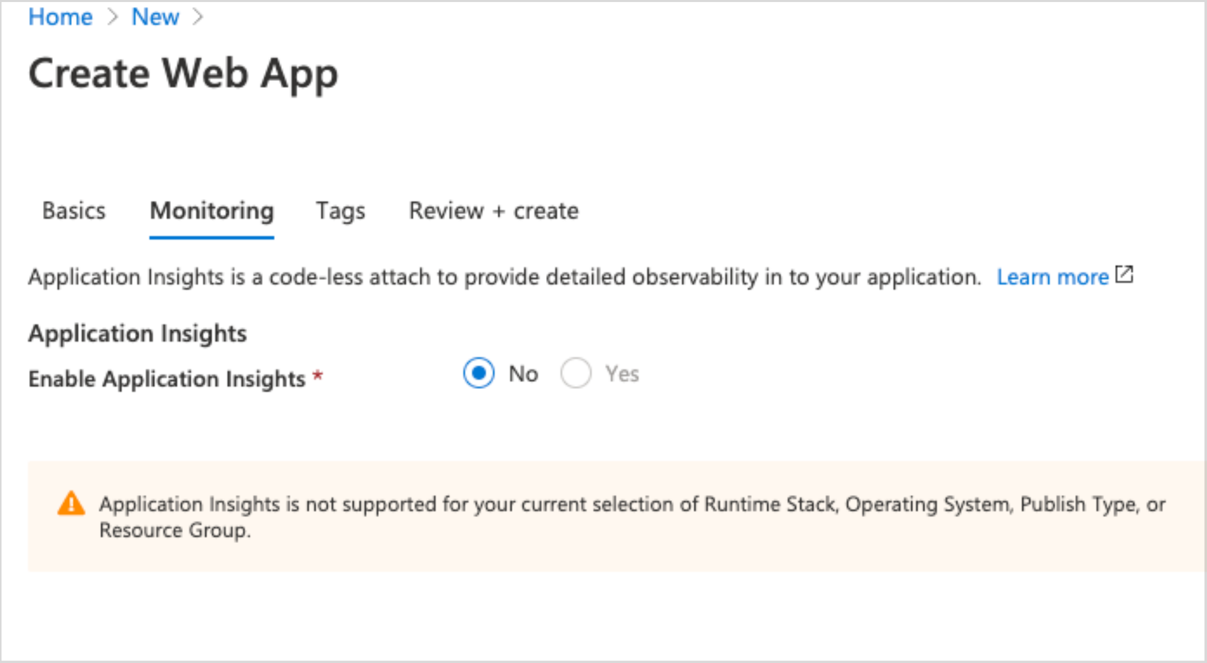
Create new

Sku and size * : Free F1 1 GB memory

Change size

Note that .Net 5 is still in “Early Access”, but this is what we are looking for.

Click the next button… Monitoring:



Create Web App

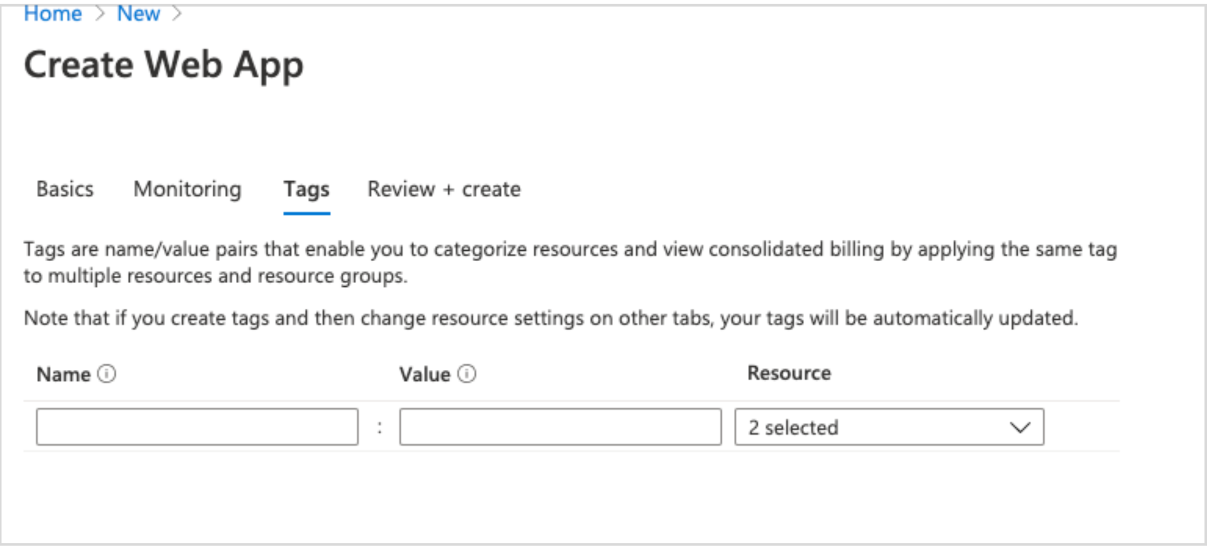
Application Insights is a code-less attach to provide detailed observability in to your application. Learn more

Application Insights

Enable Application Insights * ● No ○ Yes

Application Insights is not supported for your current selection of Runtime Stack, Operating System, Publish Type, or Resource Group.

Not much choice here. So either the Linux config does not allow this or (more likely) this is not available for freebies. Click Next: tags



Create Web App

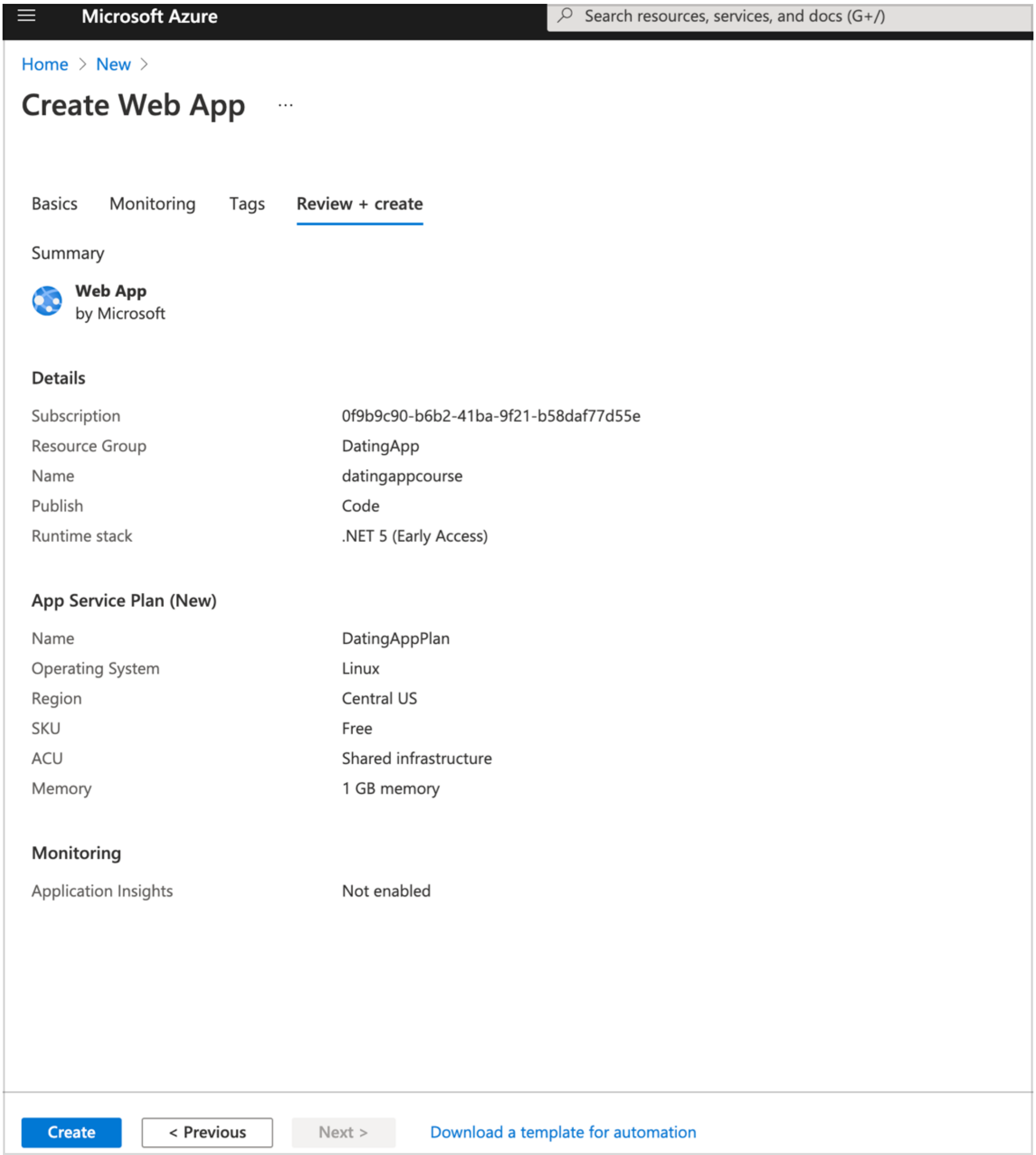
Tags are name/value pairs that enable you to categorize resources and view consolidated billing by applying the same tag to multiple resources and resource groups.

Note that if you create tags and then change resource settings on other tabs, your tags will be automatically updated.

Name		Value	Resource
	:		2 selected

I'm leaving this blank. Next: Review + Create (finally!)

Then we get to review our selections:



Summary

Web App
by Microsoft

Details

Subscription	0f9b9c90-b6b2-41ba-9f21-b58daf77d55e
Resource Group	DatingApp
Name	datingappcourse
Publish	Code
Runtime stack	.NET 5 (Early Access)

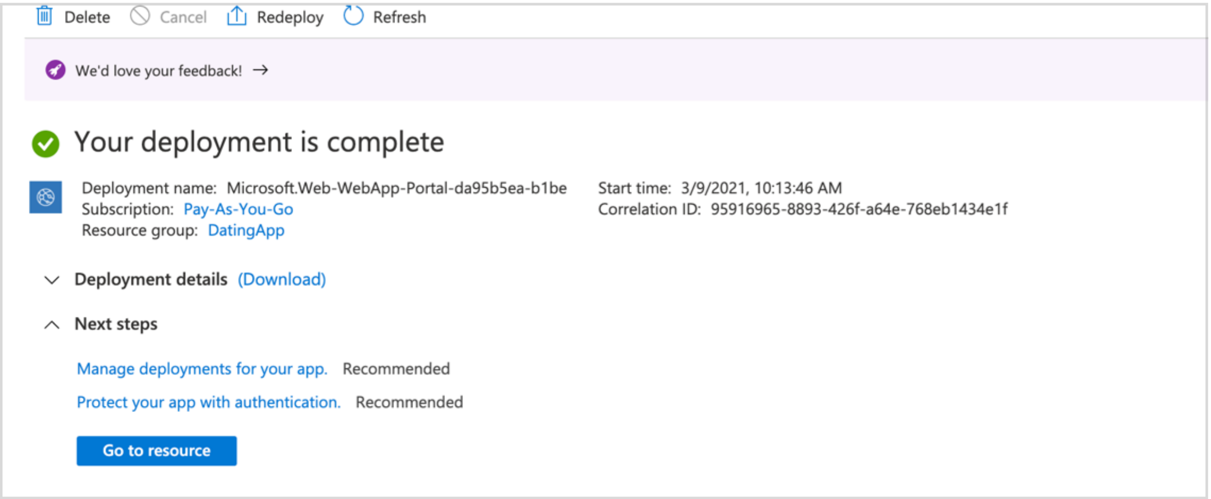
App Service Plan (New)

Name	DatingAppPlan
Operating System	Linux
Region	Central US
SKU	Free
ACU	Shared infrastructure
Memory	1 GB memory

Monitoring

Application Insights	Not enabled

Then we get to hit the Create button and go make a coffee whilst it does its thing, it will probably take a few minutes. Once its done we should see this:



Your deployment is complete

Deployment name: Microsoft.Web-WebApp-Portal-da95b5ea-b1be
Subscription: Pay-As-You-Go
Resource group: DatingApp

Start time: 3/9/2021, 10:13:46 AM
Correlation ID: 95916965-8893-426f-a64e-768eb1434e1f

Deployment details (Download)

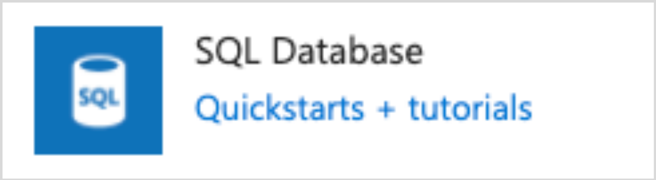
Next steps

Manage deployments for your app. Recommended

Protect your app with authentication. Recommended

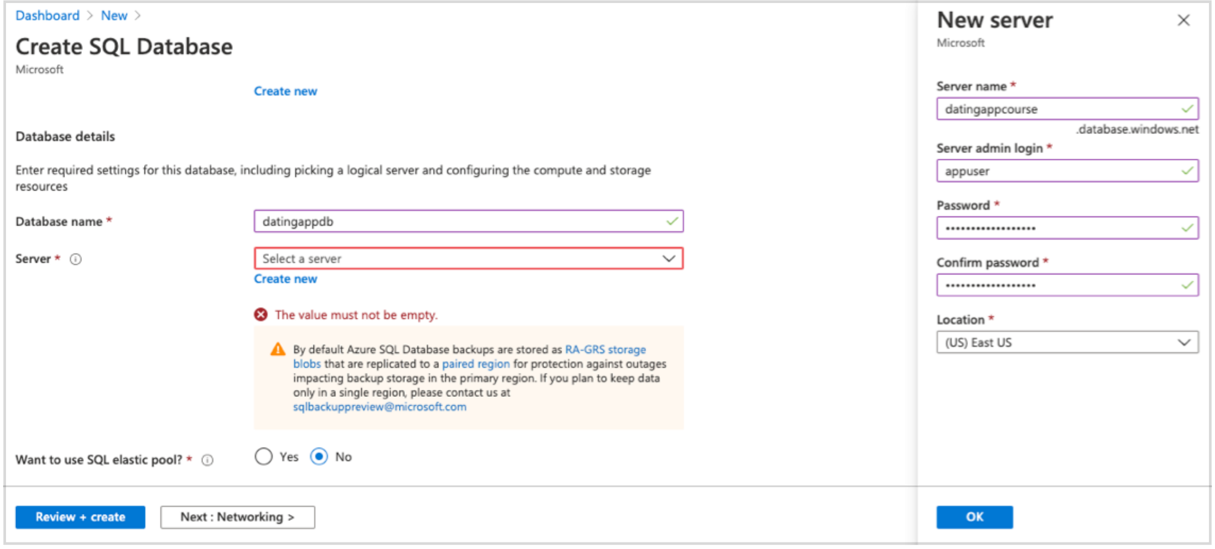
Now we need a Database server. Click the hamburger icon in the top left and hit the 'Create a resource' link.

Select the SQL Database:



We now have another form to fill out (yay!). Once again the goal here is not to accept the defaults but seek out the cheapest (or free) option available. The options here will be different depending on what your subscription here is so proceed with utmost caution unless you have more money than you know what to do with in which case the defaults will be fine.

We need to create a server here for our Database that has a unique name on Azure, as well as an admin account that cannot use the name 'admin' so I've gone for appuser as my admin account:



Dashboard > New >

Create SQL Database

Microsoft

Create new

Database details

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

Database name * datingappdb

Server * Select a server

Create new

The value must not be empty.

By default Azure SQL Database backups are stored as RA-GRS storage blobs that are replicated to a paired region for protection against outages impacting backup storage in the primary region. If you plan to keep data only in a single region, please contact us at sqlbackuppreview@microsoft.com

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

New server

Microsoft

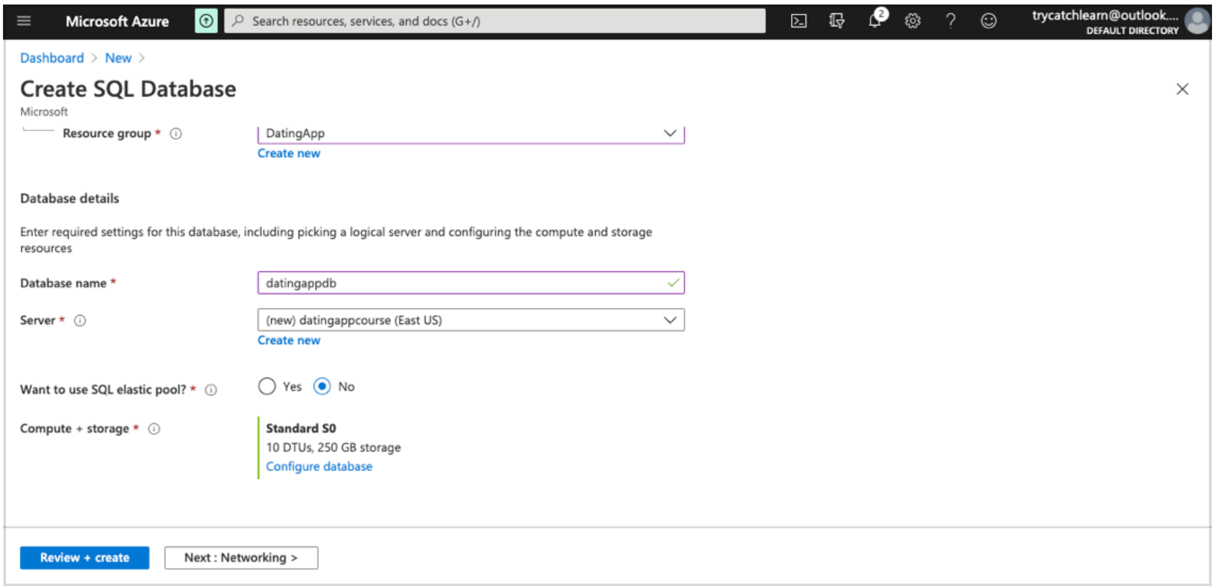
Server name * datingappcourse .database.windows.net

Server admin login * appuser

Password * ••••••••••••••••••

Confirm password * ••••••••••••••••••

Location * (US) East US



Microsoft Azure
Search resources, services, and docs (G+/)
trycatchlearn@outlook....
DEFAULT DIRECTORY

Dashboard > New >

Create SQL Database

Microsoft

Resource group * DatingApp

Create new

Database details

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

Database name * datingappdb

Server * (new) datingappcourse (East US)

Create new

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

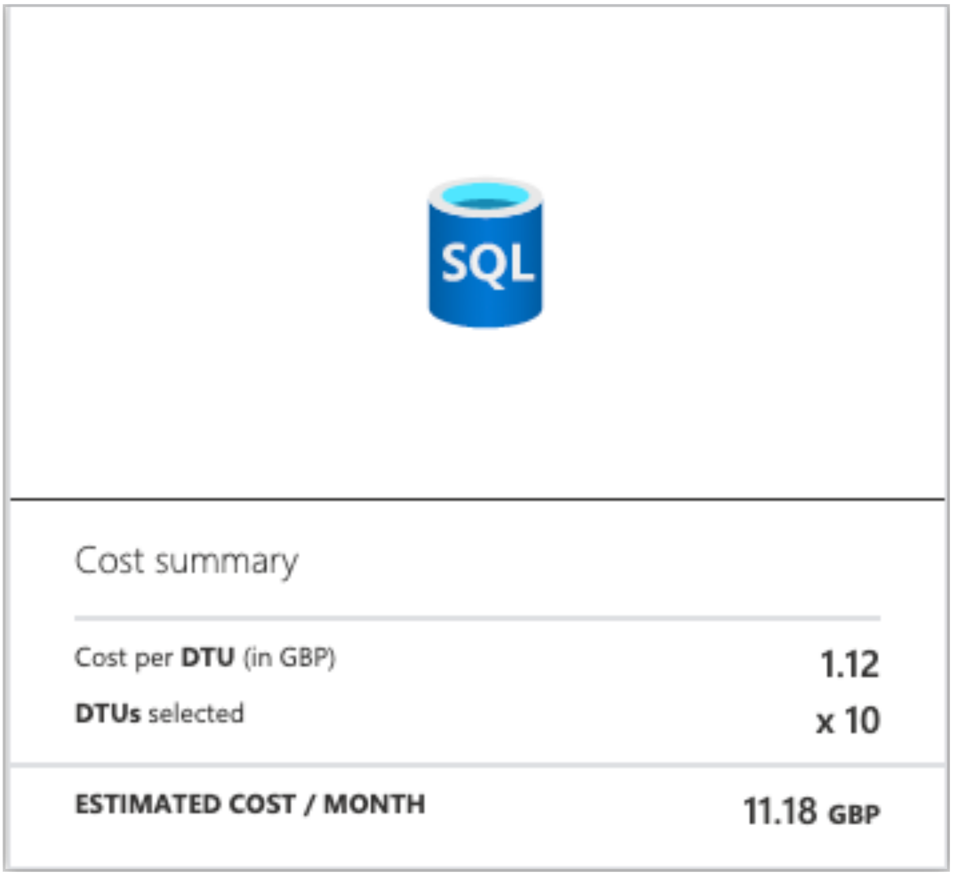
Compute + storage * Standard S0

10 DTUs, 250 GB storage

Configure database

Microsoft have kindly selected Standard S0 as the Database server that doesn't have any pricing info associated with it. I wonder how much this costs….

Click the configure database link to take a look:



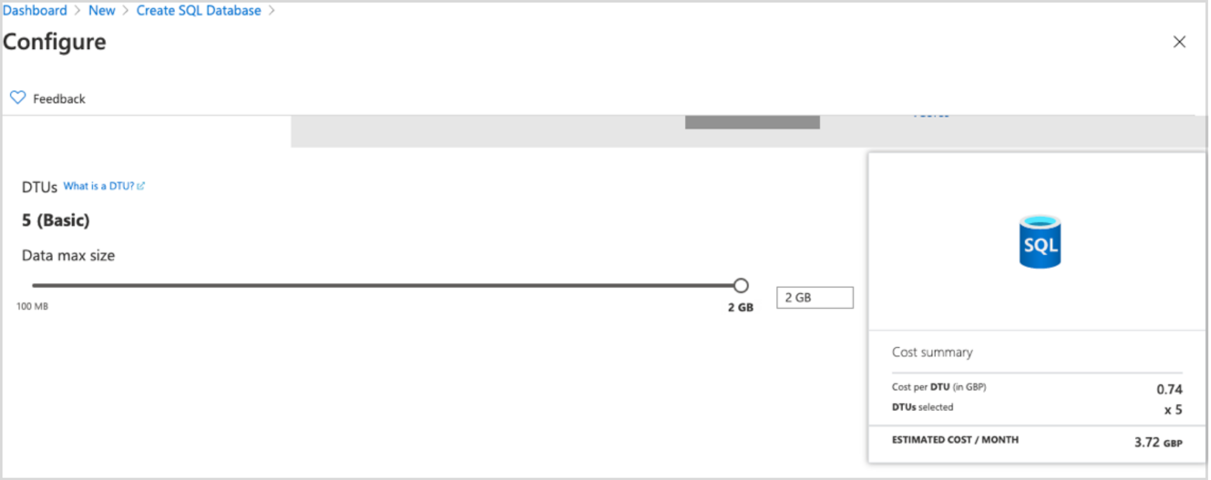
Cost summary

Cost per **DTU** (in GBP) 1.12

DTUs selected x 10

ESTIMATED COST / MONTH 11.18 GBP

Ok not as much as I thought it was going to be but let's see if we can get a freebie.



DTUs What is a DTU?

5 (Basic)

Data max size

Cost summary

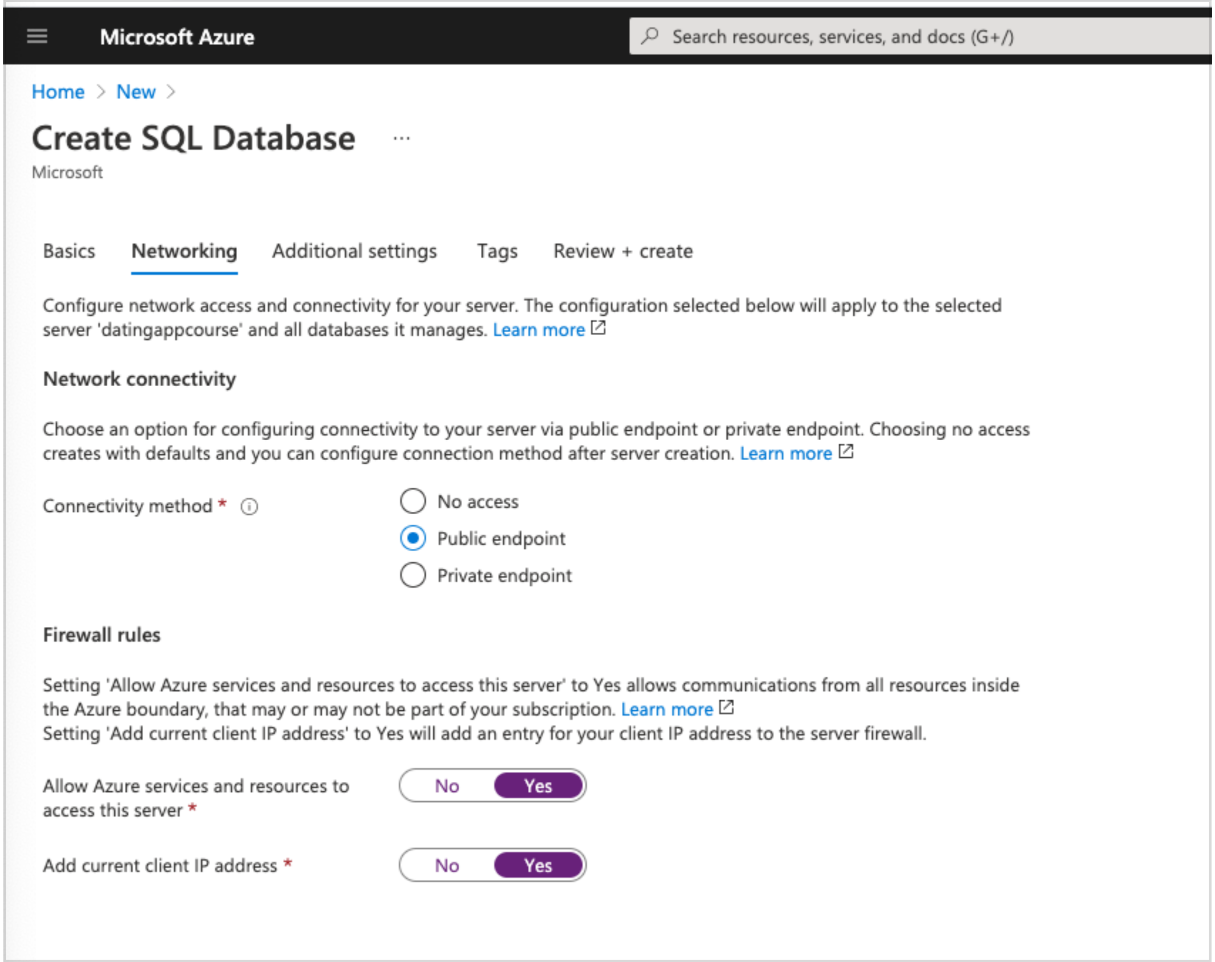
Cost per **DTU** (in GBP) 0.74

DTUs selected x 5

ESTIMATED COST / MONTH 3.72 GBP

This is the cheapest option I can see on mine. No freebies :(Looks like I'll be eating into that $150 they have provided for my new account.

Click next ➜ Networking and set the following options:



Create SQL Database ...

Microsoft

Configure network access and connectivity for your server. The configuration selected below will apply to the selected server 'datingappcourse' and all databases it manages. Learn more

Network connectivity

Choose an option for configuring connectivity to your server via public endpoint or private endpoint. Choosing no access creates with defaults and you can configure connection method after server creation. Learn more

Connectivity method *

○ No access
● Public endpoint
○ Private endpoint

Firewall rules

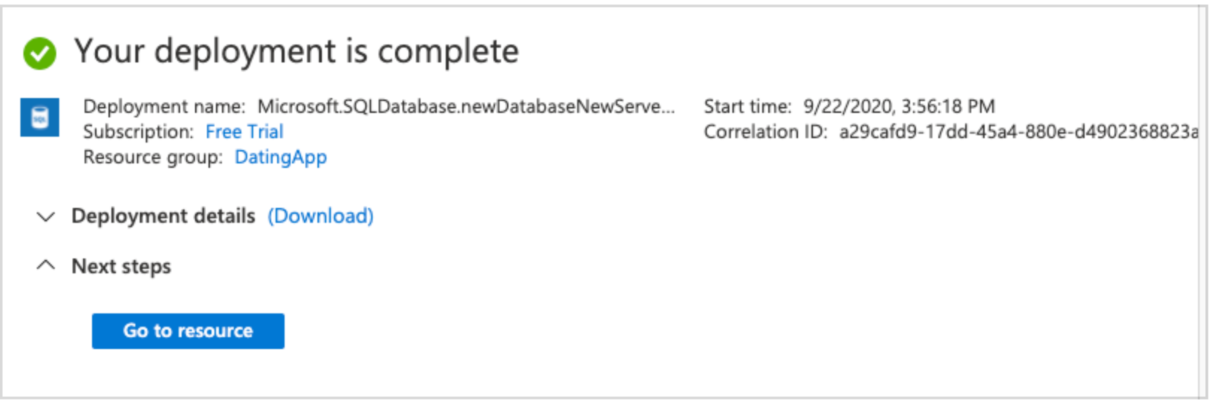
Setting 'Allow Azure services and resources to access this server' to Yes allows communications from all resources inside the Azure boundary, that may or may not be part of your subscription. Learn more
Setting 'Add current client IP address' to Yes will add an entry for your client IP address to the server firewall.

Allow Azure services and resources to access this server * No [✓ Yes]

Add current client IP address * No [✓ Yes]

You can take a look at the other options here but I'm going with this and will select create. Azure will now do its thing so time for another coffee.

Once this has complete you should see the following:

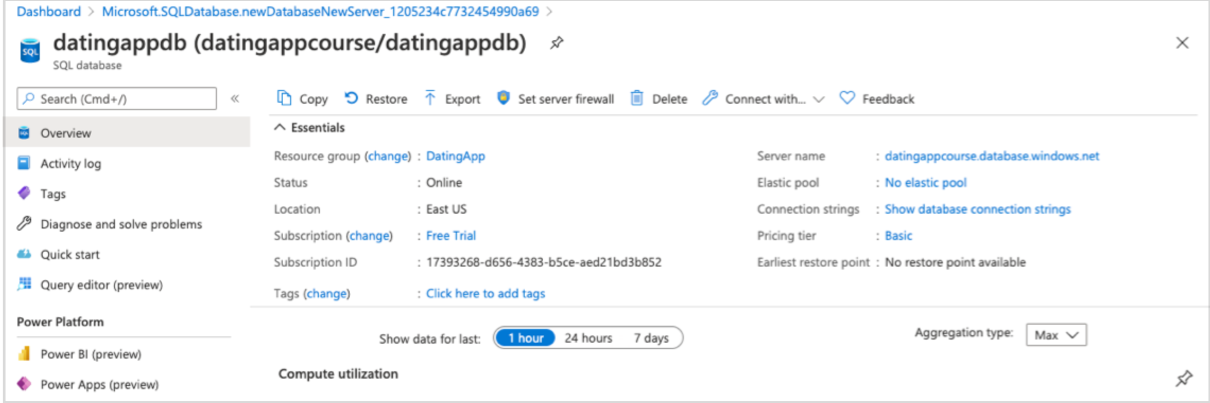


Your deployment is complete

Deployment name: Microsoft.SQLDatabase.newDatabaseNewServe...
Subscription: Free Trial
Resource group: DatingApp

Start time: 9/22/2020, 3:56:18 PM
Correlation ID: a29cafd9-17dd-45a4-880e-d4902368823a

We now need to tell our Web app about our new Database server, but we will need to get the connection string info so click the 'Go to resource' button and you should see the following:



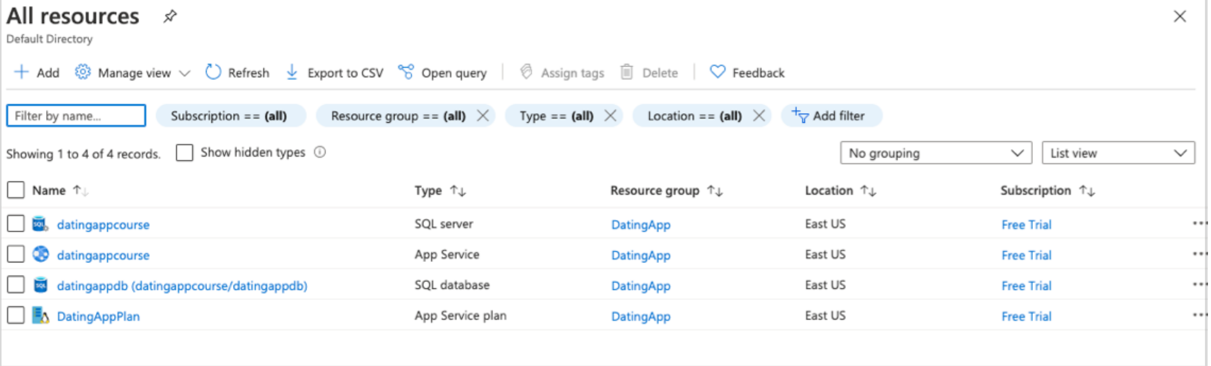
Resource group (change) : DatingApp
Status : Online
Location : East US
Subscription (change) : Free Trial
Subscription ID : 17393268-d656-4383-b5ce-aed21bd3b852
Tags (change) : Click here to add tags
Server name : datingappcourse.database.windows.net
Elastic pool : No elastic pool
Connection strings : Show database connection strings
Pricing tier : Basic
Earliest restore point : No restore point available

Click on the Show database connection strings and copy/paste them somewhere.

```
Server=tcp:datingappcourse.database.windows.net,1433;Initial
Catalog=datingappdb;Persist Security Info=False;User
ID=appuser;Password={your_password};MultipleActiveResultSets=False;Encrypt=True
;TrustServerCertificate=False;Connection Timeout=30;
```

Make a note to yourself you need to update the password as I am pretty sure I didn't set the password to {your_password}

Click on the hamburger icon and select 'All resources' to get to the list of resources we have created. Should see something like this:



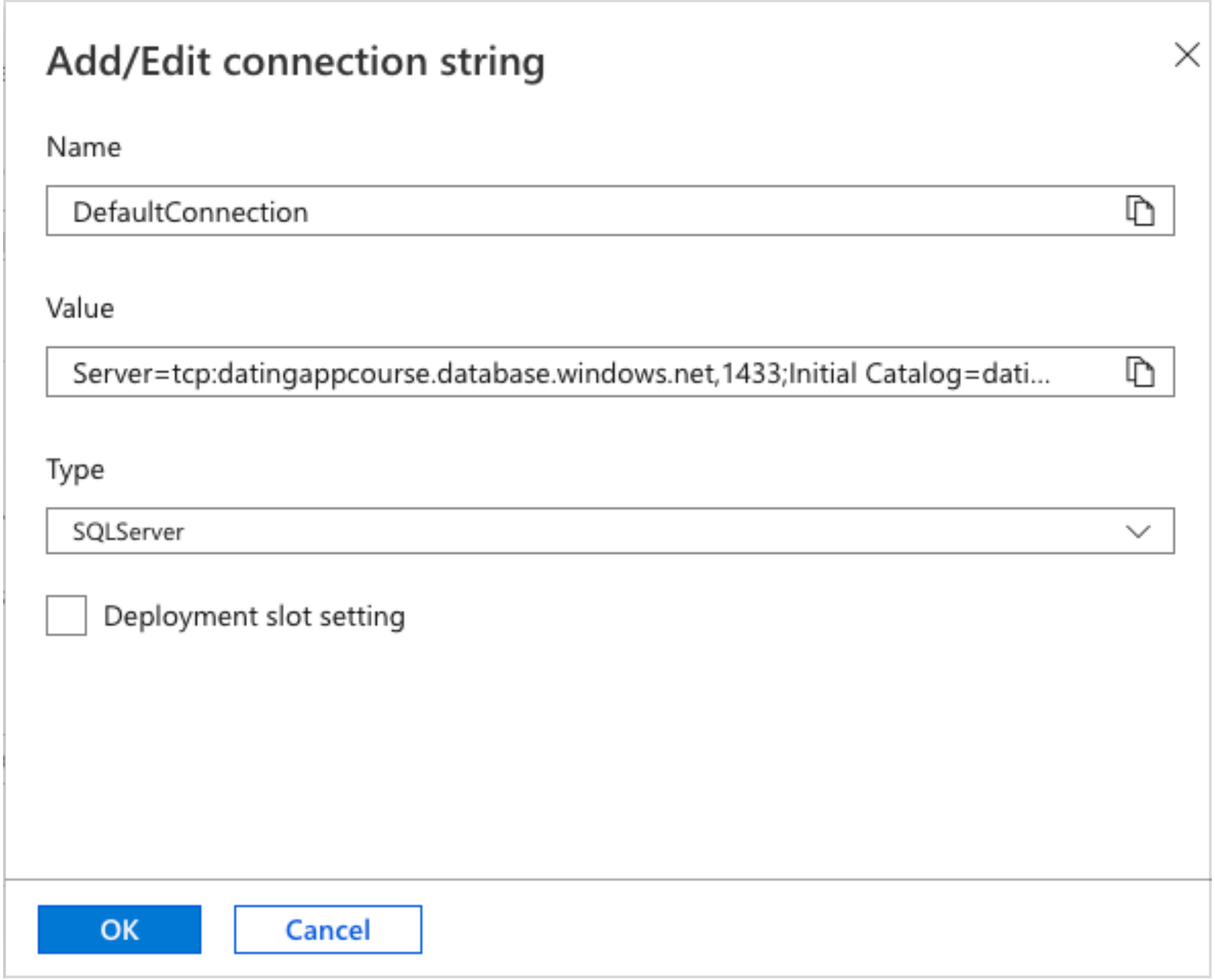
Showing 1 to 4 of 4 records.

Name	Type	Resource group	Location	Subscription
datingappcourse	SQL server	DatingApp	East US	Free Trial
datingappcourse	App Service	DatingApp	East US	Free Trial
datingappdb (datingappcourse/datingappdb)	SQL database	DatingApp	East US	Free Trial
DatingAppPlan	App Service plan	DatingApp	East US	Free Trial

We need to provide environment variables to our app so it knows the following:

1. TokenKey
2. CloudinarySettings
3. ConnectionString

Click on the App Service, then once here select ‘Configuration’ from the list on the left. Click on the button to add a new Connection String:



Add/Edit connection string

Name: DefaultConnection

Value: Server=tcp:datingappcourse.database.windows.net,1433;Initial Catalog=dati...

Type: SQLServer

□ Deployment slot setting

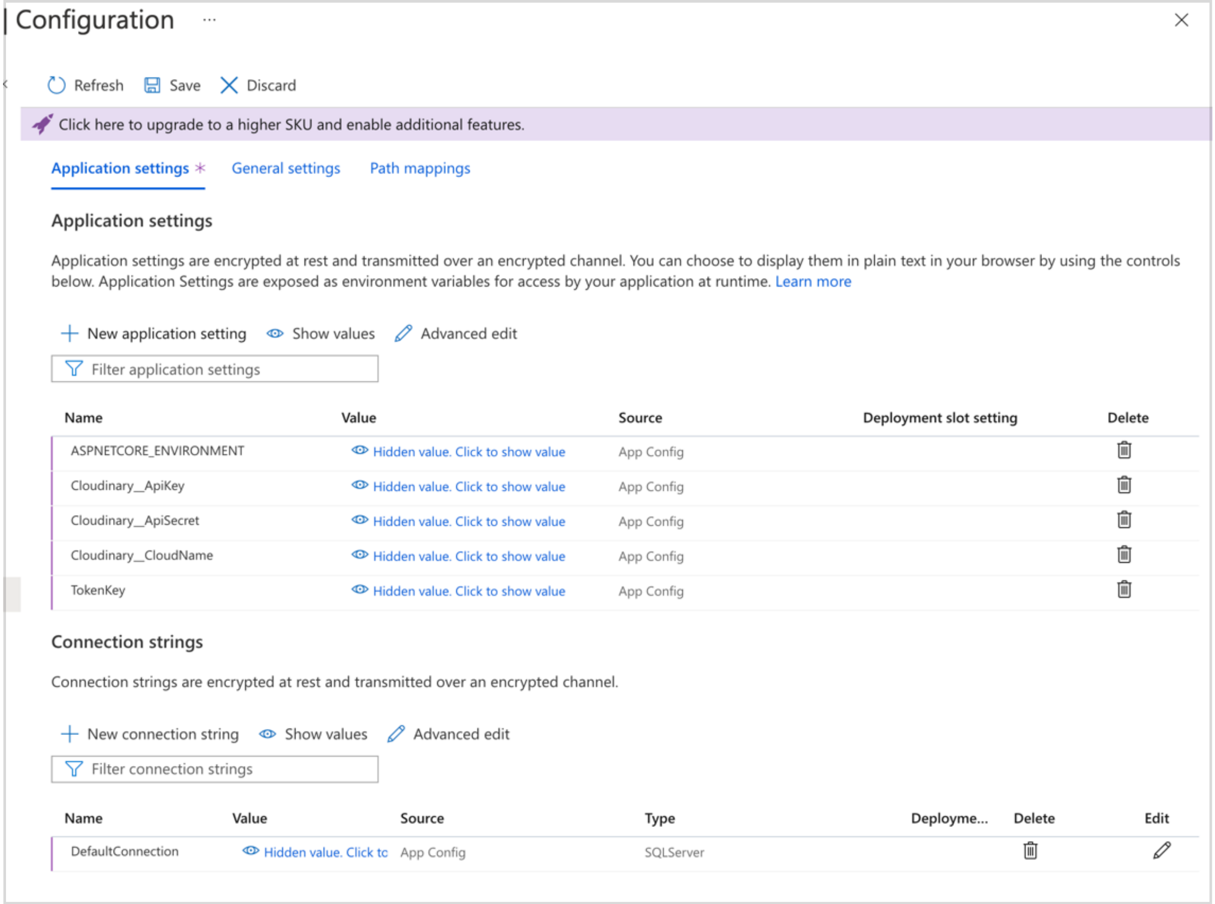
Make sure you call it “DefaultConnection” as this is what we called this connection in our app.

Don’t forget to change {your_password} to your password. Not literally ‘your password’ but the password you created earlier!

We can then add the application settings for the TokenKey and cloudinary.   Make sure your token key is a strong key so use a password generator and go for at least 32 characters.

Also, add a key for the environment as well so set ASPNETCORE_ENVIRONMENT to Production

Should see something like this once you are done:


Configuration
Refresh
Save
Discard
Click here to upgrade to a higher SKU and enable additional features.
Application settings *
General settings
Path mappings

Application settings

Application settings are encrypted at rest and transmitted over an encrypted channel. You can choose to display them in plain text in your browser by using the controls below. Application Settings are exposed as environment variables for access by your application at runtime. Learn more

+ New application setting
Show values
Advanced edit
Filter application settings

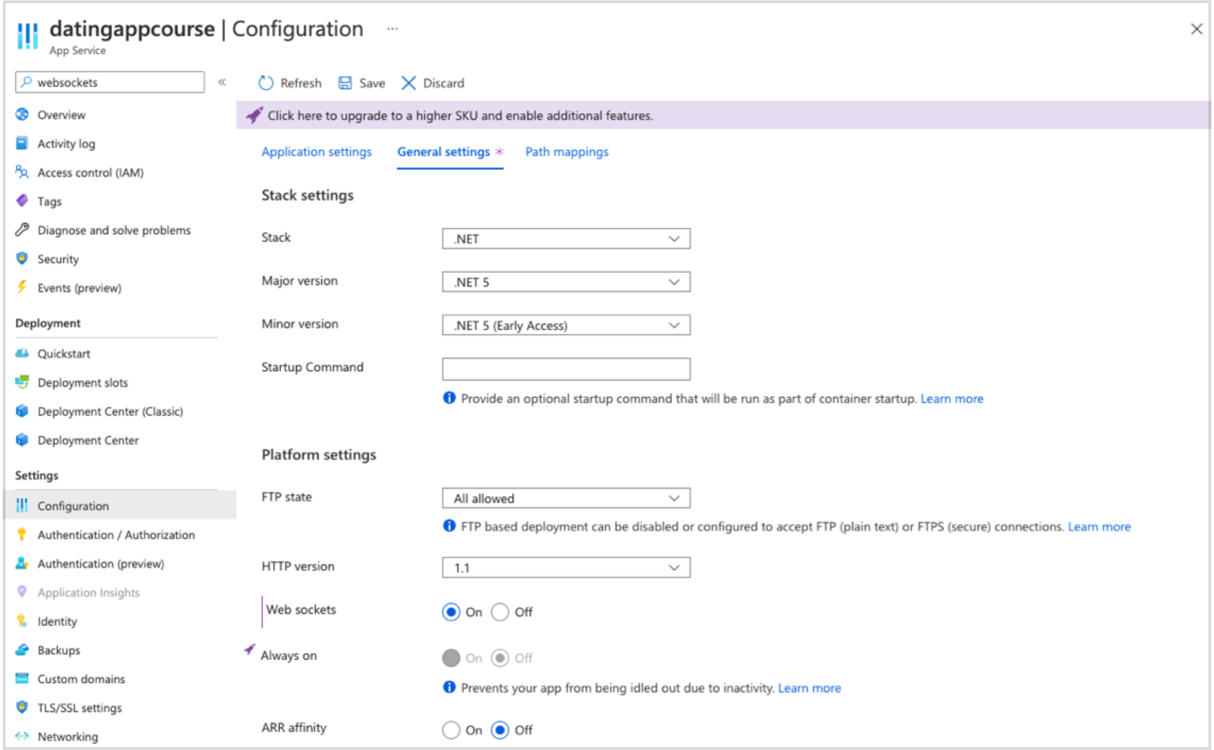
Name	Value	Source	Deployment slot setting	Delete
ASPNETCORE_ENVIRONMENT	Hidden value. Click to show value	App Config		🗑
Cloudinary__ApiKey	Hidden value. Click to show value	App Config		🗑
Cloudinary__ApiSecret	Hidden value. Click to show value	App Config		🗑
Cloudinary__CloudName	Hidden value. Click to show value	App Config		🗑
TokenKey	Hidden value. Click to show value	App Config		🗑

Connection strings

Connection strings are encrypted at rest and transmitted over an encrypted channel.

Name	Value	Source	Type	Deployme...	Delete	Edit
DefaultConnection	Hidden value. Click tc	App Config	SQLServer		🗑	✎

We also need to enable WebSockets so on the General Settings tab enable this:



Stack settings

Stack: .NET

Major version: .NET 5

Minor version: .NET 5 (Early Access)

Startup Command:

Provide an optional startup command that will be run as part of container startup. Learn more

Platform settings

FTP state: All allowed

FTP based deployment can be disabled or configured to accept FTP (plain text) or FTPS (secure) connections. Learn more

HTTP version: 1.1

Web sockets: ● On ○ Off

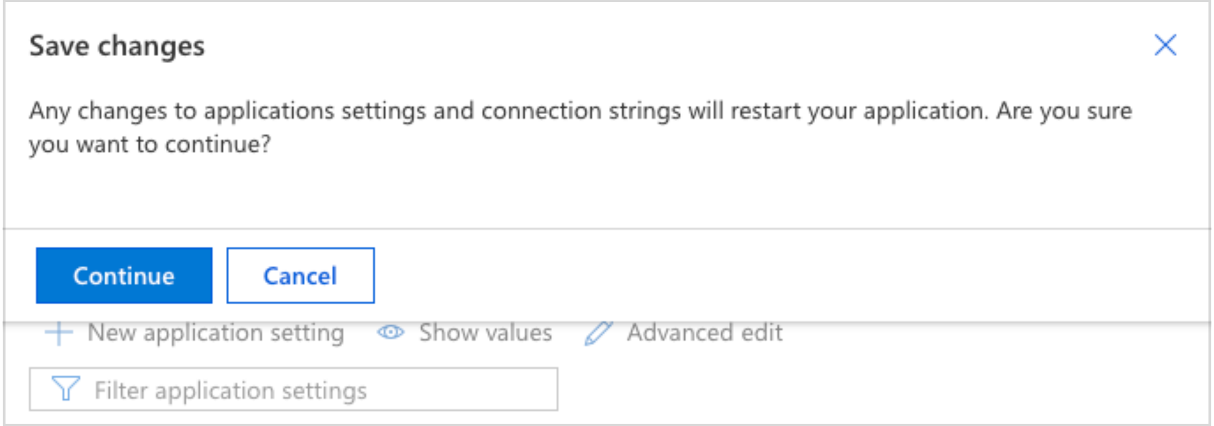
Always on: ● On ○ Off

Prevents your app from being idled out due to inactivity. Learn more

ARR affinity: ○ On ● Off

Note that we cannot use a : for nested keys as we chose a linux container earlier. When specifying a nested key we have to use a double underscore

Also, don't forget to hit the Save changes button when you are done and continue which will restart our app.



Save changes

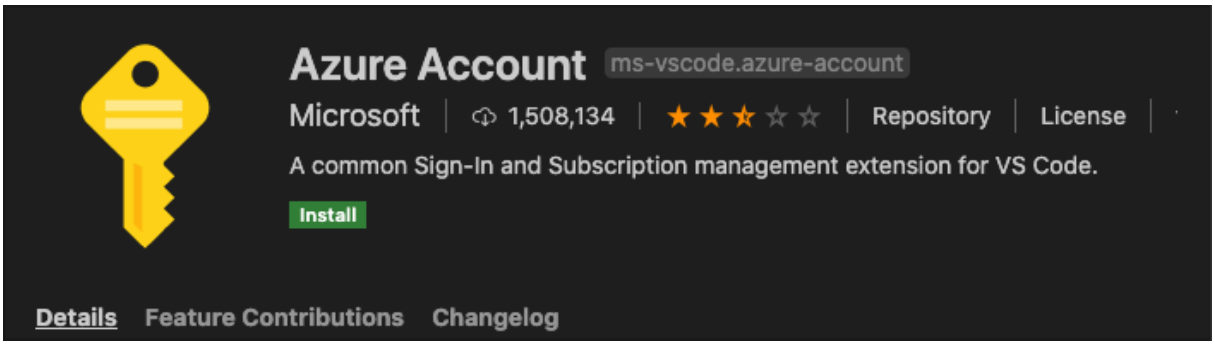
Any changes to applications settings and connection strings will restart your application. Are you sure you want to continue?

Deploying the app.

Our next task is to get our app from our Dev machine over to Azure.

To do this we will need to add 2 new extensions to VS Code to make our life a bit easier.

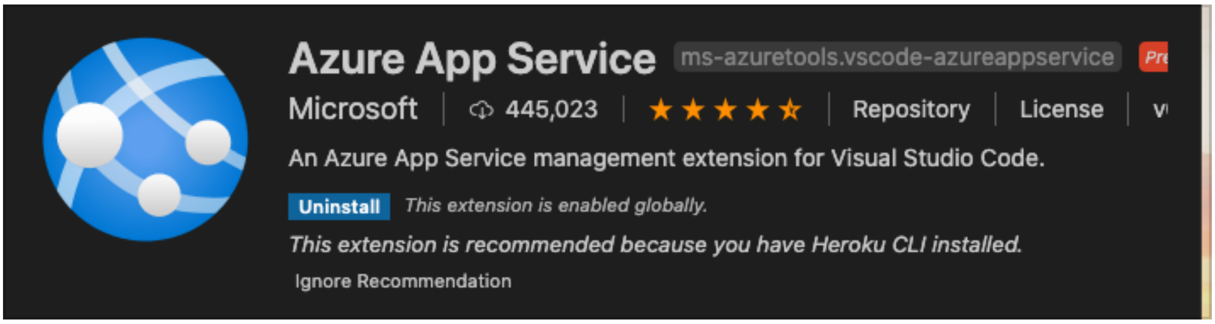
Azure Account extension and Azure App Service.



Azure Account ms-vscode.azure-account

Microsoft | 1,508,134 | ★★★☆☆ | Repository | License |

A common Sign-In and Subscription management extension for VS Code.



Azure App Service ms-azuretools.vscode-azureappservice

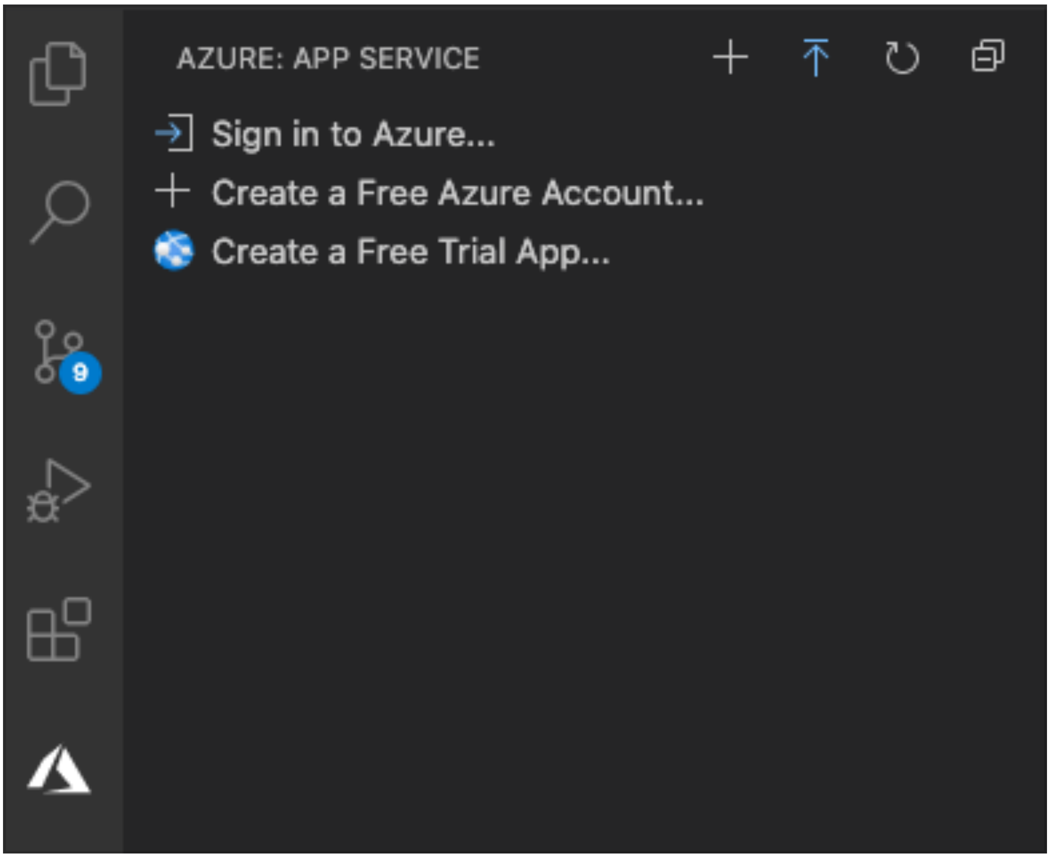
Microsoft | 445,023 | ★★★★★ | Repository | License | v

An Azure App Service management extension for Visual Studio Code.

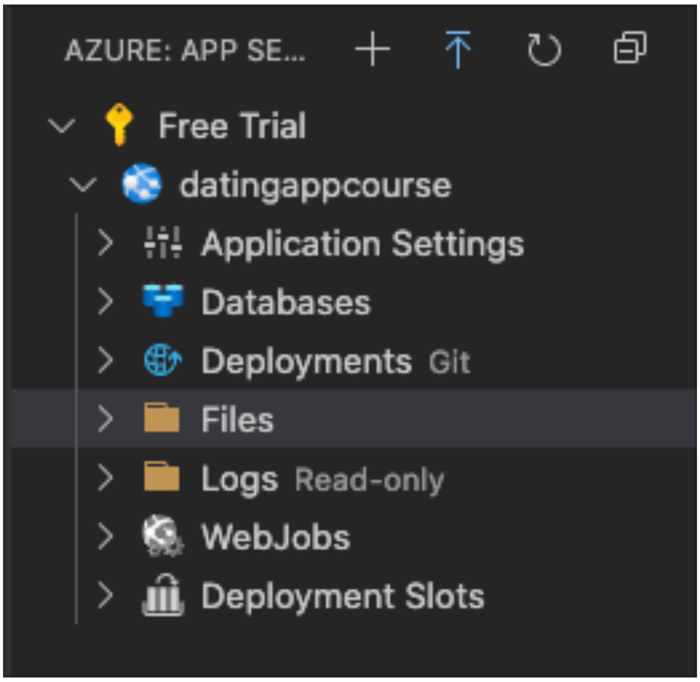
This extension is enabled globally.

This extension is recommended because you have Heroku CLI installed.

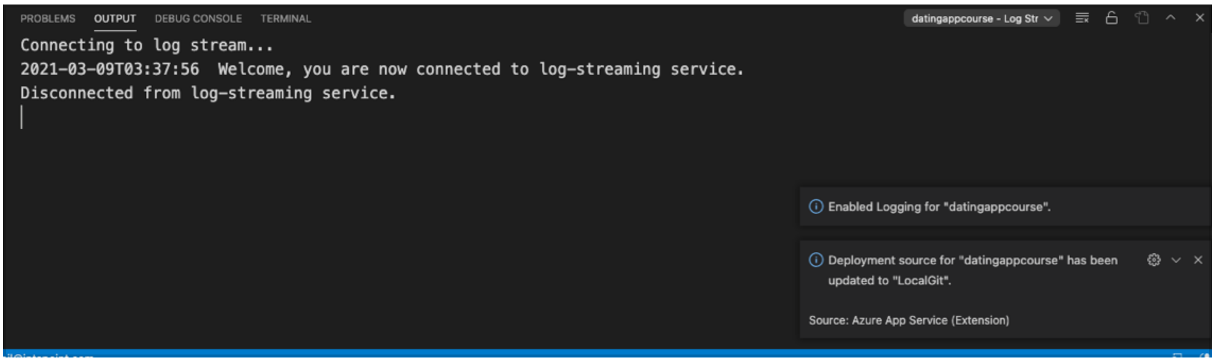
Click on the shiny new Azure Icon in VS Code and sign in with your account:



Should see the following:



Right click the app and select “start streaming logs” so that we get logging information from Azure.



```
Connecting to log stream...
2021-03-09T03:37:56  Welcome, you are now connected to log-streaming service.
Disconnected from log-streaming service.
```

Enabled Logging for "datingappcourse".

Deployment source for "datingappcourse" has been updated to "LocalGit".

Source: Azure App Service (Extension)

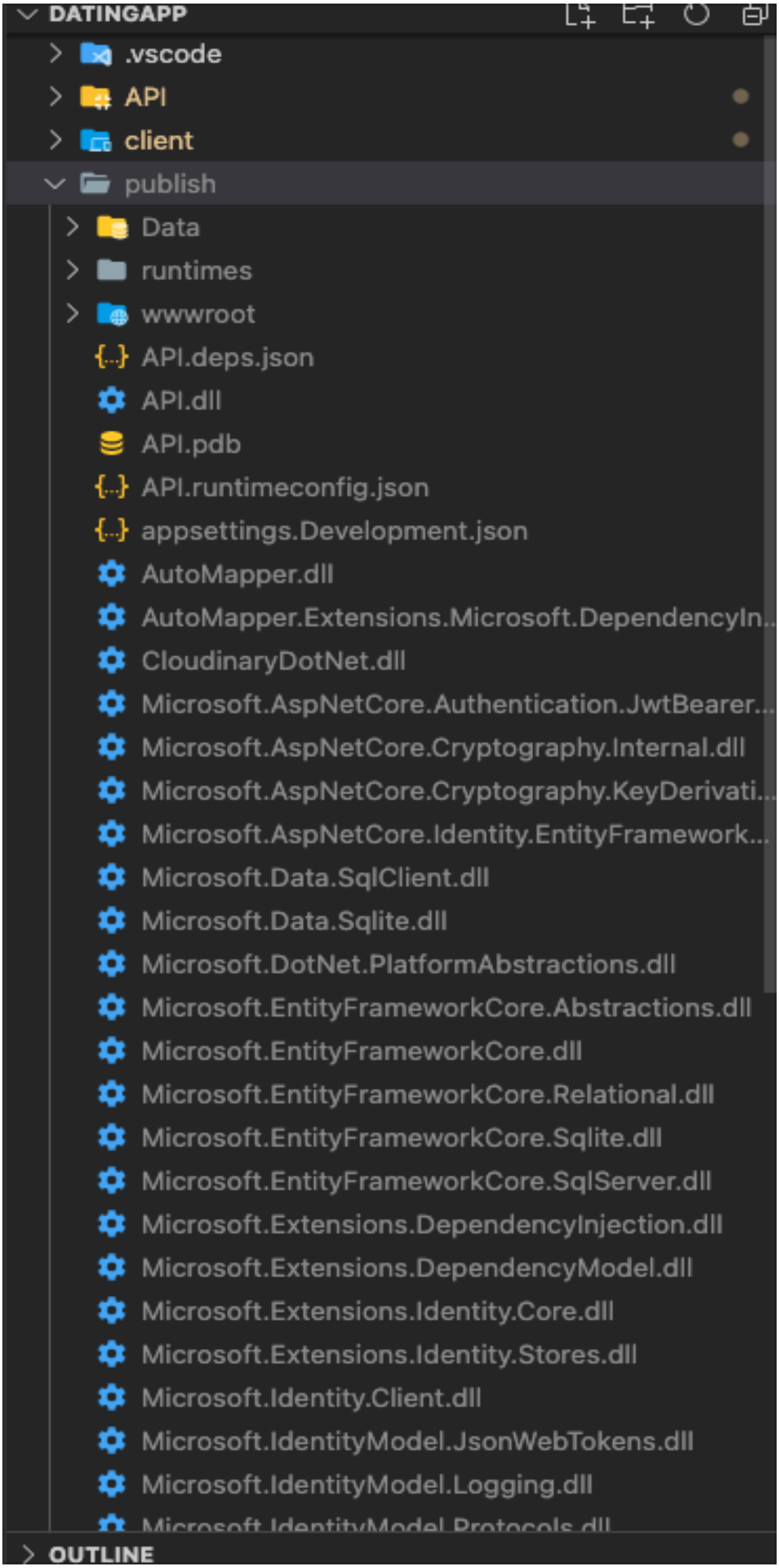
If you need to republish your Angular app for any reason now is a good time to do so.

```
ng build --prod
```

Now we will create a publish folder that we will use for deployment for the .Net application.   In the solution folder run the following:

```
dotnet publish -c Release -o publish
```

In your solution folder you should see:

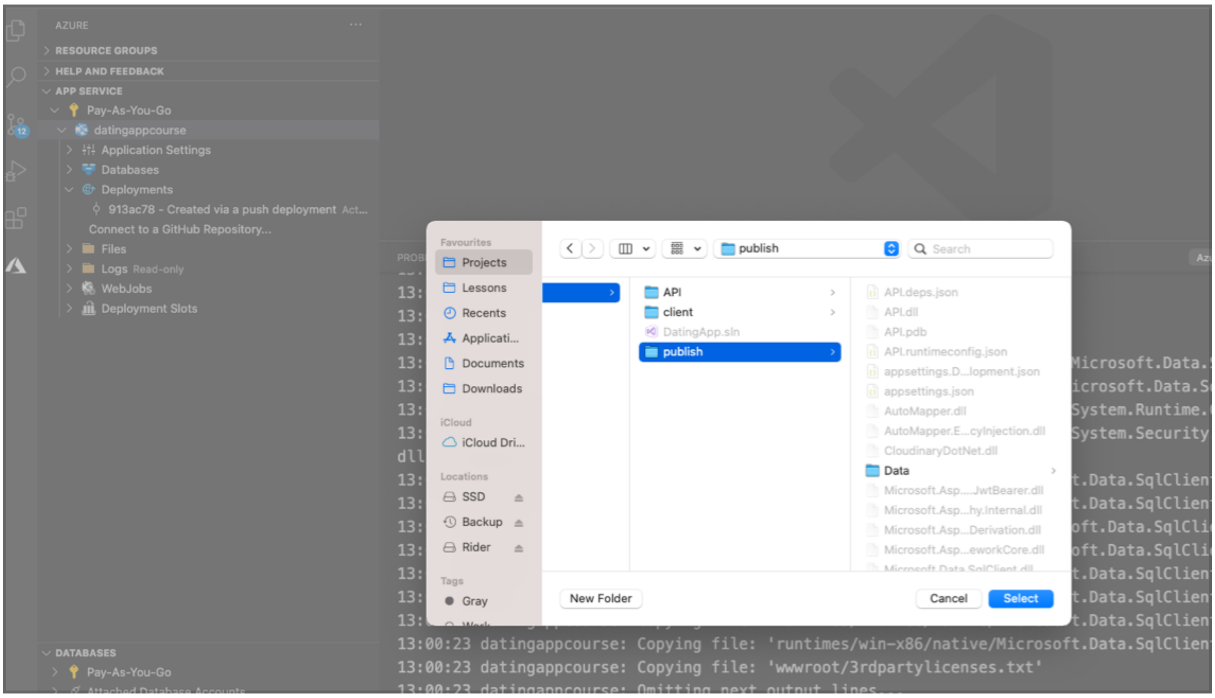


```
DATINGAPP
  > .vscode
  > API
  > client
  v publish
    > Data
    > runtimes
    > wwwroot
      API.deps.json
      API.dll
      API.pdb
      API.runtimeconfig.json
      appsettings.Development.json
      AutoMapper.dll
      AutoMapper.Extensions.Microsoft.DependencyIn...
      CloudinaryDotNet.dll
      Microsoft.AspNetCore.Authentication.JwtBearer...
      Microsoft.AspNetCore.Cryptography.Internal.dll
      Microsoft.AspNetCore.Cryptography.KeyDerivati...
      Microsoft.AspNetCore.Identity.EntityFramework...
      Microsoft.Data.SqlClient.dll
      Microsoft.Data.Sqlite.dll
      Microsoft.DotNet.PlatformAbstractions.dll
      Microsoft.EntityFrameworkCore.Abstractions.dll
      Microsoft.EntityFrameworkCore.dll
      Microsoft.EntityFrameworkCore.Relational.dll
      Microsoft.EntityFrameworkCore.Sqlite.dll
      Microsoft.EntityFrameworkCore.SqlServer.dll
      Microsoft.Extensions.DependencyInjection.dll
      Microsoft.Extensions.DependencyModel.dll
      Microsoft.Extensions.Identity.Core.dll
      Microsoft.Extensions.Identity.Stores.dll
      Microsoft.Identity.Client.dll
      Microsoft.IdentityModel.JsonWebTokens.dll
      Microsoft.IdentityModel.Logging.dll
      Microsoft.IdentityModel.Protocols.dll
> OUTLINE
```

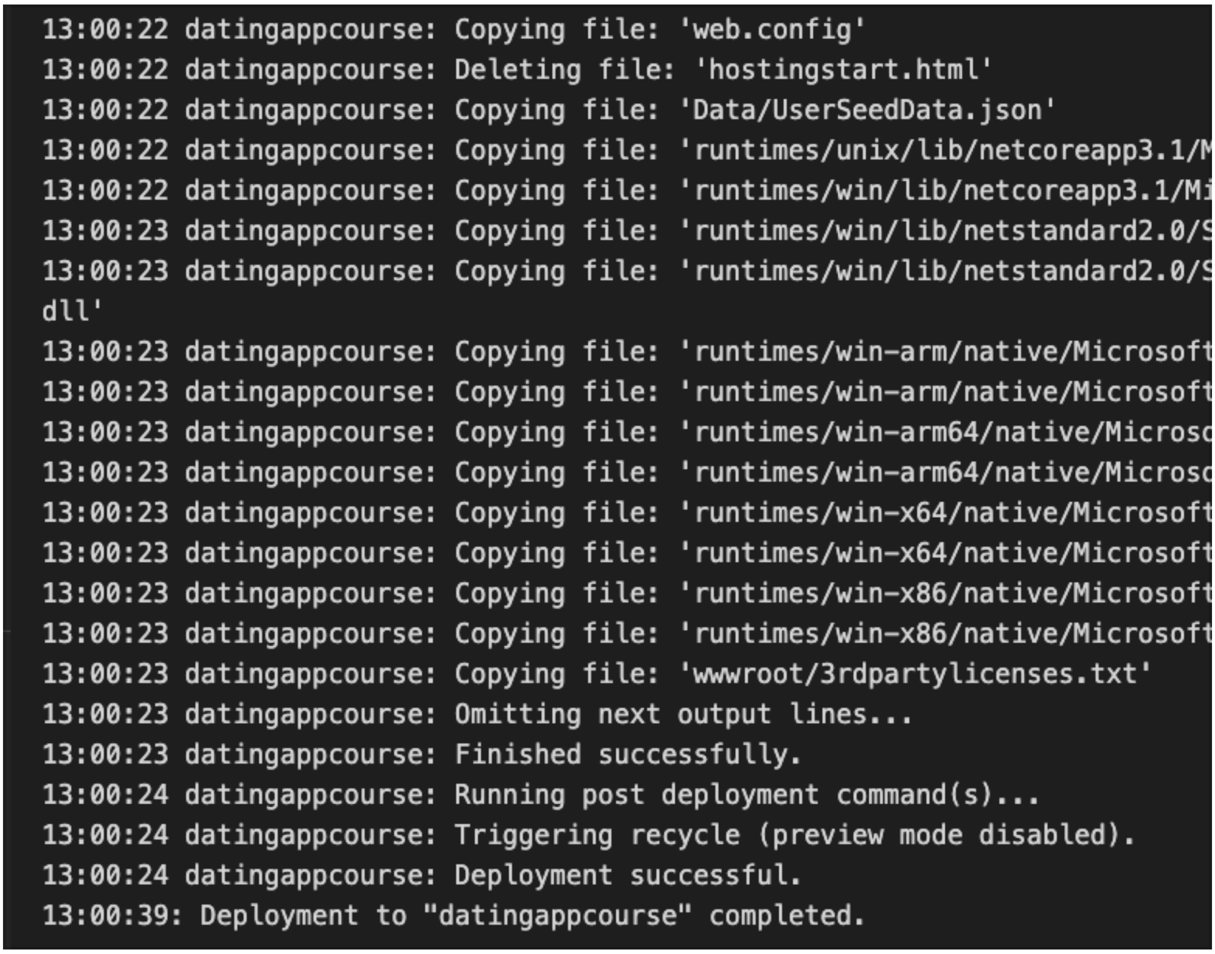
Now we are ready to publish!

Right-click the app service in the Azure extension and select deploy to web app…

Browse to the publish folder and select it.

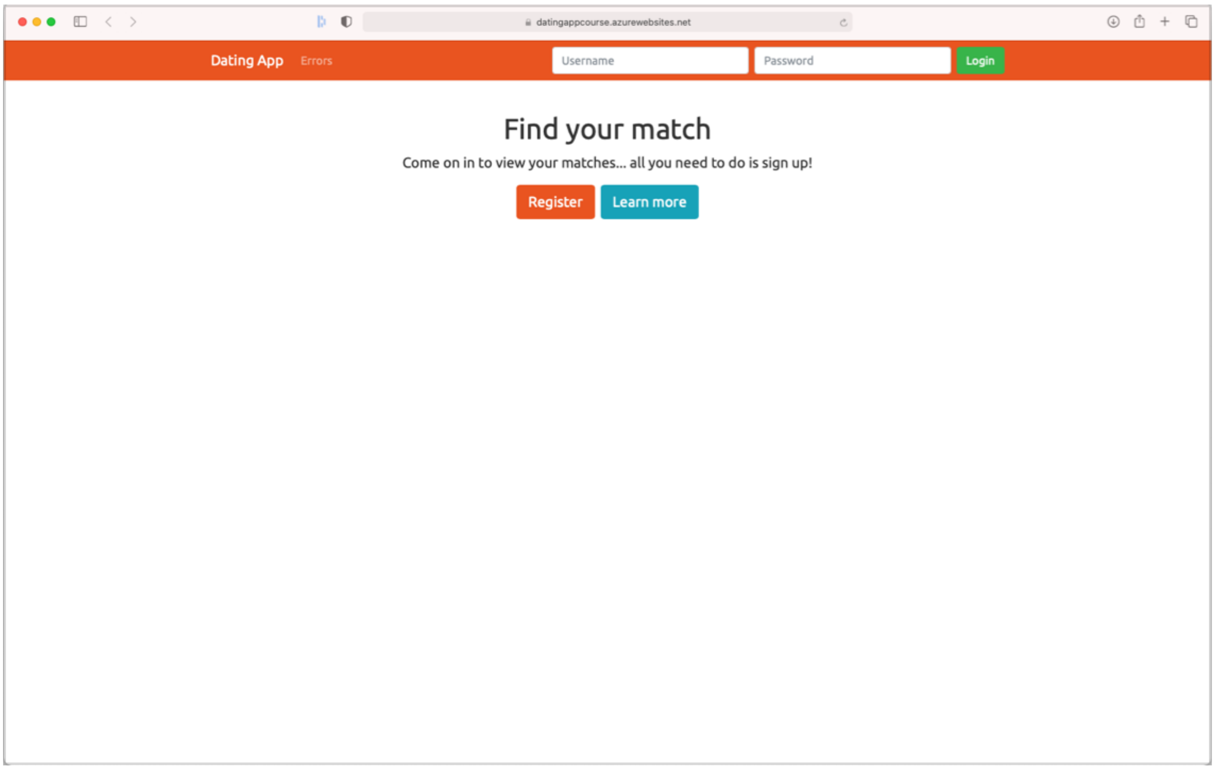


```
13:00:23 datingappcourse: Copying file: 'runtimes/win-x86/native/Microsoft.Data.SqlClien
13:00:23 datingappcourse: Copying file: 'wwwroot/3rdpartylicenses.txt'
13:00:23 datingappcourse: Omitting next output lines...
```



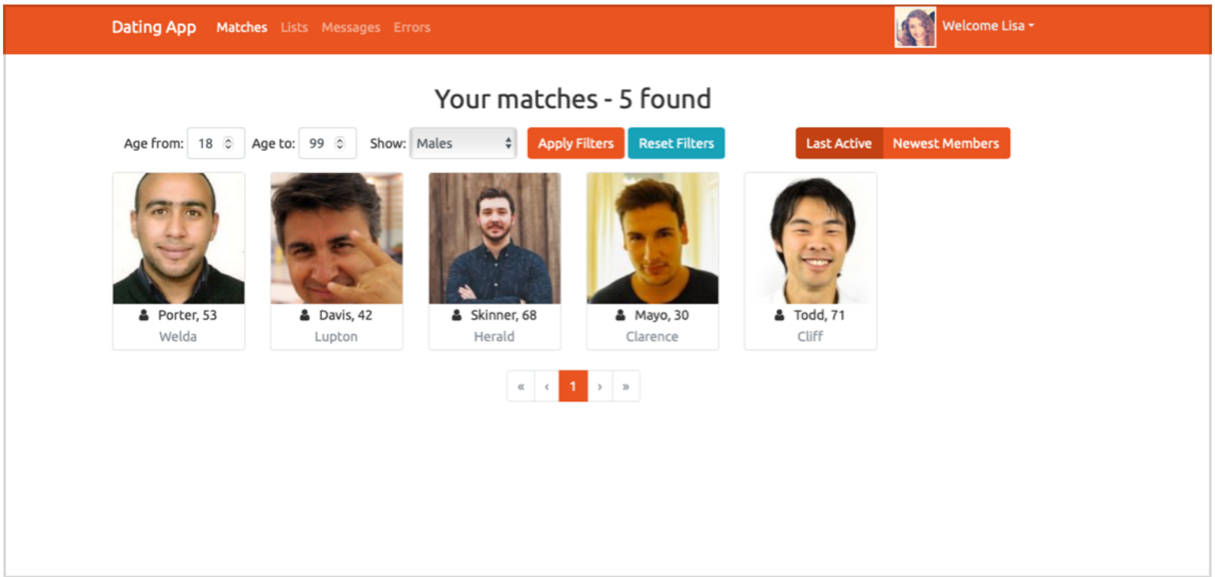
```
13:00:22 datingappcourse: Copying file: 'web.config'
13:00:22 datingappcourse: Deleting file: 'hostingstart.html'
13:00:22 datingappcourse: Copying file: 'Data/UserSeedData.json'
13:00:22 datingappcourse: Copying file: 'runtimes/unix/lib/netcoreapp3.1/M
13:00:22 datingappcourse: Copying file: 'runtimes/win/lib/netcoreapp3.1/Mi
13:00:23 datingappcourse: Copying file: 'runtimes/win/lib/netstandard2.0/S
13:00:23 datingappcourse: Copying file: 'runtimes/win/lib/netstandard2.0/S
dll'
13:00:23 datingappcourse: Copying file: 'runtimes/win-arm/native/Microsoft
13:00:23 datingappcourse: Copying file: 'runtimes/win-arm/native/Microsoft
13:00:23 datingappcourse: Copying file: 'runtimes/win-arm64/native/Microso
13:00:23 datingappcourse: Copying file: 'runtimes/win-arm64/native/Microso
13:00:23 datingappcourse: Copying file: 'runtimes/win-x64/native/Microsoft
13:00:23 datingappcourse: Copying file: 'runtimes/win-x64/native/Microsoft
13:00:23 datingappcourse: Copying file: 'runtimes/win-x86/native/Microsoft
13:00:23 datingappcourse: Copying file: 'runtimes/win-x86/native/Microsoft
13:00:23 datingappcourse: Copying file: 'wwwroot/3rdpartylicenses.txt'
13:00:23 datingappcourse: Omitting next output lines...
13:00:23 datingappcourse: Finished successfully.
13:00:24 datingappcourse: Running post deployment command(s)...
13:00:24 datingappcourse: Triggering recycle (preview mode disabled).
13:00:24 datingappcourse: Deployment successful.
13:00:39: Deployment to "datingappcourse" completed.
```

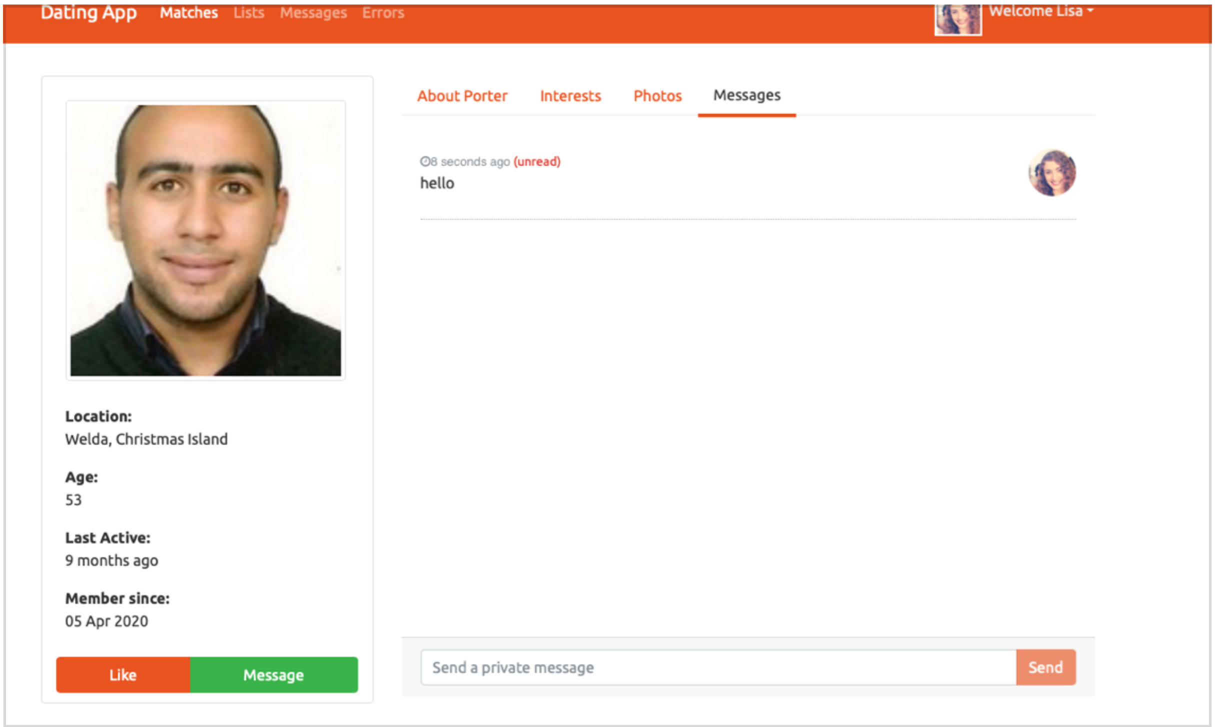
It will pop up with an option to browse the site and you can say yes and hey presto! The site is live and published in Azure.



Find your match

Come on in to view your matches... all you need to do is sign up!





Location:
Welda, Christmas Island

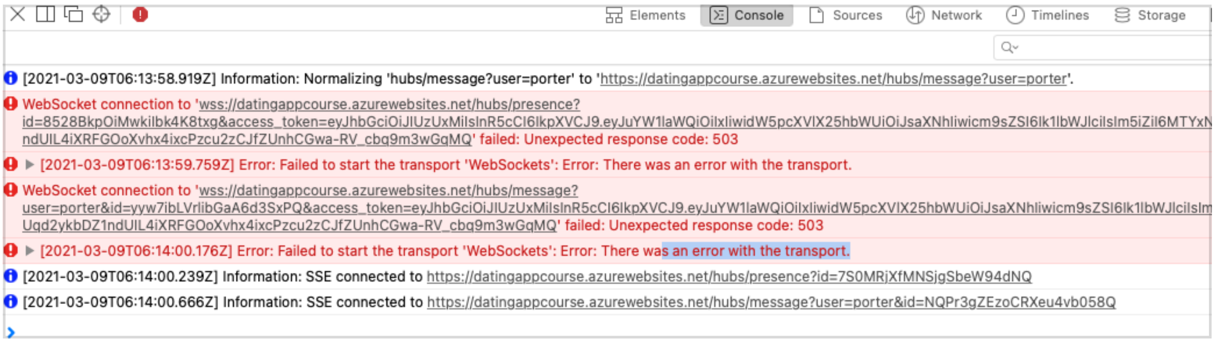
Age:
53

Last Active:
9 months ago

Member since:
05 Apr 2020

8 seconds ago (unread)
hello

Note: At time of writing there is an issue with Azure web sockets so it is defaulting to Server Side Events. You may see errors in the logs like follows:



```
[2021-03-09T06:13:58.919Z] Information: Normalizing 'hubs/message?user=porter' to 'https://datingappcourse.azurewebsites.net/hubs/message?user=porter'.
WebSocket connection to 'wss://datingappcourse.azurewebsites.net/hubs/presence?
id=8528BkpOiMwkilbk4K8txg&access_token=eyJhbGciOiJIUzUxMiIsInR5cCI6IkpXVCJ9.eyJuYW1laWQiOiIxIiwidW5pcXVlX25hbWUiOiJsaXNhIiwicm9sZSI6Ik1lbWJlciIsIm5iZiI6MTYxN
ndUIL4iXRFGOoXvhx4ixcPzcu2zCJfZUnhCGwa-RV_cbq9m3wGqMQ' failed: Unexpected response code: 503
▶ [2021-03-09T06:13:59.759Z] Error: Failed to start the transport 'WebSockets': Error: There was an error with the transport.
WebSocket connection to 'wss://datingappcourse.azurewebsites.net/hubs/message?
user=porter&id=yyw7ibLVrlibGaA6d3SxPQ&access_token=eyJhbGciOiJIUzUxMiIsInR5cCI6IkpXVCJ9.eyJuYW1laWQiOiIxIiwidW5pcXVlX25hbWUiOiJsaXNhIiwicm9sZSI6Ik1lbWJlciIsIm
Uqd2ykbDZ1ndUIL4iXRFGOoXvhx4ixcPzcu2zCJfZUnhCGwa-RV_cbq9m3wGqMQ' failed: Unexpected response code: 503
▶ [2021-03-09T06:14:00.176Z] Error: Failed to start the transport 'WebSockets': Error: There was an error with the transport.
[2021-03-09T06:14:00.239Z] Information: SSE connected to https://datingappcourse.azurewebsites.net/hubs/presence?id=7S0MRjXfMNSjgSbeW94dNQ
[2021-03-09T06:14:00.666Z] Information: SSE connected to https://datingappcourse.azurewebsites.net/hubs/message?user=porter&id=NQPr3gZEzoCRXeu4vb058Q
```

This just means that Websockets is not available. The chat/presence still works but just using a different protocol. The fact that .Net is still in “Early access” and not released probably has something to do with this.

#datingapp-final/section19