

# Deployment Notes

---

Documentation on deploying Whatfits media application

Authors: Roberto Sanchez (014587792), Aaron Pham (011033891)

Date: April 29, 2018

Revision: 2

# Table of Contents

---

<b>Table of Contents</b>	<b>2</b>
<b>Part 0: Preparing the code</b>	<b>3</b>
List of modifications	3
The modifications:	3
<b>Part 1: Compiling the files</b>	<b>8</b>
Compiling the client side	8
Publishing the WebAPI portion	9
<b>Part 2: Uploading the folder to the server</b>	<b>16</b>
<b>Part 3: Setting up the Database</b>	<b>17</b>
<b>Part 4: Setting up files / folders in Server</b>	<b>20</b>
Moving the files	20
<b>Part 5: Deploying the web application</b>	<b>22</b>
Installing Certificate on IIS	22
Client-side Deployment	24
Backend Deployment	27
Setting up Images Folder	30
<b>Part 6: Configuring IIS</b>	<b>34</b>
Enabling SSL on Client	34
Doing URL Rewrite on Client	35
Create Rule	35
Add Bindings	38

# Part 0: Preparing the code

---

## List of modifications

The following files need to be modified (from master branch):

- On Client
  - Chat.vue (src/components/UserProfile/)
  - Follows.vue (src/components/UserProfile/)
  - User.js (src/store/modules/)
  - Request.js (src/store/modules/)
  - Index.js (src/router/)
  - MainNavBar.vue (src/components/NavBar/)
- On Server
  - Web.config (Whatfits/server/server/)

## The modifications:

- **Chat.vue**

- Line 59:

- BEFORE:

```
this.ws = new WebSocket('ws://localhost/server/v1/chat/connect' +  
'?username=' + this.onlineUser)
```

- AFTER:

```
this.ws = new WebSocket('wss://www.whatfits.social/server/v1/chat/connect'  
+ '?username=' + this.onlineUser)
```

- **Follows.vue**

**Line 35:**

BEFORE:

```
url: 'http://localhost/server/v1/follows/Addfollows',
```

AFTER:

```
url: 'https://www.whatfits.social/server/v1/follows/Addfollows',
```

**Line 64:**

BEFORE:

```
url: 'http://localhost/server/v1/follows/Deletefollows',
```

AFTER:

```
url: 'https://www.whatfits.social/server/v1/follows/Deletefollows',
```

**Line 94:**

BEFORE:

```
url: 'http://localhost/server/v1/follows/Isfollows',
```

AFTER:

```
url: 'https://www.whatfits.social/server/v1/follows/Isfollows',
```

**IN STORE (src/store/modules/)**

- **Request.js**

**Line 3:**

BEFORE:

```
'Access-Control-Allow-Origin': 'http://localhost:8080',
```

AFTER:

```
'Access-Control-Allow-Origin': 'https://www.whatfits.social',
```

**Line 9:**

BEFORE:

```
url: 'http://localhost/server/'
```

AFTER:

```
url: 'https://www.whatfits.social/server/'
```

- **User.js**

**Line 8:**

BEFORE:

```
profilepicture: 'http://localhost/images/profileImages/',
```

AFTER:

```
profilepicture: 'https://www.whatfits.social/images/profileImages/',
```

- Index.js of router folder

**Line 12:**

**Before:**

```
mode: 'history',
```

**After:**

```
// mode: 'history',
```

- **MainNavBar.vue**

**Line 87:**

**Allow to go back to your profile from mainnavbar**

**Before:**

```
this.$router.push('/profile')
```

**After:**

```
this.$store.dispatch('actviewprofile', {ViewProfile:this.$store.getters.getUsername})
```

```
location.reload()
```

```
this.$router.push('/profile')
```

## **IN SERVER**

### **WEB.CONFIG**

**Before**

```
<!-- SQL Connection String for Local development. -Rob -->
  <connectionStrings>
    <add name="WhatfitsDb" providerName="System.Data.SqlClient"
connectionString="data source=.\SQLEXPRESS; initial catalog=WhatfitsDb;
integrated security=SSPI" />
  </connectionStrings>
  <!-- SQL Connection String for Azure Database. -Rob -->
  <!--
  <connectionStrings>
    <add name="WhatfitsDb" providerName="System.Data.SqlClient"
connectionString="Data
Source=tcp:webapp491a.database.windows.net,1433;Initial
Catalog=WhatfitsDb;Persist Security Info=False;User
```

```
ID=rsanchez;Password=ajfLV5nMkLAZ6Dpj;MultipleActiveResultSets=False;Encrypt=True;TrustServerCertificate=False;Connection Timeout=30;"/>
    </connectionStrings>
    -->
</configuration>
```

After (Previous connection string is removed and this is uncommented)

```
<!-- SQL Connection String for Azure Database. -Rob -->
<connectionStrings>
    <add name="WhatfitsDb" providerName="System.Data.SqlClient"
connectionString="Data
Source=tcp:webapp491a.database.windows.net,1433;Initial
Catalog=WhatfitsDb;Persist Security Info=False;User
ID=rsanchez;Password=ajfLV5nMkLAZ6Dpj;MultipleActiveResultSets=False;Encrypt=True;TrustServerCertificate=False;Connection Timeout=30;"/>
    </connectionStrings>
</configuration>
```

## END OF EDITING CODE

# Part 1: Compiling the files

---

## Compiling the client side

This part is for publishing the client side of the project. Open the client side on Visual Studios Code -> **open Integrated Terminal**

### Steps:

1. Open **Visual Studio Code**
2. Open the project **client**
3. **Open Integrated Terminal on View -> Integrated Terminal**
4. Run the following command in the terminal
  - a. ~~npm install~~ (This step will be done before hand)
  - b. **npm run build**

```
rsanc@Sapphire-WinLaptop MINGW64 ~/Documents/Github/WhatFits/client (UserPro
$ npm run build

> client@1.0.0 build C:\Users\rsanc\Documents\Github\WhatFits\client
> node build/build.js

Hash: bf7a360d6f1e785ef5ee
Version: webpack 3.10.0
Time: 23164ms

      Asset      Size
static/js/0.c215e256eb5cd08eb2bf.js.map  57.7 kB
static/img/backgroundImage2.171e4ea.jpg  41.6 kB
static/img/navbarLogo.b583f42.png       28.3 kB
static/js/0.c215e256eb5cd08eb2bf.js     12.4 kB
```

[..Lots of files later...]



```
Index.html 894 bytes
static/ProfileDummy/backgroundImage.jpg 28 kB
static/ProfileDummy/backgroundImage2.jpg 41.6 kB
static/genericProfileImage.jpg 31.6 kB
static/ProfileDummy/profilePicture.jpg 128 kB
static/ProfileDummy/profilePicture2.jpg 49.1 kB

Build complete.

Tip: built files are meant to be served over an HTTP server.
Opening index.html over file:// won't work.

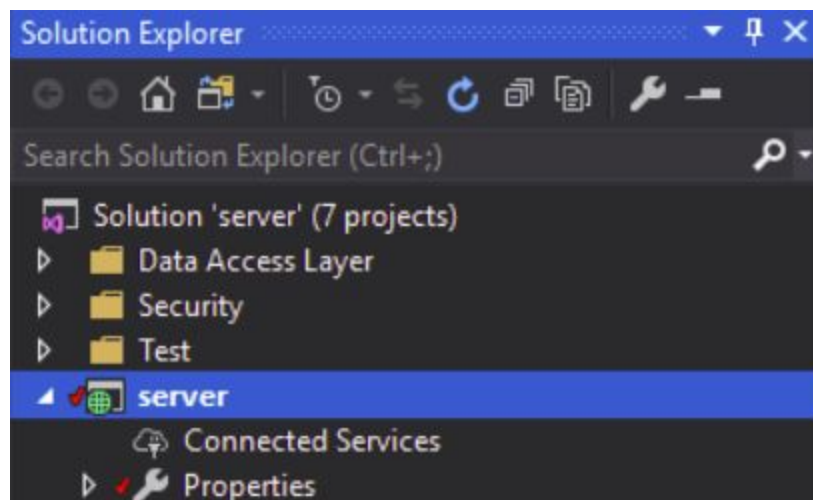
rsanc@Sapphire-WinLaptop MINGW64 ~/Documents/Github/WhatFits/client (UserProf
$
```

At this point you have finished compiling the client side.

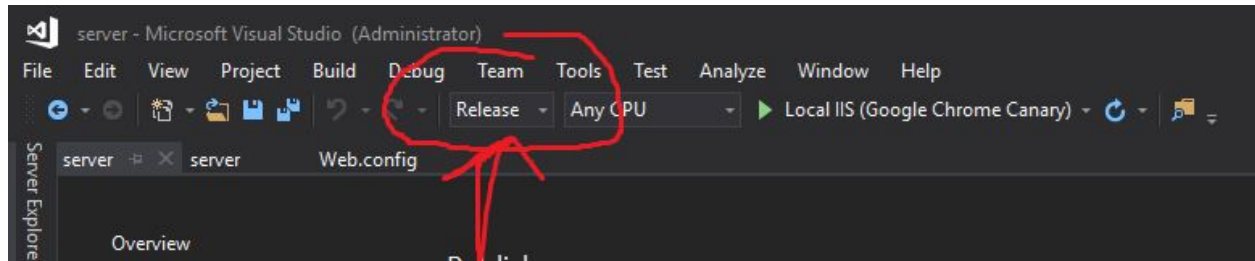
## Publishing the WebAPI portion

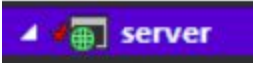
At this point open VISUAL STUDIO COMMUNITY

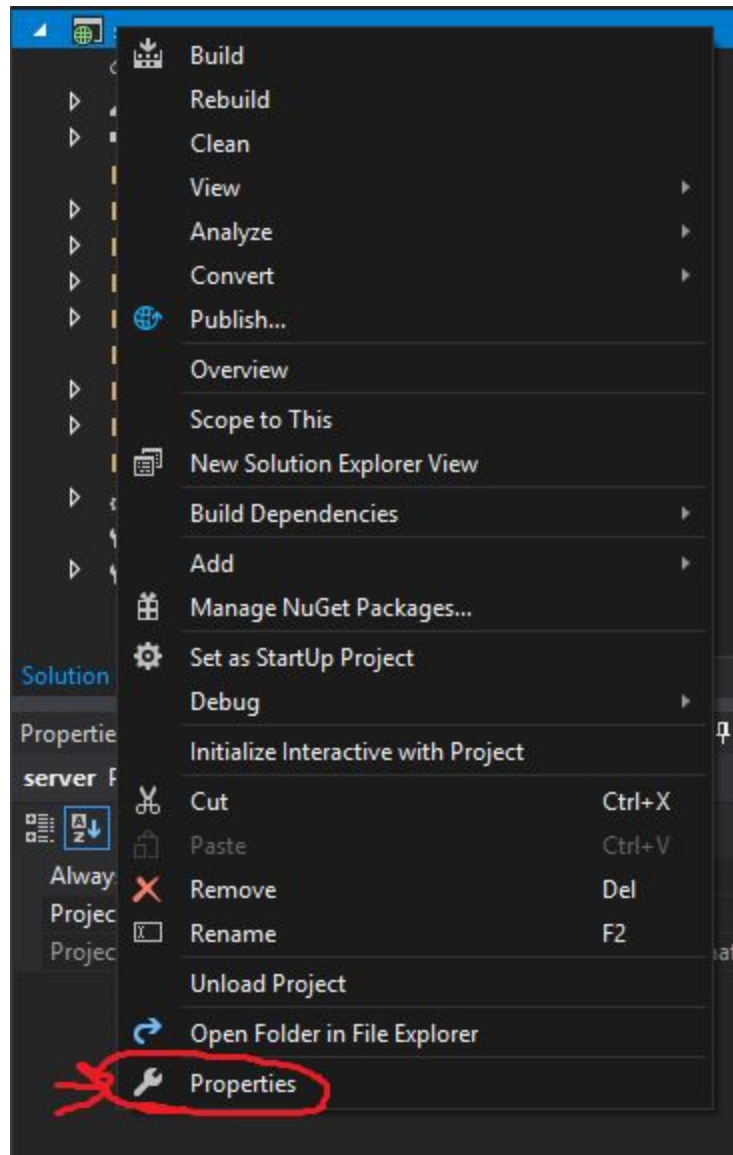
1. Open the server solution



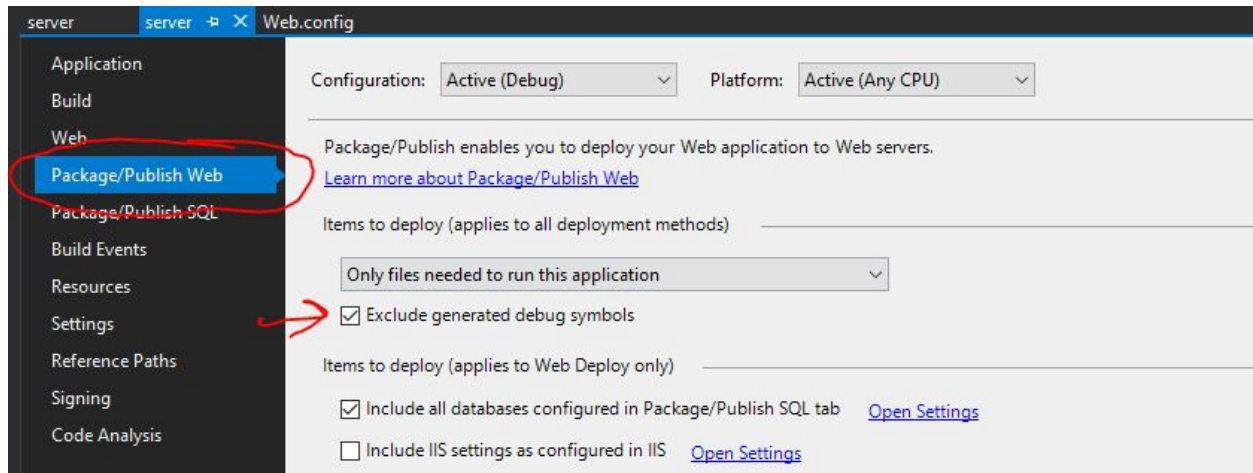
2. On the top bar, click Debug to Release




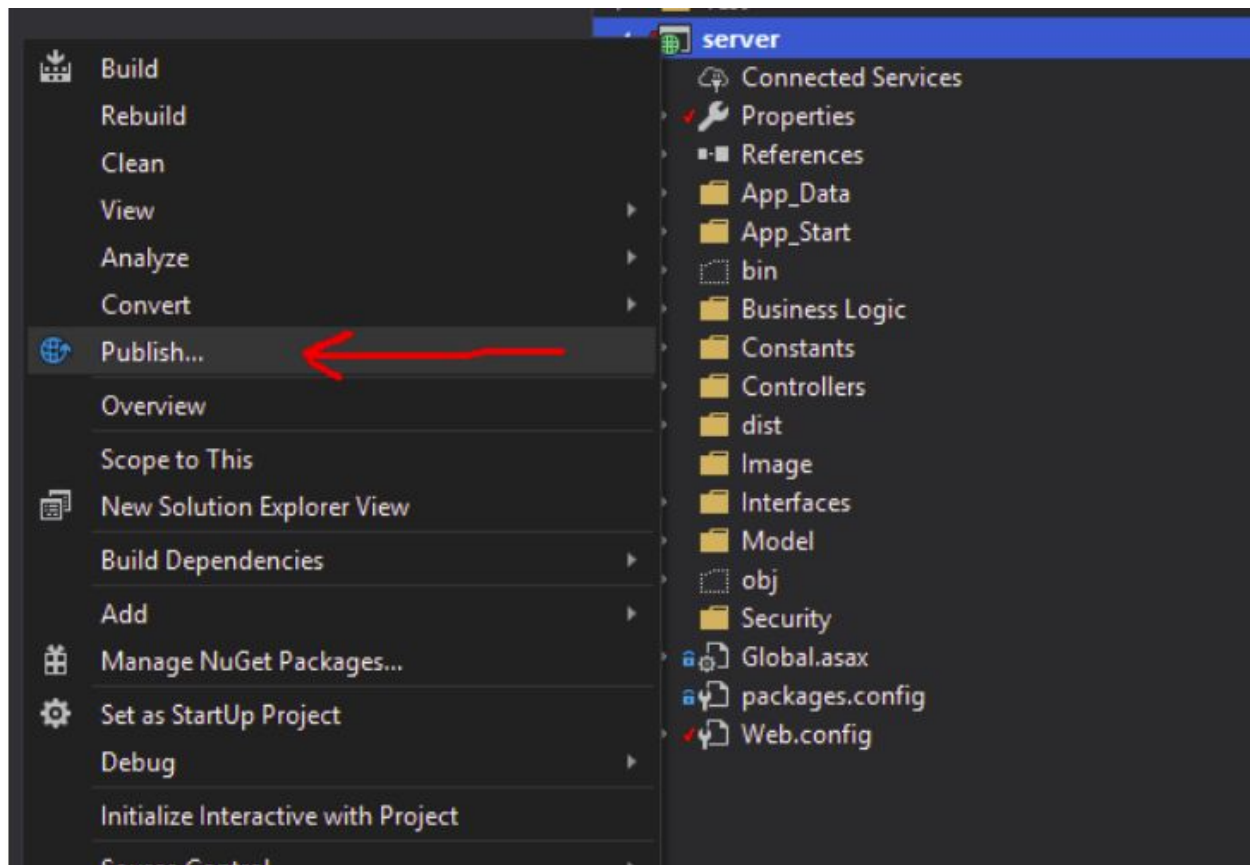
3. On  Right click, then click on properties



4. Then under Package/Publish Web click on Exclude generated debug symbols. This will prevent the creation of .pdb debug files from being created when you published.




5. Then right click on  and click on publish



6. Click on settings

## Publish

Publish your app to Azure or another host. [Learn more](#)

 CustomProfile




Publish

[Create new profile](#)

## Summary

Target Location

C:\Users\rsanc\Downloads\WEBAPI PROJECT 

[Settings...](#)

Delete Existing Files

True

[Rename Profile...](#)

Configuration

Release

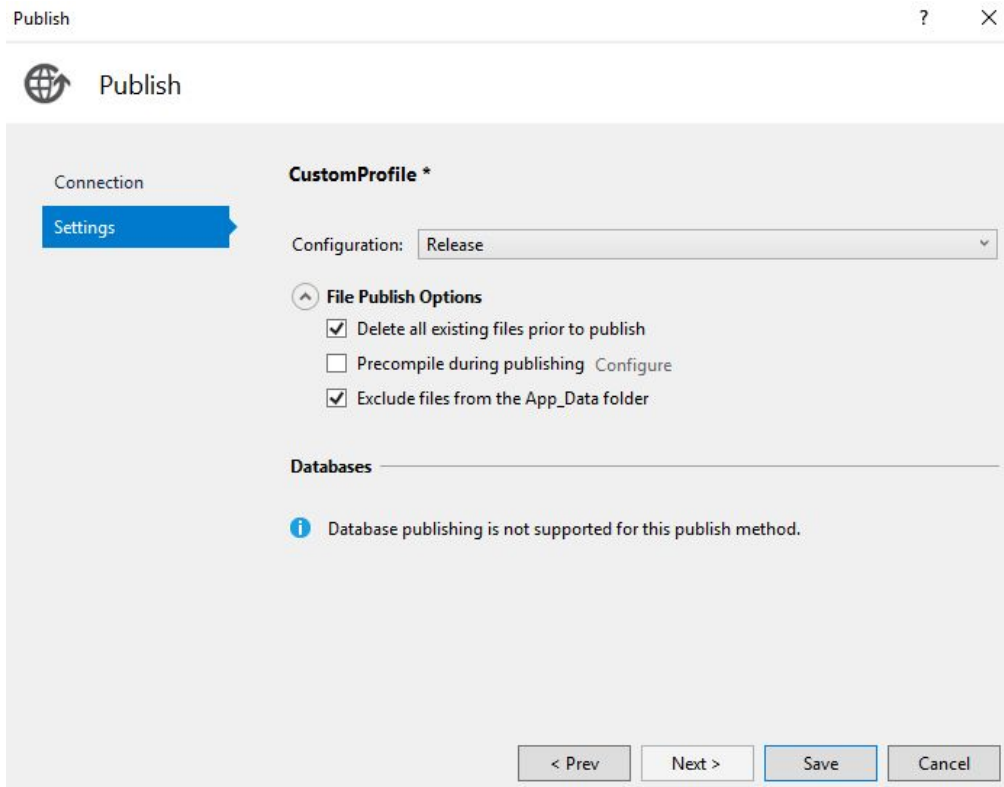
[Delete Profile](#)

## Continuous Delivery

Automatically publish your application to Azure with continuous delivery

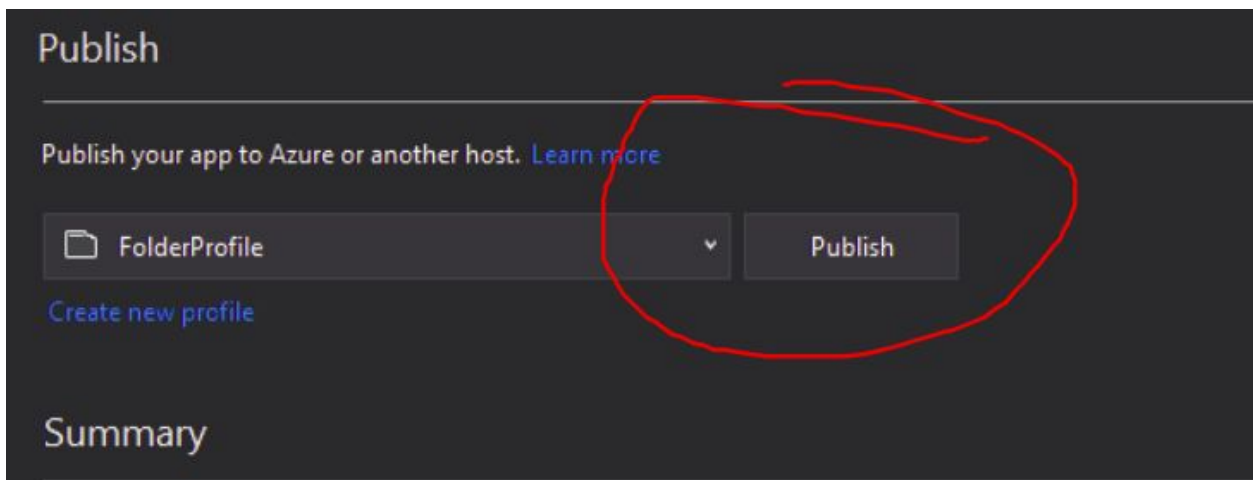
Start

## 7. Change the following



## 8. Click on Save




## 9. Then Publish to the default location



## 10. At this point if it succeeds you should have a folder in downloads with the published files.

```
Output
Show output from: Build
1>----- Build started: Project: server, Configuration: Release Any CPU -----
1> server -> C:\Users\rsanc\Downloads\WhatFits-master\server\server\bin\server.dll
2>----- Publish started: Project: server, Configuration: Release Any CPU -----
2>Connecting to C:\Users\rsanc\Downloads\WEBAPI PROJECT...
2>Transformed Web.config using C:\Users\rsanc\Downloads\WhatFits-master\server\server\Web.Release.conf
2>Copying all files to temporary location below for package/publish:
2>obj\Release\Package\PackageTmp.
2>Deleting existing files...
2>Publishing folder /...
2>Publishing folder bin...
2>Publishing folder bin/roslyn...
2>Web App was published successfully file:///C:/Users/rsanc/Downloads/WEBAPI%20PROJECT
2>
===== Build: 1 succeeded, 0 failed, 5 up-to-date, 0 skipped =====
===== Publish: 1 succeeded, 0 failed, 0 skipped =====
|
```

11. In the folder that was in the publishing settings

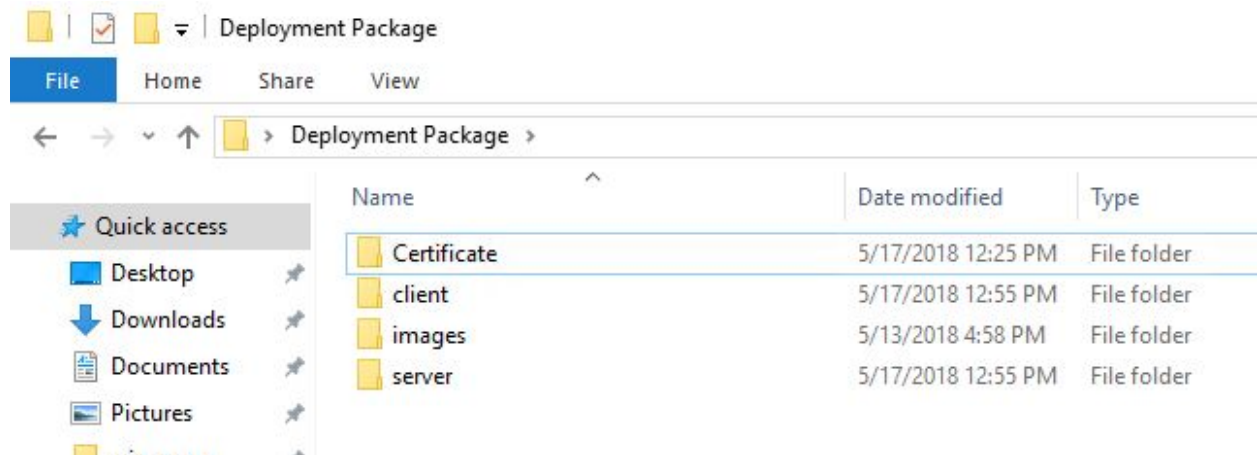
Name	Date modified	Type
 bin	5/17/2018 12:42 PM	File folder
 Global.asax	5/17/2018 12:39 PM	ASAX File
 Web.config	5/17/2018 12:42 PM	CONFIG File

## END OF PUBLISHING FILES



## Part 2: Uploading the folder to the server

1. Remote Desktop Connection should be open during deployment. Else here are the credentials for it
  - a. IP Address: XXXXXXXX
  - b. UserName: XXXXX
  - c. Password: XXXXX
2. Remember that we have published **two folders**, one called **dist** and one called **WEB API PROJECT**. These two folders represent both the client portion of the web application and the backside of the web application.
3. The contents in the **WEB API PROJECT** will be placed inside the folder called '**sever**' on the web server hosting our application. The contents of the **dist** will be placed inside a folder called '**client**' in the web server.
4. One open, **Drag and drop** the folder you published on the previous steps into the server
5. **NOTE:** We are not showing this step but you should at least mention that that's how we are moving our stuff on there.
6. **NOTE:** For the presentation we are having the deployed folder with the published files on there. These files will be located in the following directory: **DESKTOP**



Everything you need for deployment will be in this folder

## END OF PUSHING TO SERVER

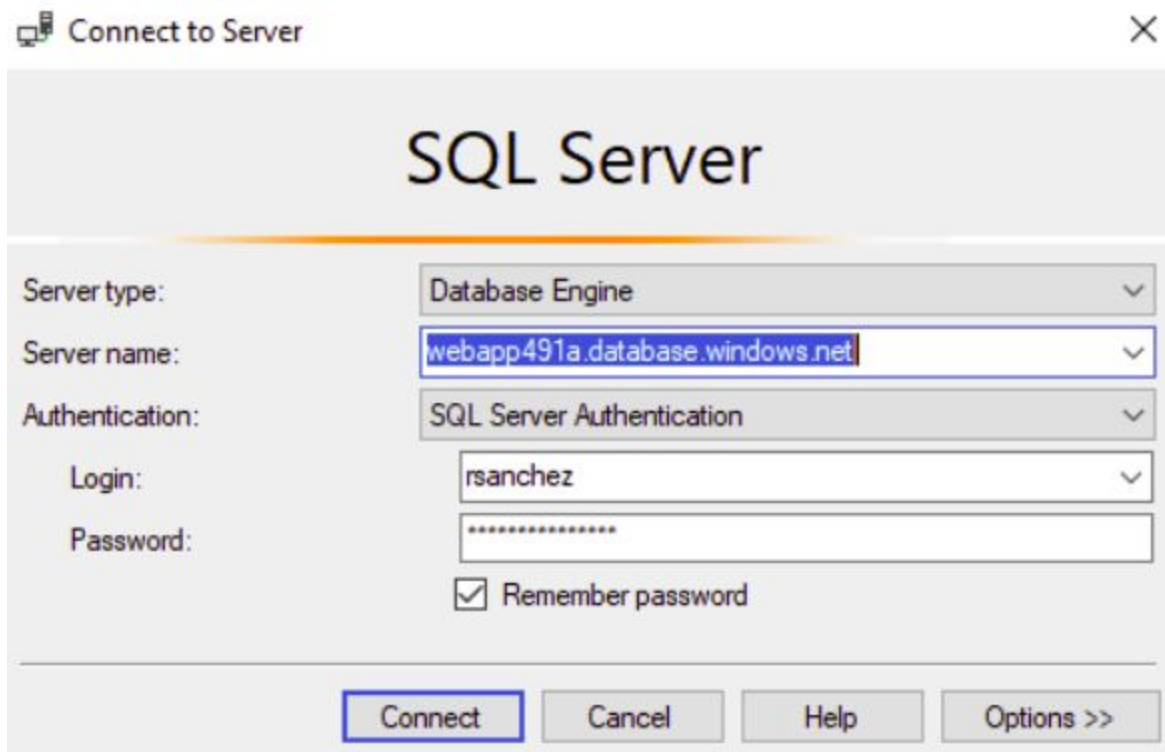


## Part 3: Setting up the Database

---

This portion requires SQL Server Management Studio (SSMS) to connect to the remote database (Azure) to not only show that its empty but that we can create it on the spot with the predetermined seed data.

1. Open SSMS
2. Connect to the Azure database (**NOTE: CSULB firewall causes problems, use hotspot**)

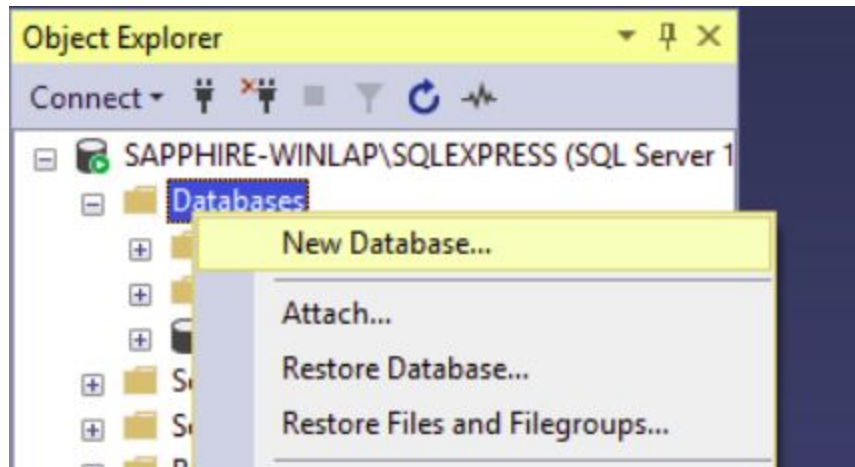


**ServerName:** webapp491a.database.windows.net

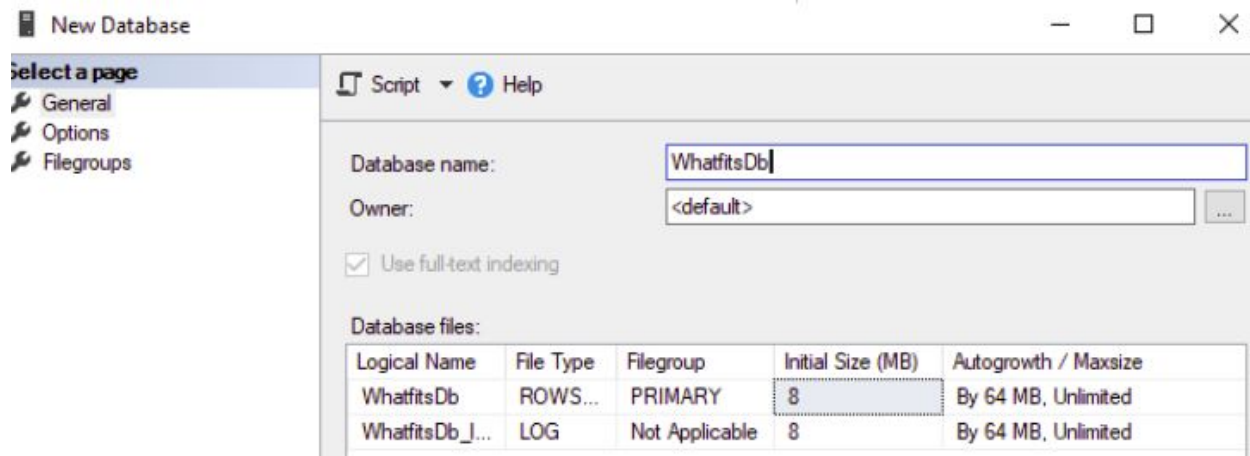
**Login:** XXXXX

**Password:** XXXXXXXXXXXX

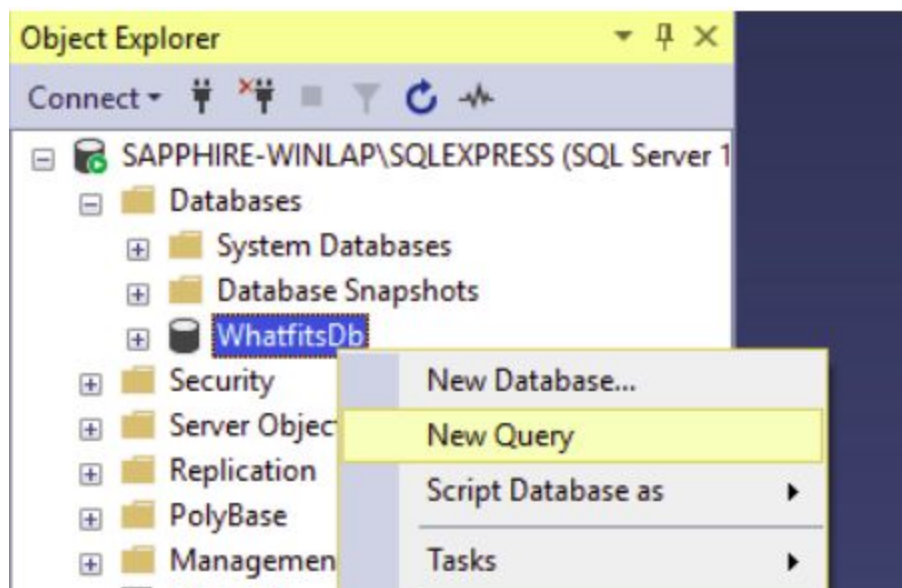
3. Show that it is empty
4. Right Click on the database folder and create a database



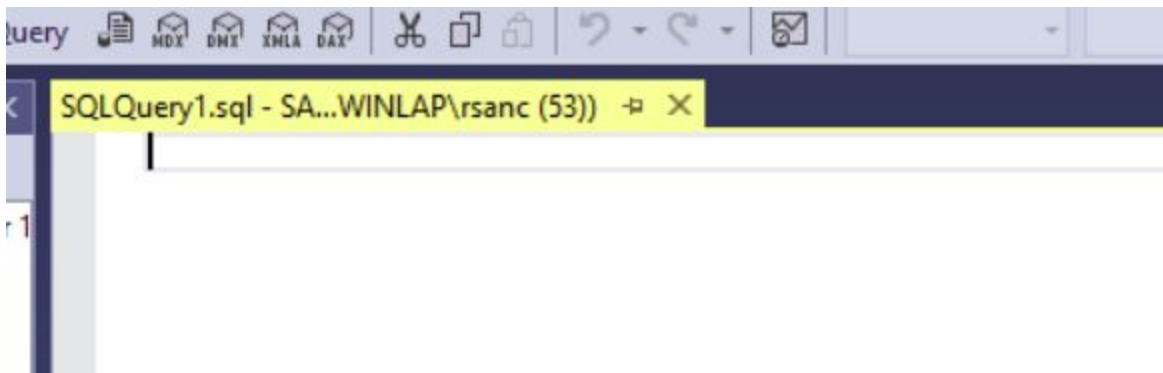
5. Name the database (WhatfitsDb) exactly



6. Right Click on the newly created database and click on New Query



7. At this point you should have a sql script file with the table creation code and seed data. Copy and paste that into the query page in SSMS and Execute.



Copy seed data into that and click on execute

8. You should see stuff being printed at the output.
9. Expand the WhatfitsDb and expand the table folder to see tables. Double check if the seed data was created by seeing data in the following tables:
  - a. Security Questions
    - i. We need this for registering new users. Else you won't be able to register anyone.
  - b. Credentials
    - i. We will have 5 pre-seeded users, one administrator, and four general users to show off search.
10. At this point the server has been setup

## END OF DATABASE SETUP

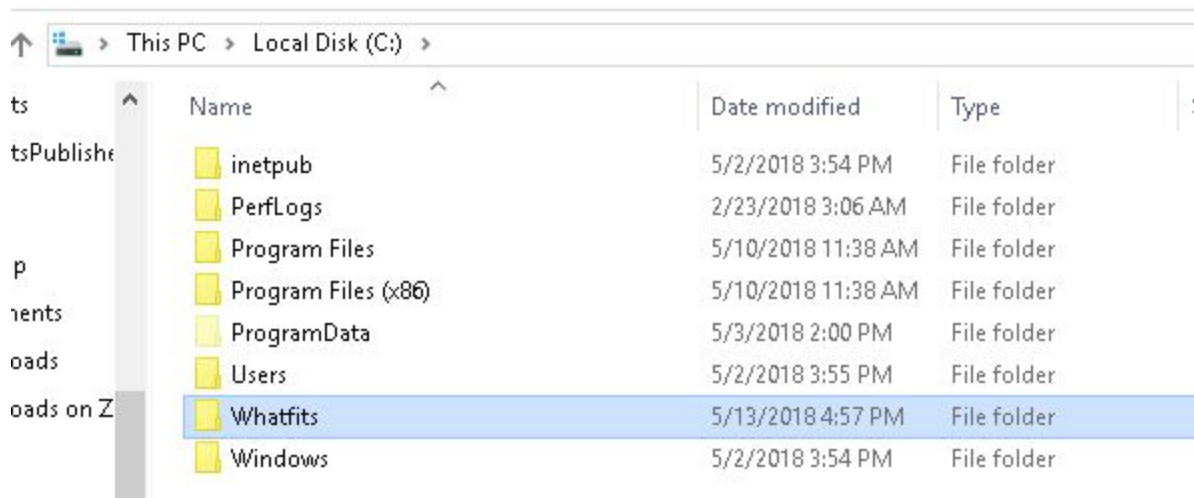
## Part 4: Setting up files / folders in Server

---

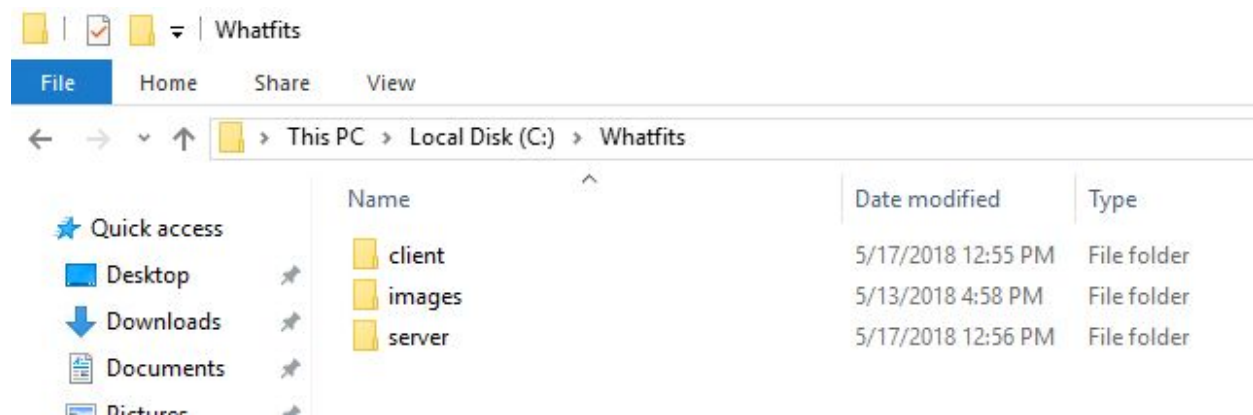
### Moving the files

We have agreed to host our files in the C:// drive under the folder Whatfits. **NOTE: A batch file would be used if time permitted.**

1. Go to the C:// Drive and create a new folder called **Whatfits**



2. Move the folders in the **Deployment package** folder located in the Desktop to the Whatfits folder on the C drive



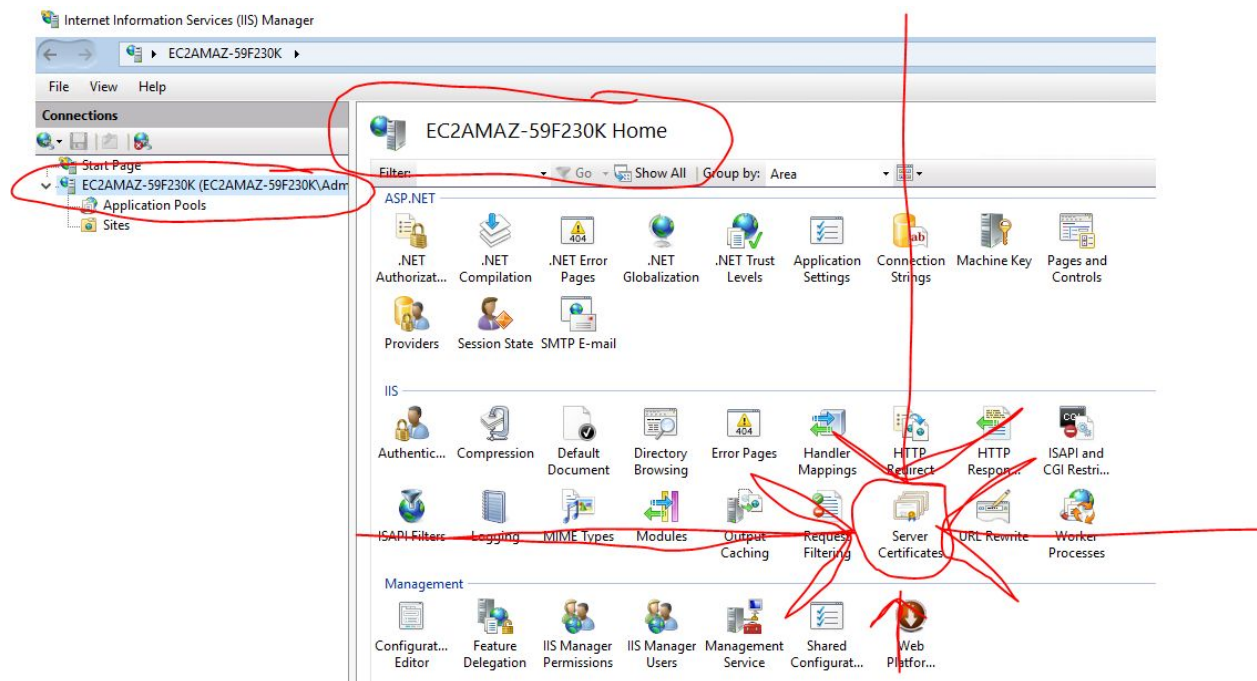
END OF SETTING UP FILES

## Part 5: Deploying the web application

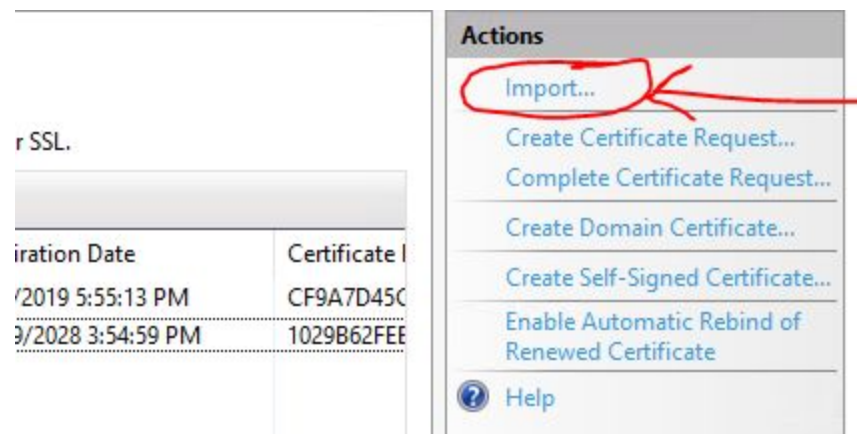
This portion of the deployment deals with deploying both the client, backend, and images applications to be accessible from the web.

### Installing Certificate on IIS

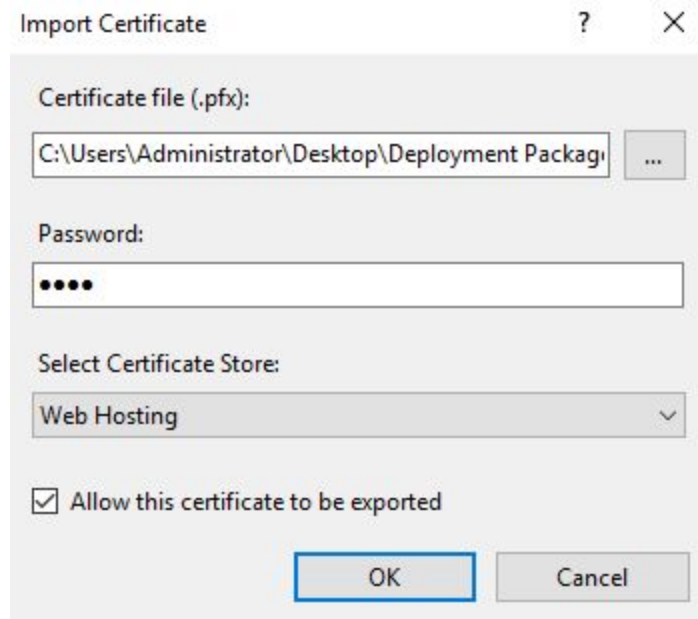
1. Double click on the web server and double click on **Server Certificates**



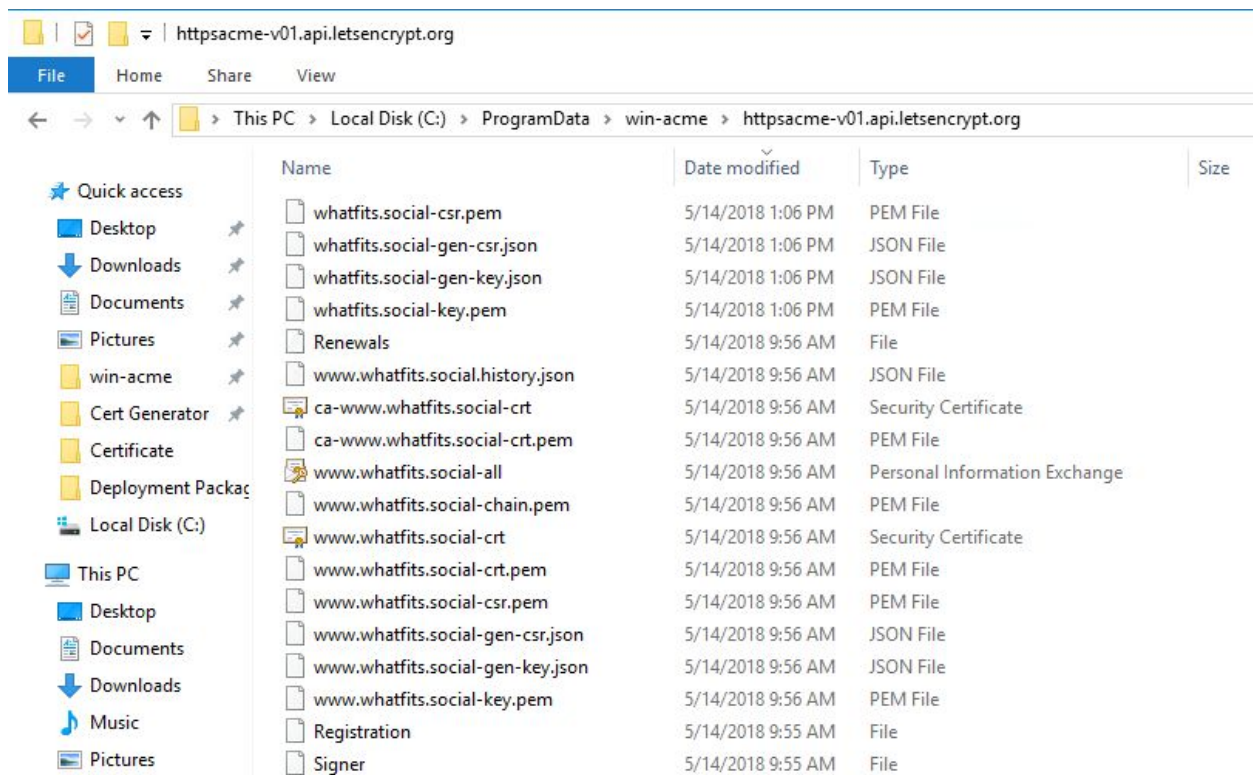
2. Click on Import



3. Find the certificate on the **deployment package** on the desktop. Password is **asdf**, and make it **Web Hosting**



4. Click OK
5. Copy all the files in side the **Certificate Files** Folder in the **Deployment Package** and paste them in the following directory  
(C:\ProgramData\win-acme\httpsacme-v01.api.letsencrypt.org)

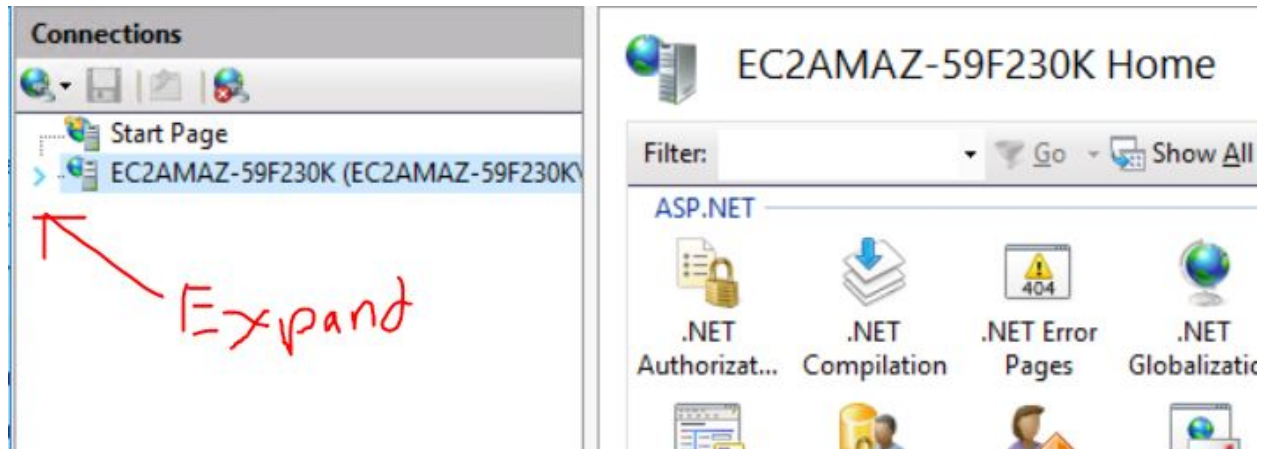


6. Done



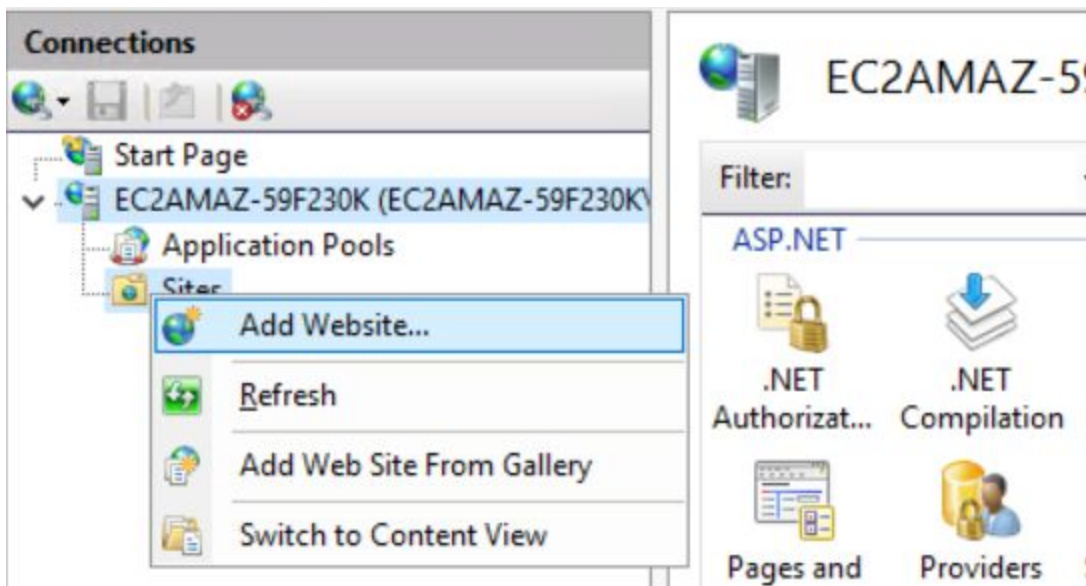
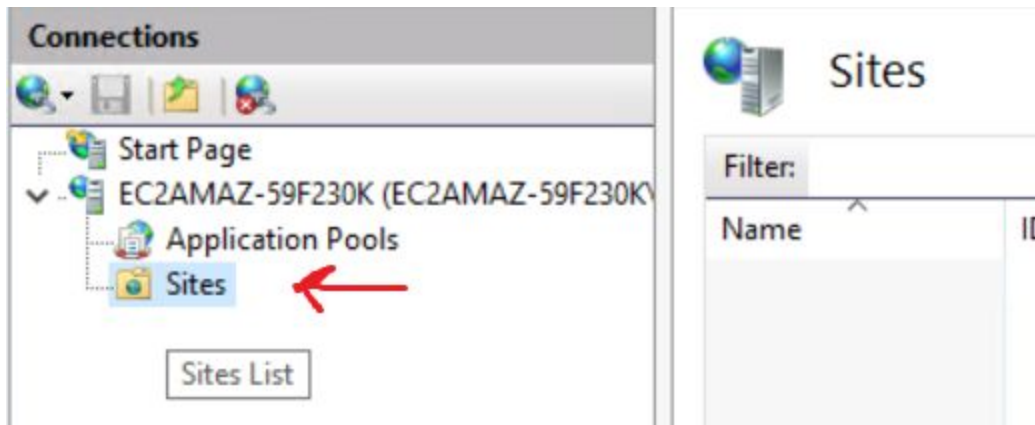
## Client-side Deployment

1. Open Internet Information Service (IIS)
2. Expand the server options on the left





3. Right Click on sites and click on Add



4. Add the following data (**NOTE: ADD CLIENT NOT DIST**)

Add Website

Site name: Whatfits Application pool: Whatfits Select...

Content Directory

Physical path: C:\Whatfits\WhatfitsPublished\dis client ...

Pass-through authentication

Connect as... Test Settings...

Binding

Type: https IP address: 172.31.3.249 Port: 443

Host name: www.whatfits.social

☒ Require Server Name Indication

SSL certificate: www.whatfits.social 2018/5/14 9:56:01 AM Select... View...

☒ Start Website immediately

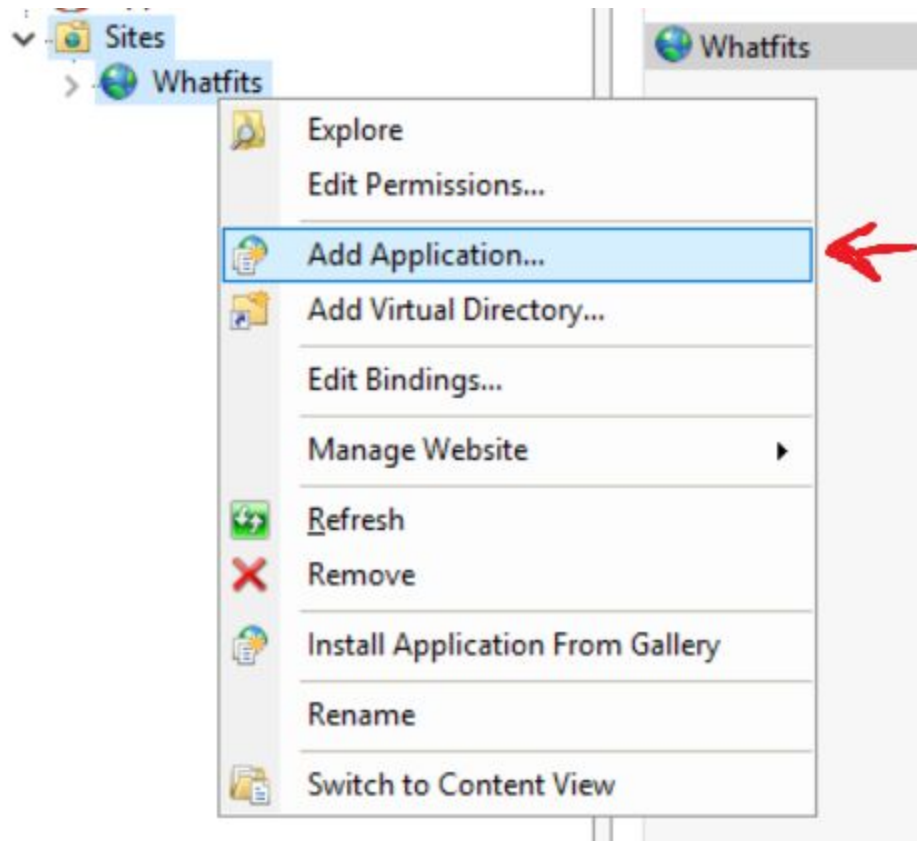
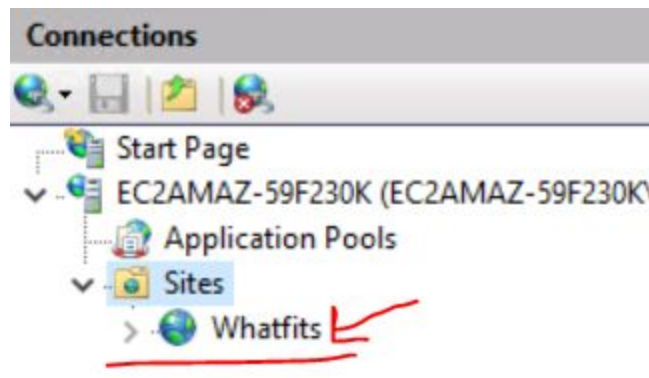
OK Cancel

5. At this point the client side has been deployed.

## END OF CLIENT SIDE DEPLOYMENT

## Backend Deployment

1. Right click on the newly created website



2. Click on Add Application. Then enter the following:  
**CLICK ON SERVER ON WHATFITS IN C DRIVE**

Add Application ? X

Site name: Whatfits  
Path: /

Alias: server Application pool: .NET v4.5 Select...

Example: sales

Physical path: C:\Whatfits\WhatfitsPublished server ...

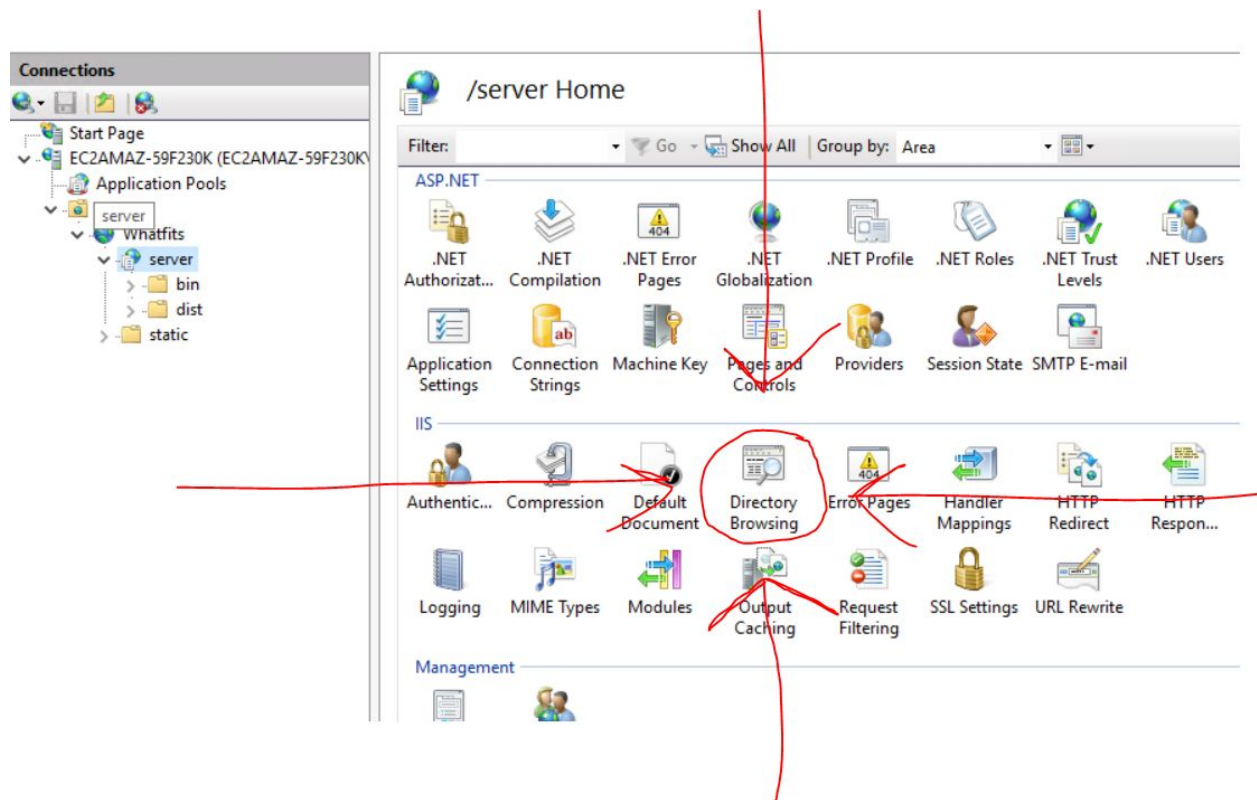
Pass-through authentication  
Connect as... Test Settings...

☐ Enable Preload

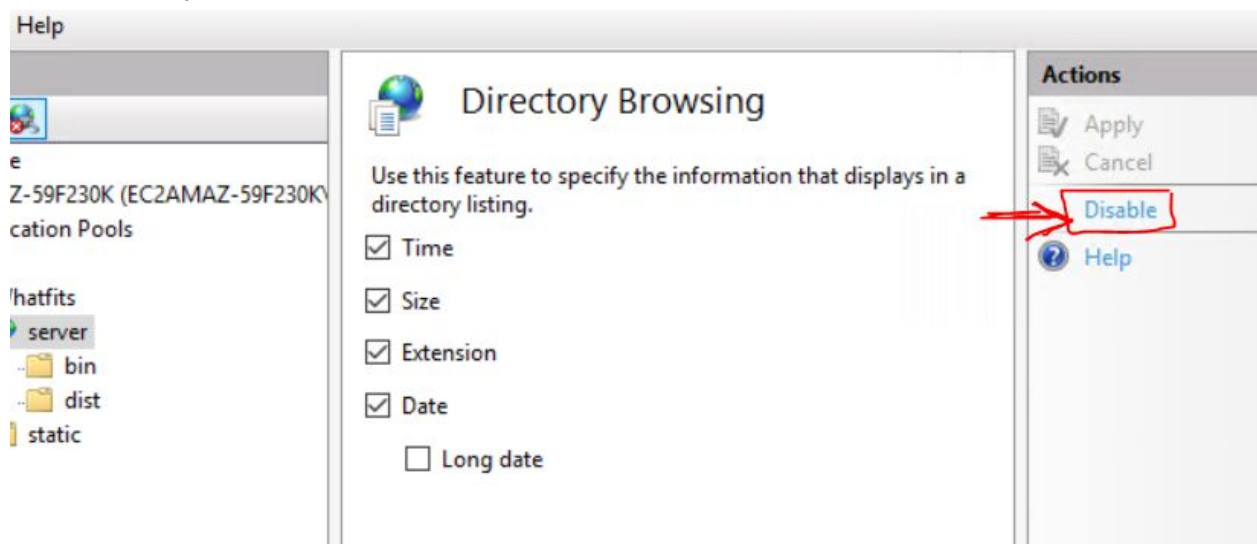
OK Cancel

3. At this point the server has been deployed

4. Next you need to disable directory browsing to ensure that no one outside the system can navigate through the files.



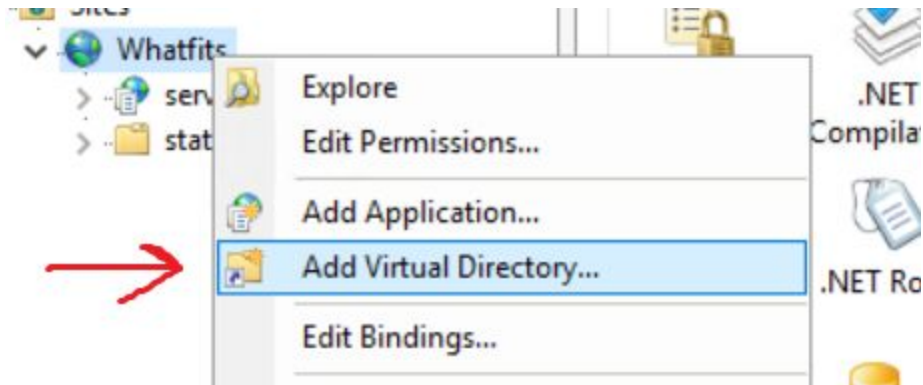
5. Make sure you disable that



# END OF BACKEND DEPLOYMENT

## Setting up Images Folder

1. Next it's time to deploy our images project. Right click on What fits Site and click on add Virtual Directory



2. Fill in the following information

Add Virtual Directory

Site name: Whatfits  
Path: /

Alias:  
images

Example: images

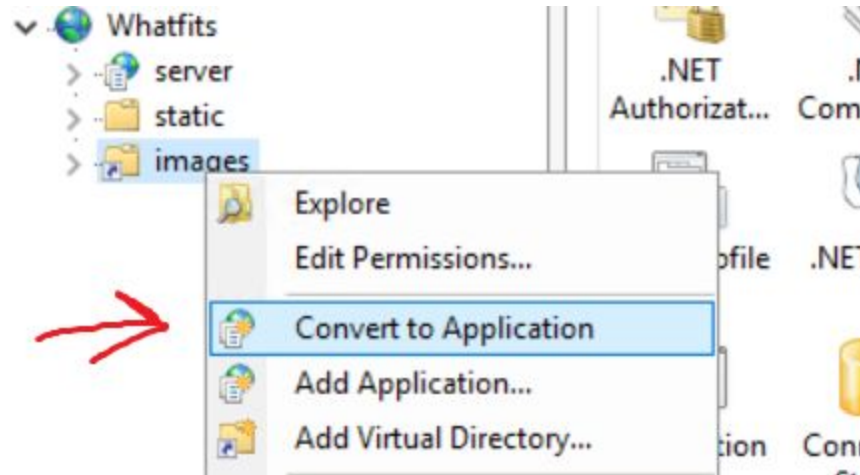
Physical path:  
C:\Whatfits\images

Pass-through authentication  
Connect as... Test Settings...

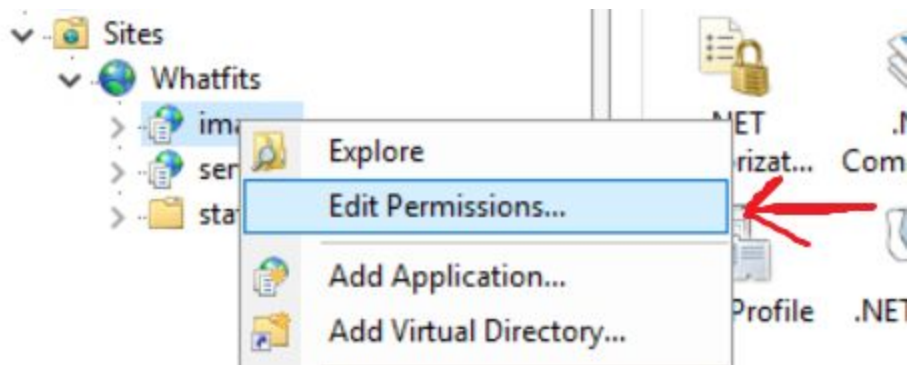
OK Cancel

A screenshot of the 'Add Virtual Directory' dialog box. The 'Site name' is 'Whatfits' and the 'Path' is '/'. The 'Alias' field contains 'images' and has a red arrow pointing to it. The 'Physical path' field contains 'C:\Whatfits\images' and has a red arrow pointing to the browse button (three dots in a box) next to it. The 'Example' field shows 'images'. At the bottom, there are buttons for 'Connect as...', 'Test Settings...', 'OK', and 'Cancel'.

3. Then, you need to convert the virtual directory into an application

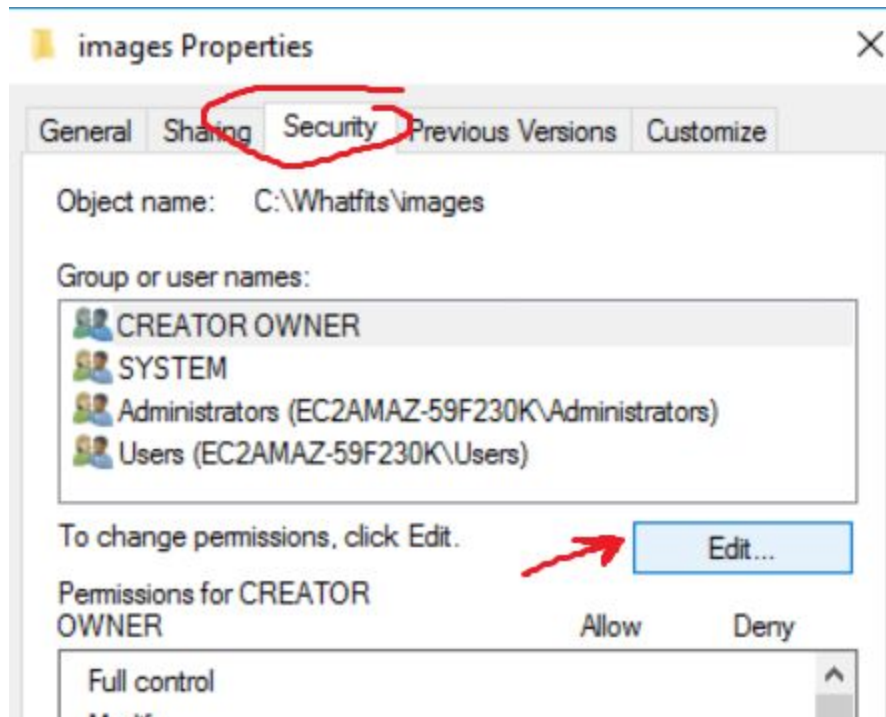


4. A window will open, just click on OK
5. Right click on the images application and click on edit permissions

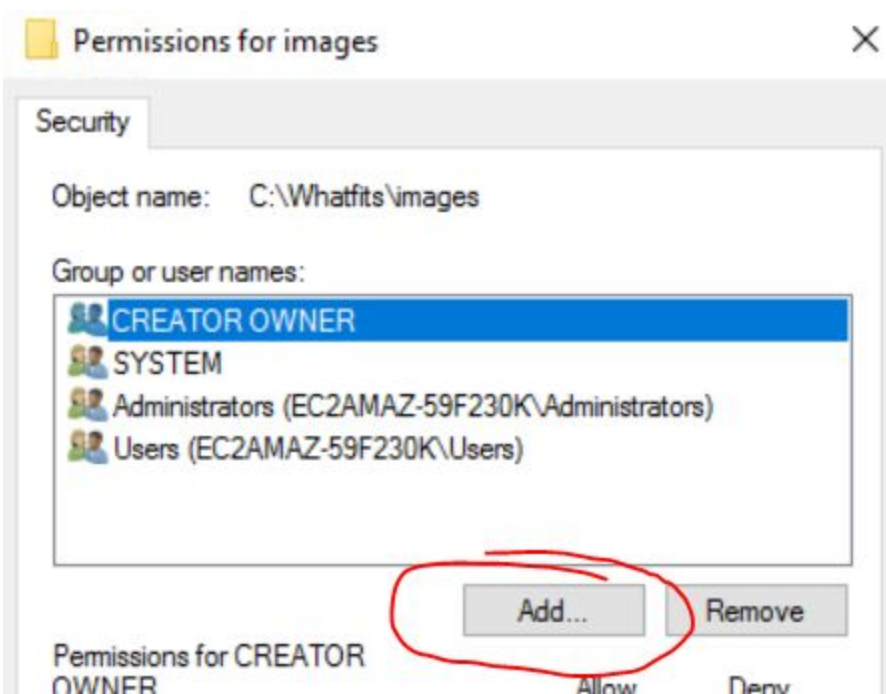




6. On the image properties, click on the Security Tab and click on Edit

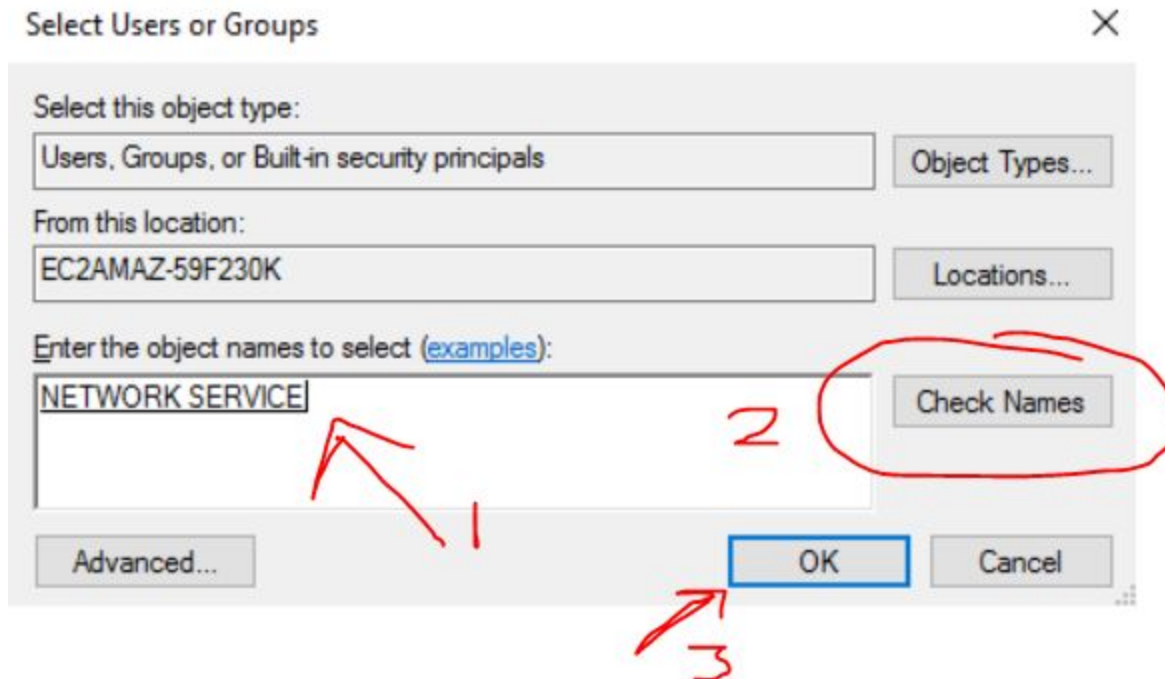


7. Click on Add





8. In the Enter the object name type NETWORK SERVICE. Then click on the CHECK NAMES, the word should now appear underlined. Click OK.



9. Click OK on the following windows until you are done.

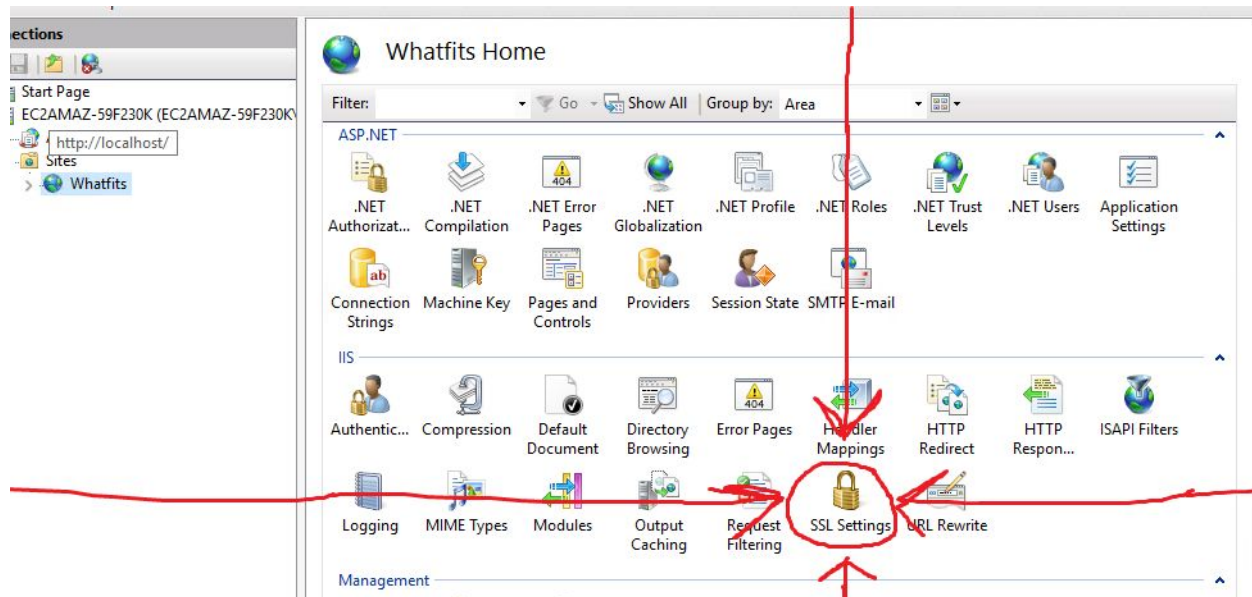
## END OF IMAGE DEPLOYMENT

## Part 6: Configuring IIS

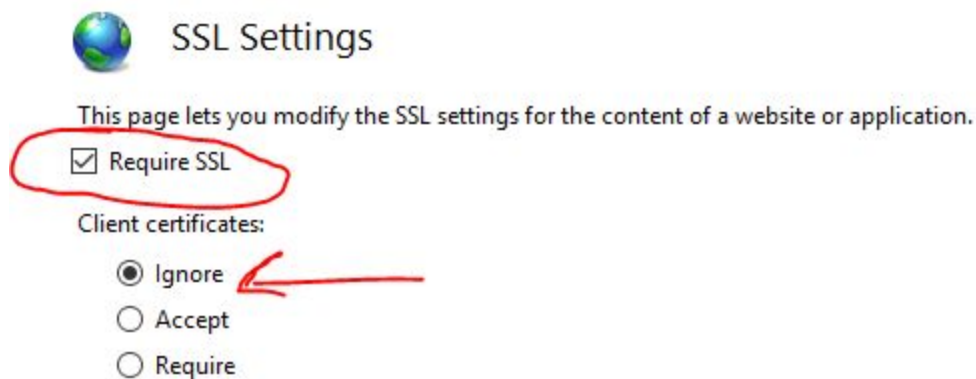
This part will configure IIS to handle HTTPS request to the server.  
Disabling Directory Browsing on server and images

### Enabling SSL on Client

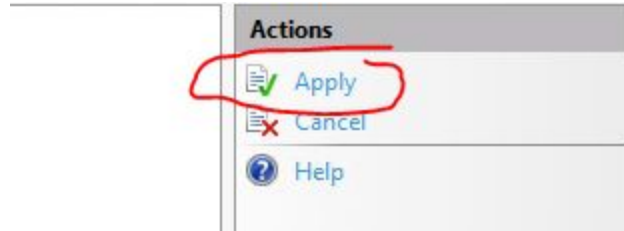
1. On the client side, Double click on SSL Settings



2. Click on enable SSL, and leave it as Ignore



3. Click on Apply

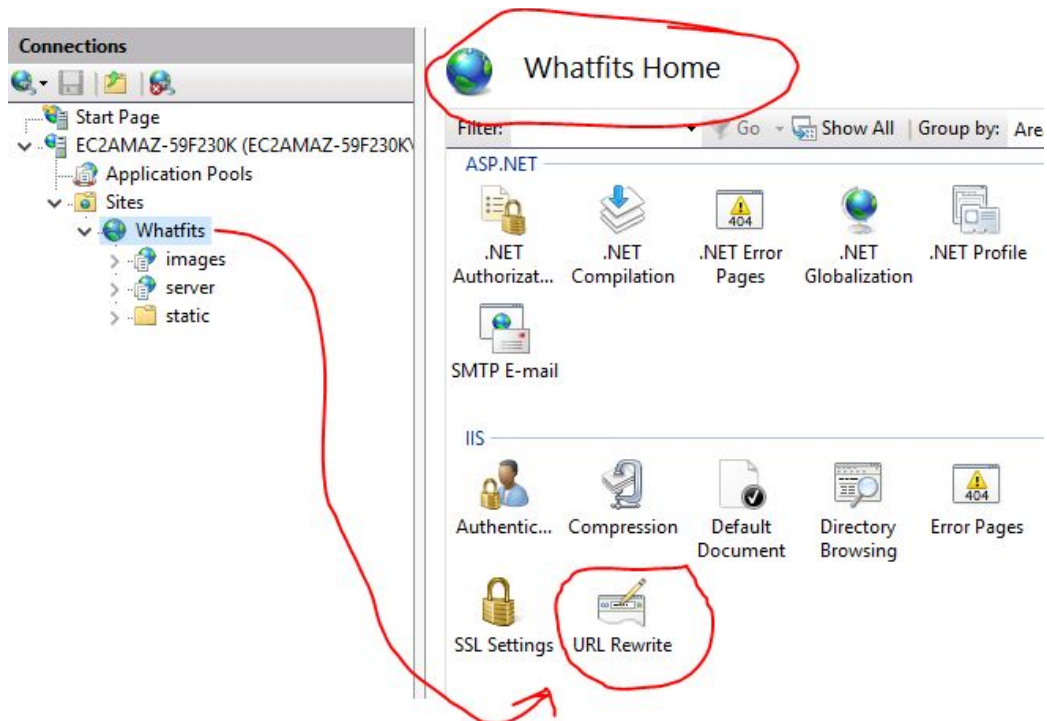


4. Done

## Doing URL Rewrite on Client

### Create Rule

1. Click on **URL Rewrite**



2. Click **Add Rule(s)**

e requested URL address and the content of an HTTP response.

. address:

	Match	Pattern	Action Type
after '/'	Matches	(.*)	Redirect
OST}	Does Not Match the Pattern	^www\.whatfits\.social\$	

**Actions**

- Add Rule(s)...
- Revert to Parent
- Manage Server Variables**
  - View Server Variables...
- Manage Providers**
  - View Rewrite Maps...
  - View Providers...
- Inbound Rules** ^
  - Import Rules...
- Outbound Rules** ^
  - View Preconditions...
  - View Custom Tags...
- Help

### 3. Click **Canonical domain name**

Add Rule(s) ? X

Select a rule template:

**Inbound rules**

- Blank rule
- Request blocking
- Rule with rewrite map

**Inbound and Outbound Rules**

- User-friendly URL
- Reverse Proxy

**Outbound rules**

- Blank rule
- Search Engine Optimization (SEO)
- Enforce lowercase URLs
- Canonical domain name
- Append or remove the trailing slash symbol

Select this template to create a rule that you can use to redirect Web clients to a primary domain for the Web site.

OK Cancel

### 4. Input **www.whatfits.social** then click OK

Add a rule to redirect to a canonical domain name

? X

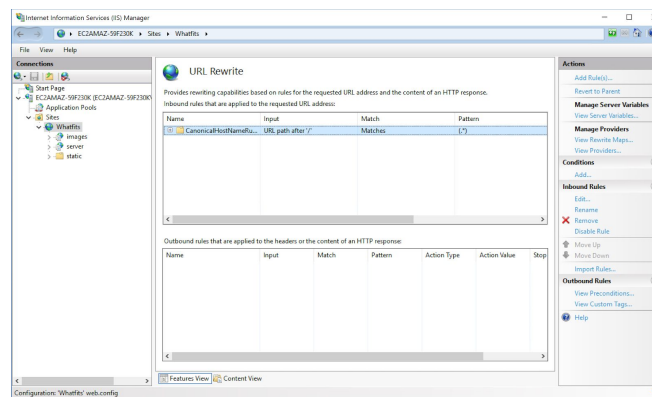
Search engines treat Web sites that can be accessed by more than one domain name, such as [www.contoso.com](http://www.contoso.com) and [contoso.com](http://contoso.com), as if they are two different sites. This affects the page ranking for the Web site. Use this rule template to create a redirect rule that will enforce the use of a single domain name for the Web site.

Select the primary host name:

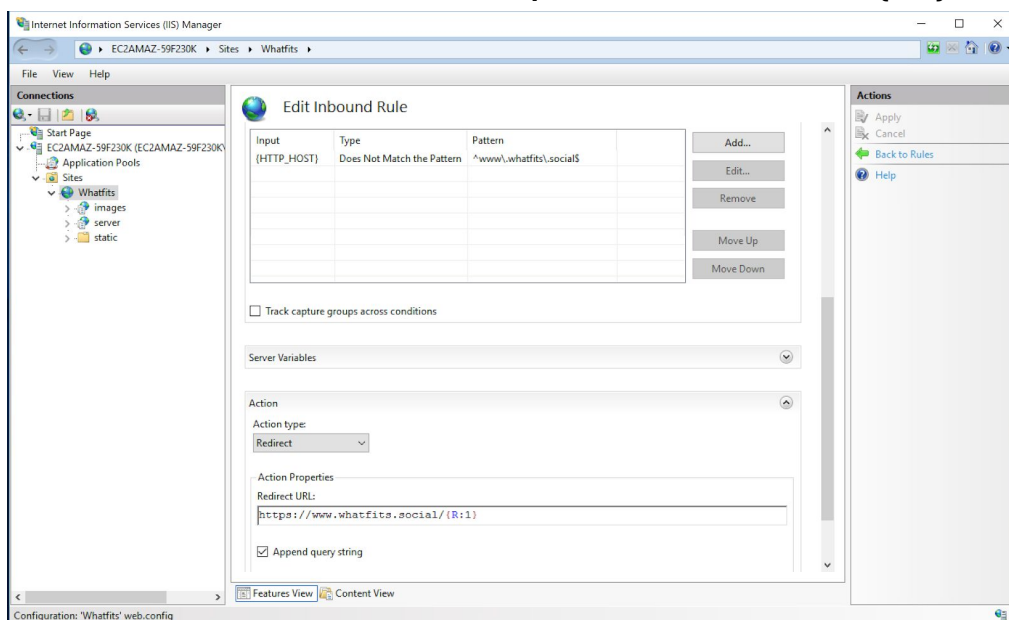
OK

Cancel

## 5. Edit... rule

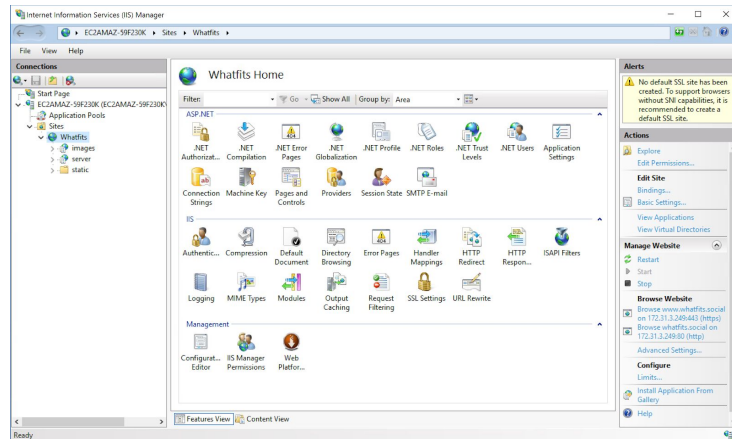


## 6. Scroll down and edit **Redirect URL:** to **https://www.whatfits.social/{R:1}**

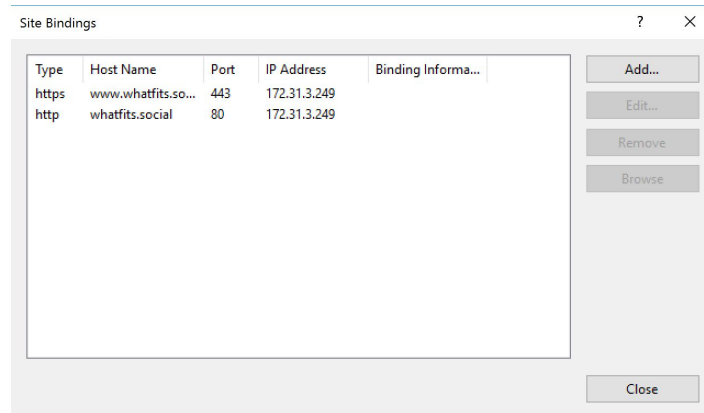


## Add Bindings

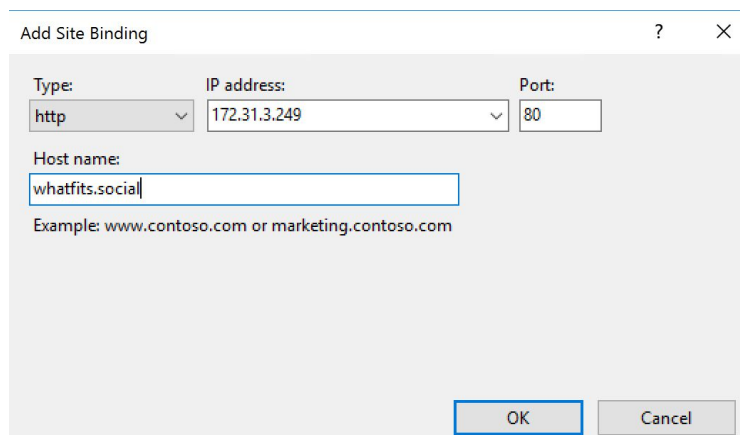
1. Go back to Whatfits Home then click **Bindings...**



2. Click **Add**



3. Change IP address and Host name then click **OK**



# DONE

## Misc Stuff:

**Security about the database:** Azure Database only allows certain IP addresses access to their database if they have the credentials. The credentials for the database are different from the Azure account associated with it.

**Getting the certificate:** Our certificate is pre-installed on IIS. Due to us being limited to only 5 certificated per week see (Letsencrypt policy on how many certificates per domain are allowed) we can not demo this part multiple times due to that restriction. There may be an alternative to that where we export the installed certificate and remove it from IIS and later during demo import it.

Let's Encrypt program from <https://github.com/Lone-Coder/letsencrypt-win-simple/releases>

```
[INFO] A Simple ACME Client for Windows (WACS)
[INFO] Software version 1910.1.6696.24132 (RELEASE)
[INFO] IIS version 10.0
[INFO] ACME server https://acme-v01.api.letsencrypt.org/
[INFO] Please report issues at https://github.com/PKISharp/win-acme
```

### N: Create new certificate

```
M: Create new certificate with advanced options
L: List scheduled renewals
R: Renew scheduled
S: Renew specific
A: Renew *all*
V: Revoke certificate
C: Cancel scheduled renewal
X: Cancel *all* scheduled renewals
Q: Quit
```

Please choose from the menu: **n**

```
[INFO] Running in Simple mode
```

### 1: Single binding of an IIS site

```
2: SAN certificate for all bindings of an IIS site
3: SAN certificate for all bindings of multiple IIS sites
4: Manually input host names
C: Cancel
```

Which kind of certificate would you like to create?: 1

### 1: www.whatfits.social (SiteId 1) [@C:\Whatfits\dist]

```
C: Cancel
```

Choose site: 1

[INFO] Plugin IISBinding generated target [IISBinding] www.whatfits.social (SiteId 1) [ @ C:\Whatfits\dist]  
[INFO] Authorize identifier: www.whatfits.social  
[INFO] Authorizing www.whatfits.social using http-01 validation (SelfHosting)  
[INFO] Answer should now be browsable at  
http://www.whatfits.social/.well-known/acme-challenge/W\_L9FP\_kuuZnWyG5v5qVj-wdYickk  
v\_ZBQ2vuI2O\_JM  
[INFO] Authorization result: valid  
[INFO] Requesting certificate www.whatfits.social 2018/5/13 17:15:31 PM  
[INFO] Saving certificate to  
C:\ProgramData\win-acme\httpsacme-v01.api.letsencrypt.org  
[INFO] Installing certificate in the certificate store  
[INFO] Adding certificate www.whatfits.social 2018/5/13 17:15:31 PM to store  
WebHosting  
[INFO] Installing with IIS...  
[INFO] Adding new https binding www.whatfits.social:443  
[INFO] Committing 1 https binding changes to IIS  
[INFO] IIS will serve the new certificates after the Application Pool IdleTimeout has been reached.  
[INFO] Adding renewal for www.whatfits.social  
[INFO] Next renewal scheduled at 2018/7/8 0:15:32 AM