



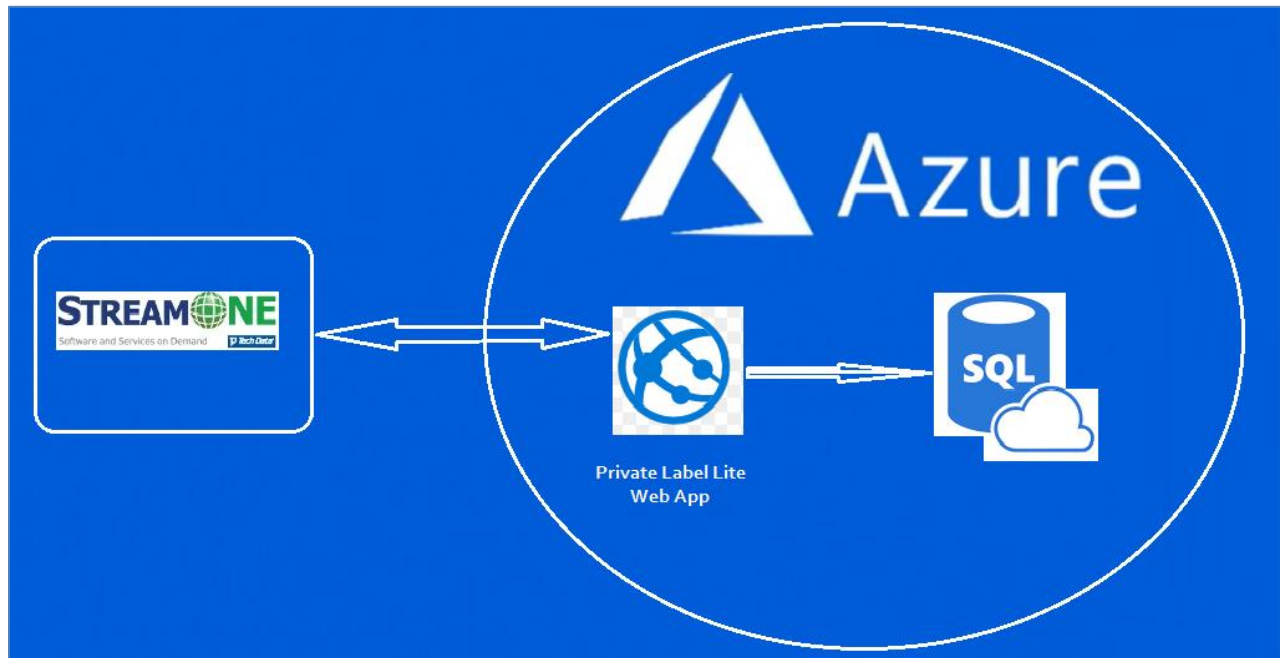
App Deployment- Reseller Technical Document

Table of Contents

Open source code	3
Minimum Requirements	3
MSFT Authentication.....	3
E-mail Server.....	4
Hosting Steps.....	4
Azure Account Setup.....	4
Microsoft Account Creation & Registration	14
Code Deployment	18

Open Source Code

Simple overview of the application with the components details:



Minimum Requirements

1. Must have Microsoft account.
2. Azure account with active subscription.
 - a. Web app [services pack: D1, shared tier]
 - b. SQL Server [**SQL database recommended:** Basic tier Single DataBase (5 DTU)]

MSFT Authentication

This application needs to be registered on the Microsoft developer portal. Please refer to the hosting section for more details.

Please refer to the below link for more technical details around Microsoft authentication :

<https://docs.microsoft.com/en-us/aspnet/core/security/authentication/social/microsoft-logins?tabs=aspnetcore2x>

E-mail Server

We are using a Microsoft account to send emails, therefore a Microsoft account is used as the email server in this application.

Hosting Steps

1. Azure account creation and setup
2. Microsoft account creation & app registration
3. Code deployment.

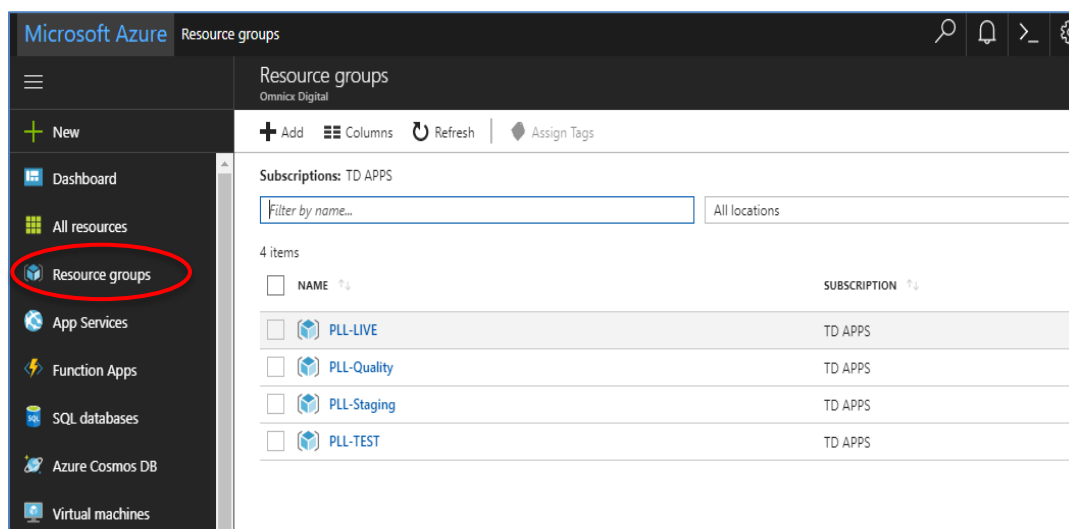
Azure Account Setup

This document will help you to setup the hosting on the Azure portal. Minimum Requirement are already shared to host and deploy the code.

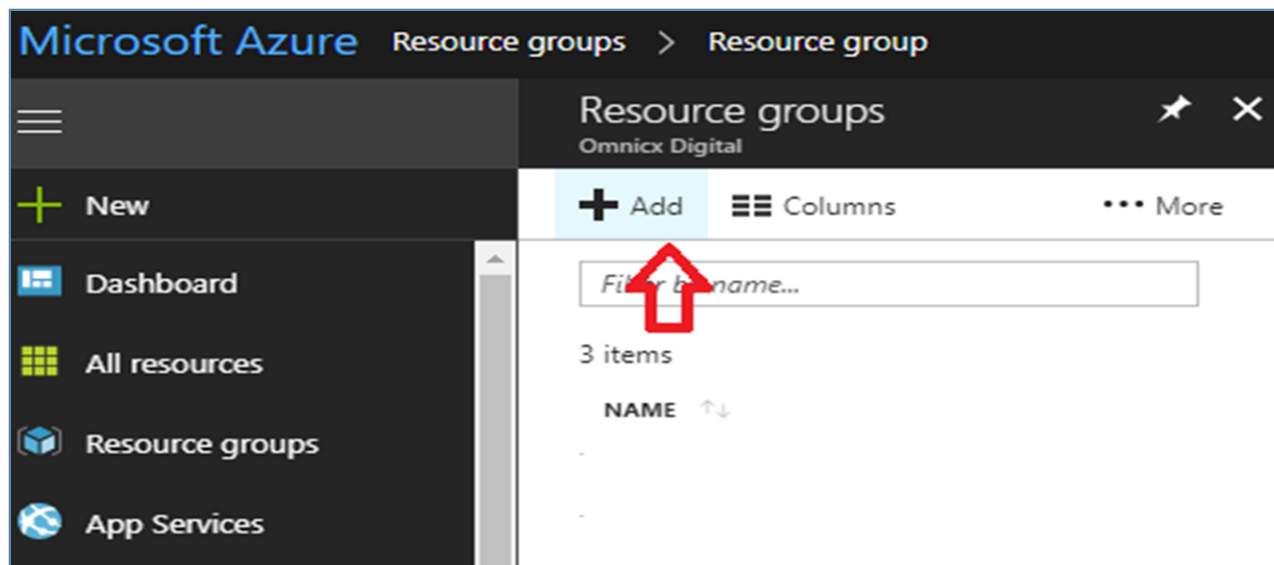
Note- Reseller must have an Azure account and with an active subscription. If they do not have an account please contact Tech Data.

Please follow the below steps before starting the hosting and code deployment.

1. Go to the Azure portal: <https://portal.azure.com/>
2. Login to your Microsoft account. If you don't have account? Please click here to [create one!](#)
3. Click Resource Groups

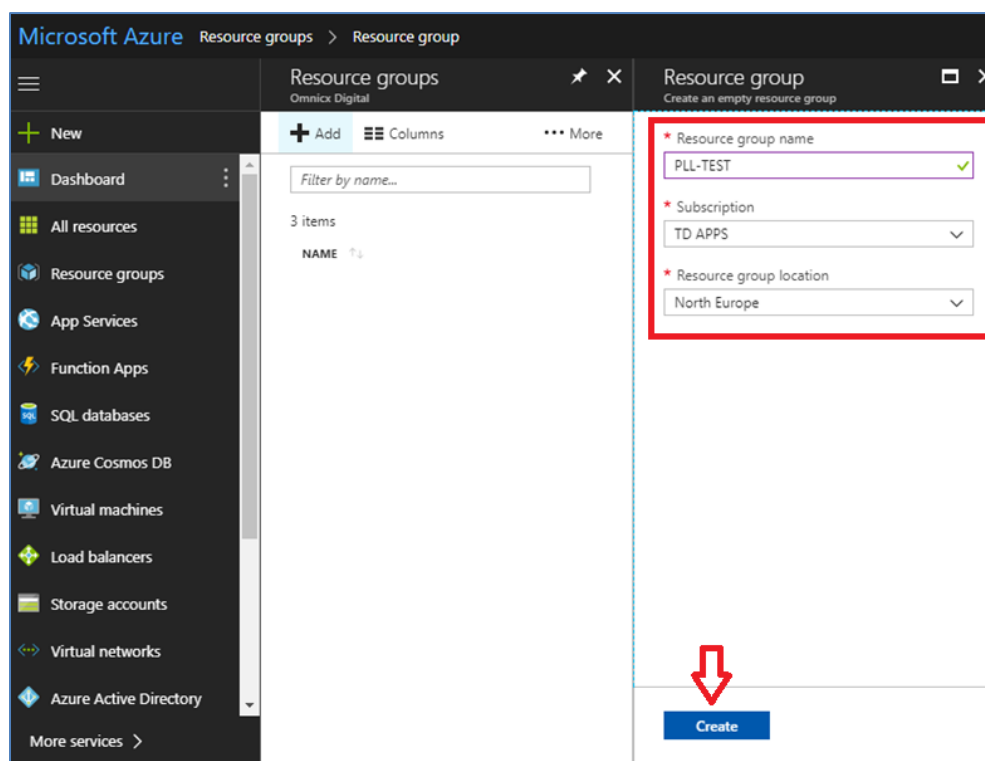


4. Click on **Add**

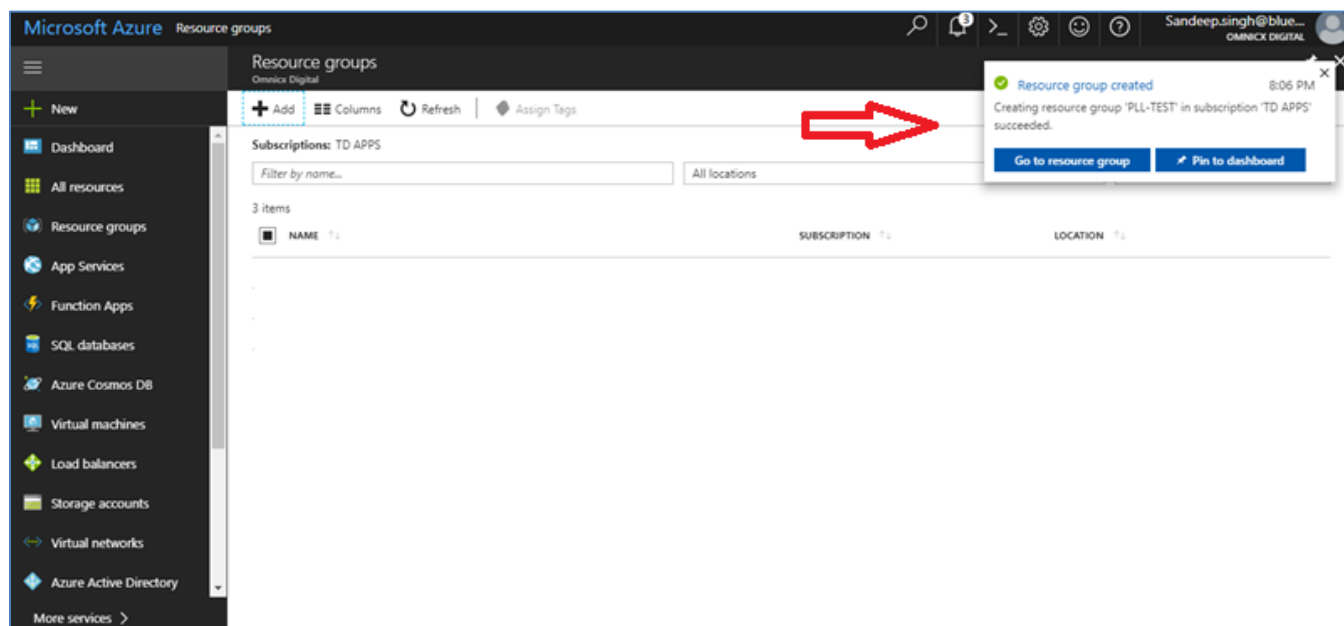


5. Please provide the input details of the resource group and then click on **Create**:

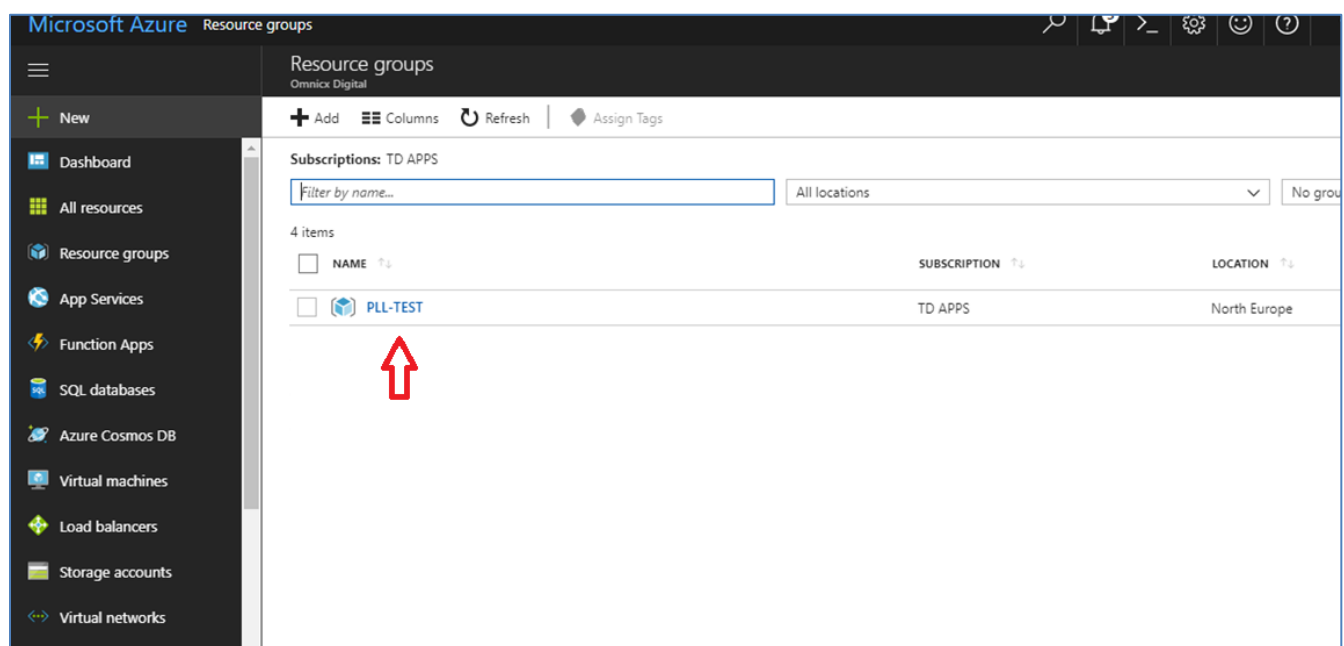
- Resource group Name (PLL-Test)
- Subscription details (TD APPS)
- Resource group location (North Europe)



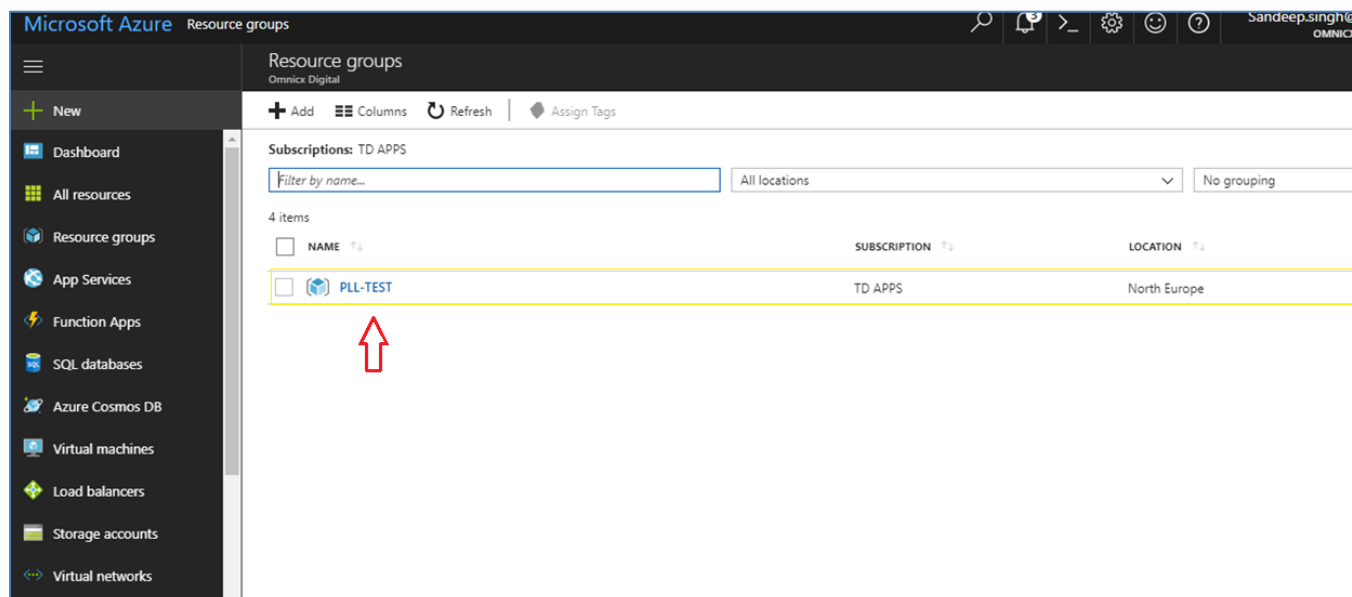
6. Now you will see this message “Resource group created”



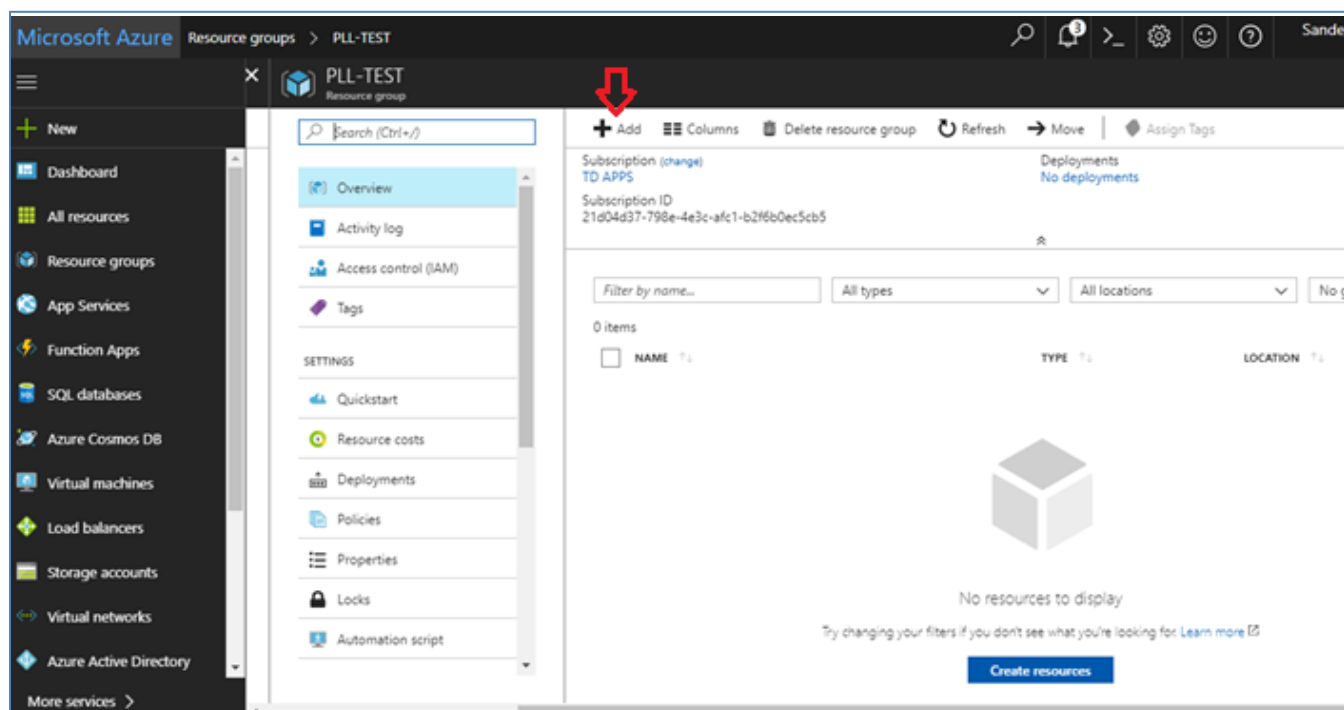
7. The resource group is then added and will appear in the list as shown below:



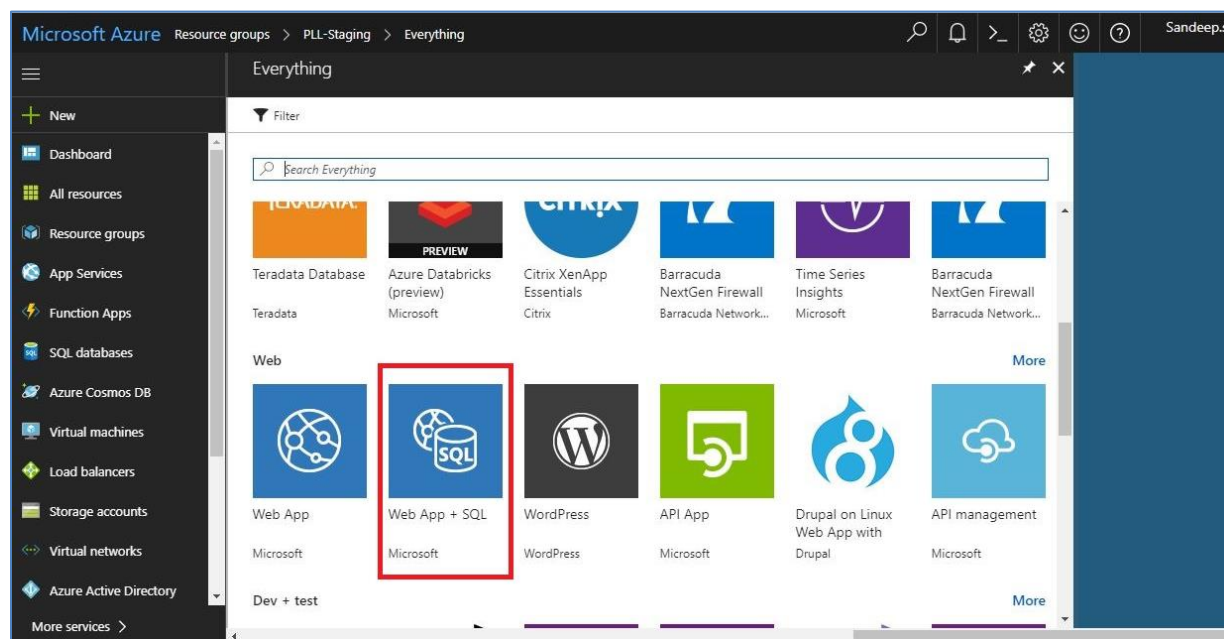
8. Now add resources to the resource group by selecting the resource group:



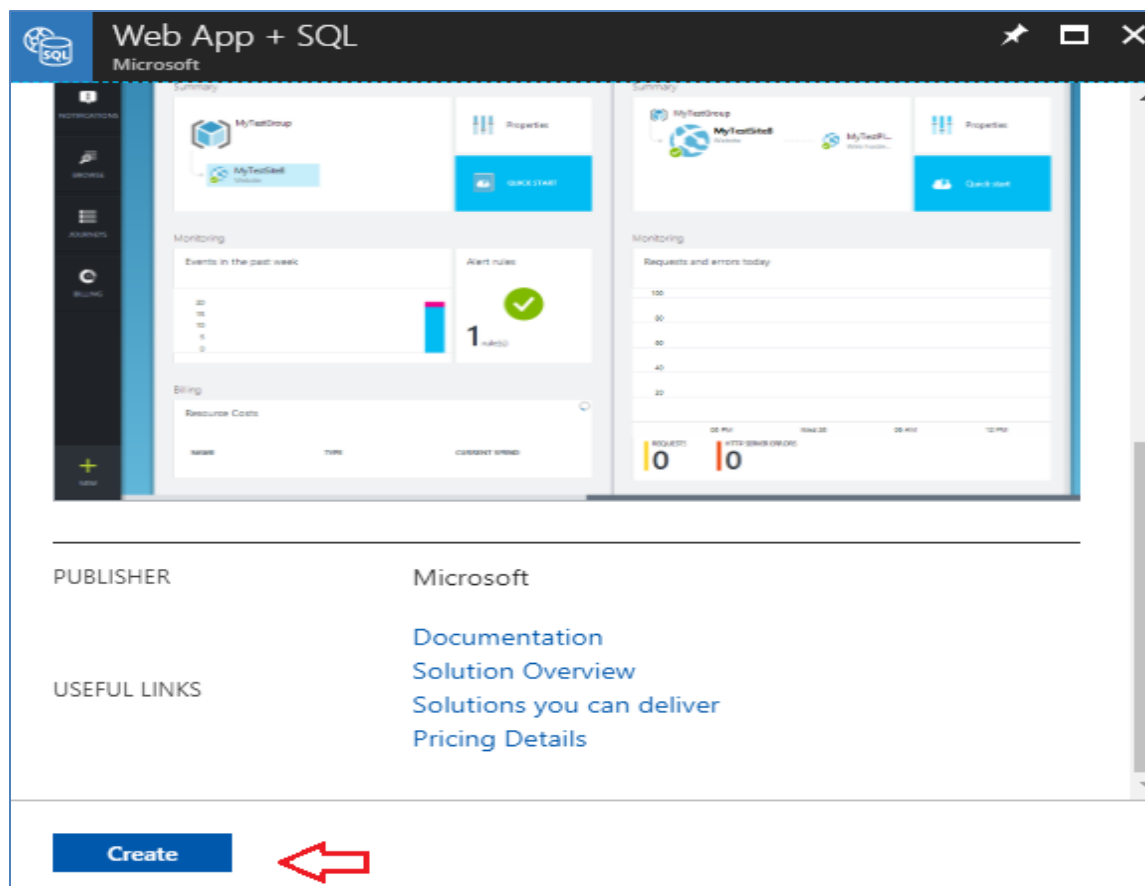
9. Click Add



10. Select Web App+ SQL

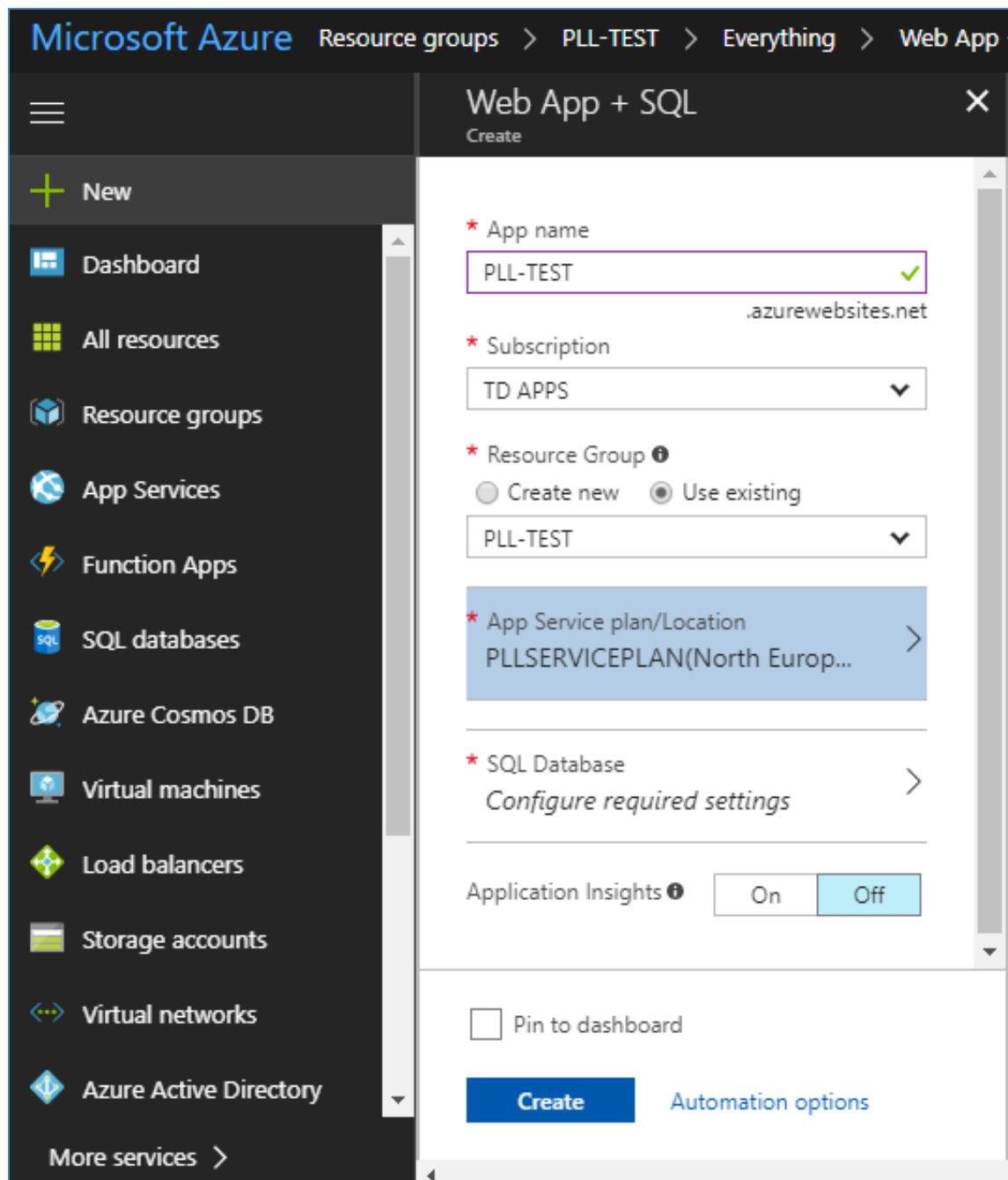


12. Click Create



13. Now provide inputs based on the required setup:

- App name
- Subscription
- Resource group
- App service plan/location
- SQL database



Microsoft Azure Resource groups > PLL-TEST > Everything > Web App + SQL

Web App + SQL
Create

+ New

- Dashboard
- All resources
- Resource groups
- App Services
- Function Apps
- SQL databases
- Azure Cosmos DB
- Virtual machines
- Load balancers
- Storage accounts
- Virtual networks
- Azure Active Directory
- More services >

* App name
PLL-TEST ✓
.azurewebsites.net

* Subscription
TD APPS ▼

* Resource Group ⓘ
☐ Create new ☒ Use existing
PLL-TEST ▼

* App Service plan/Location
PLLSERVICEPLAN(North Europ... >

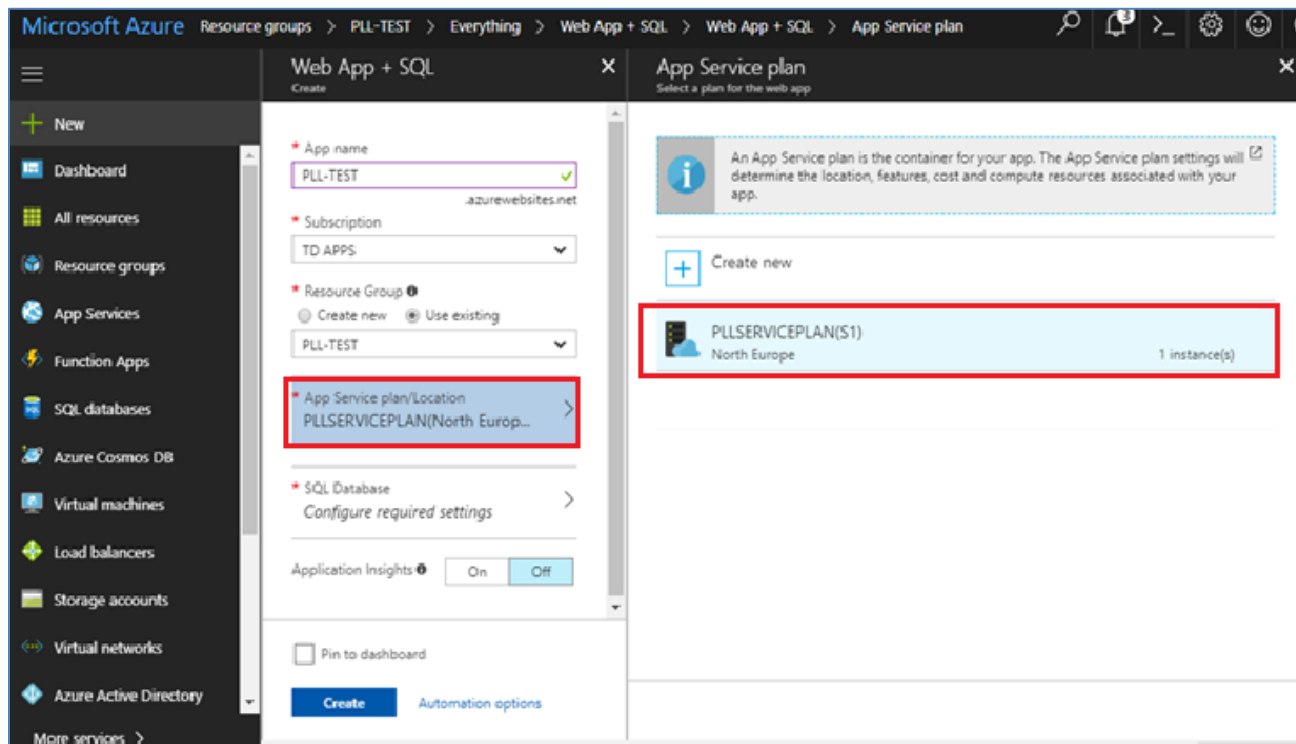
* SQL Database
Configure required settings >

Application Insights ⓘ

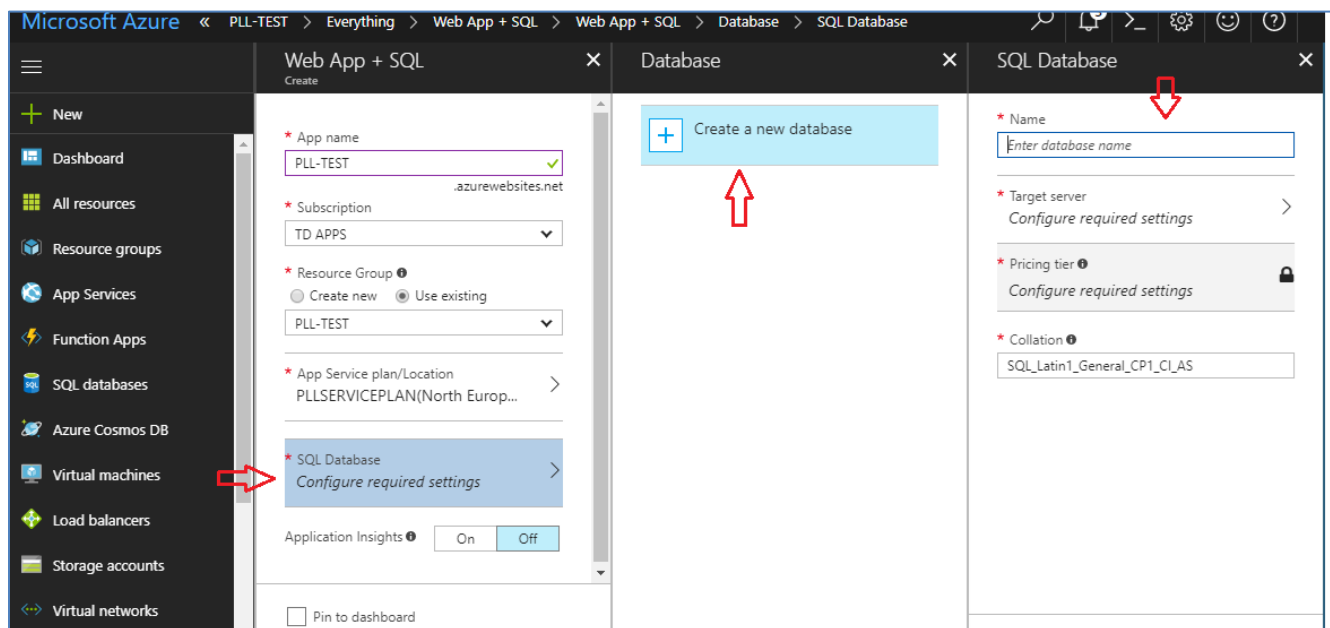
☐ Pin to dashboard

Create Automation options

- **App name** is the place where you customize the app URL. You choose the sub-domain that will be a part of the URL for both the reseller and customer. For example, <https://PLL-Test.azurewebsites.net>
- **Subscription:** Is the Azure subscription name that belongs to your Azure account
- **Resource group:** These are the logical containers for a collection of resources like the ones we have already created. For example the **PLL-Test** resource group.
- **Application service plan:** Please select S1 Basic Plan for the minimum requirements of the app. Take into consideration your geographic region. Like in our case we have used **North Europe Region**.

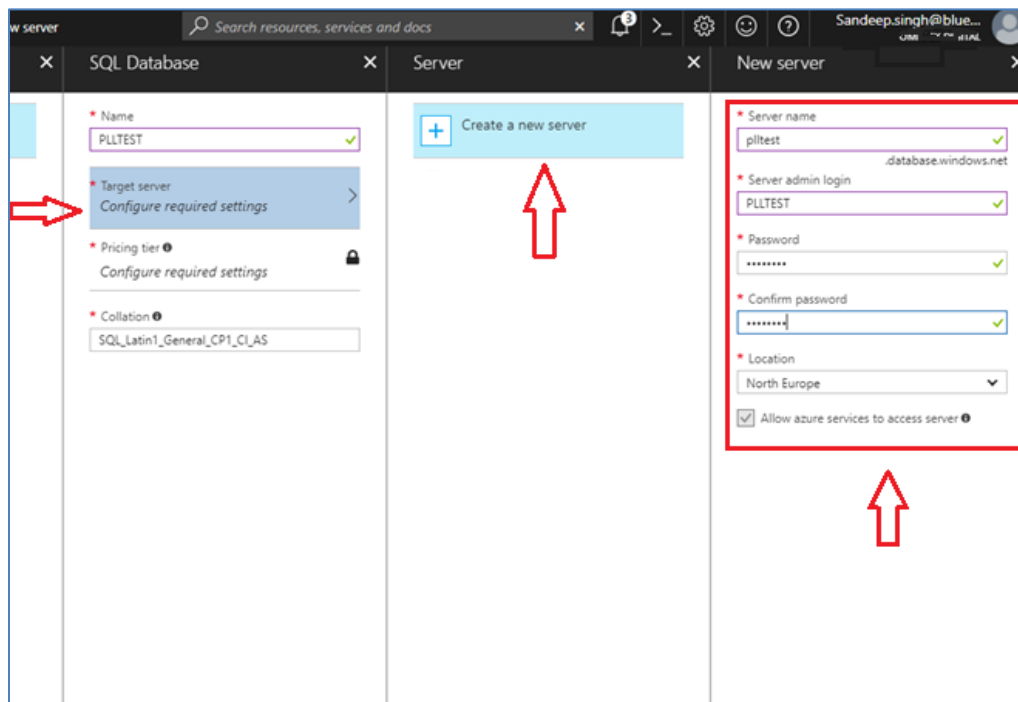


14. Click on **SQL Database** and then click on **create a new database**

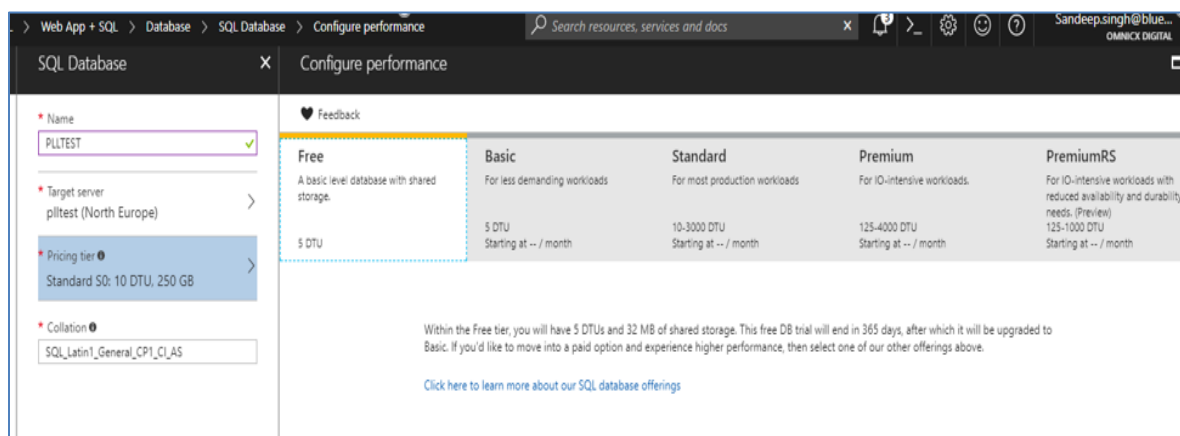


15. Provide the inputs based on the required setup:

- SQL database name
- Target server
- Pricing tier
- Collation (chosen by the system by default)



16. Pricing tier allows you to scale up or down the storage, by default it sets it to standard SO plan which can be adjusted based on your requirements:

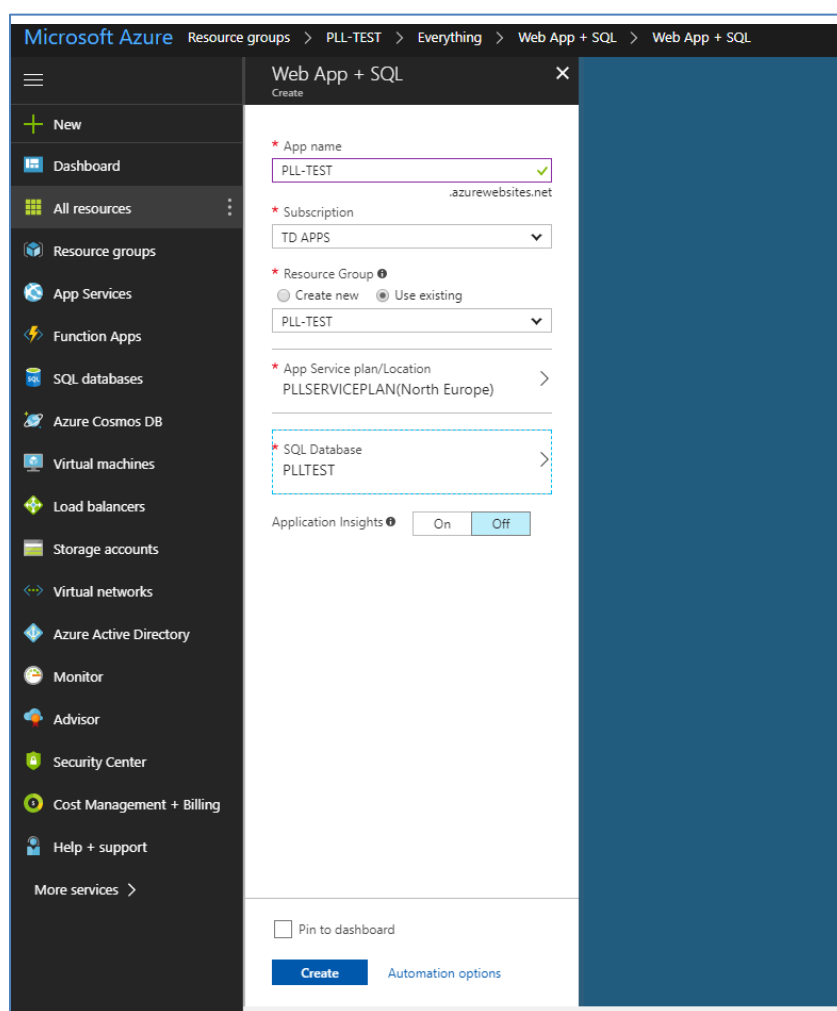


Free	Basic	Standard	Premium	PremiumRS
A basic level database with shared storage.	For less demanding workloads	For most production workloads	For IO-intensive workloads.	For IO-intensive workloads with reduced availability and durability needs. (Preview)
5 DTU	5 DTU Starting at -- / month	10-3000 DTU Starting at -- / month	125-4000 DTU Starting at -- / month	125-1000 DTU Starting at -- / month

Within the Free tier, you will have 5 DTUs and 32 MB of shared storage. This free DB trial will end in 365 days, after which it will be upgraded to Basic. If you'd like to move into a paid option and experience higher performance, then select one of our other offerings above.

[Click here to learn more about our SQL database offerings](#)

17. Now click on **create** to add all resources to the resource group:



Microsoft Azure Resource groups > PLL-TEST > Everything > Web App + SQL > Web App + SQL

Web App + SQL
Create

* App name
PLL-TEST ✓
.azurewebsites.net

* Subscription
TD APPS

* Resource Group
☐ Create new ☒ Use existing
PLL-TEST

* App Service plan/Location
PLLSERVICEPLAN(North Europe)

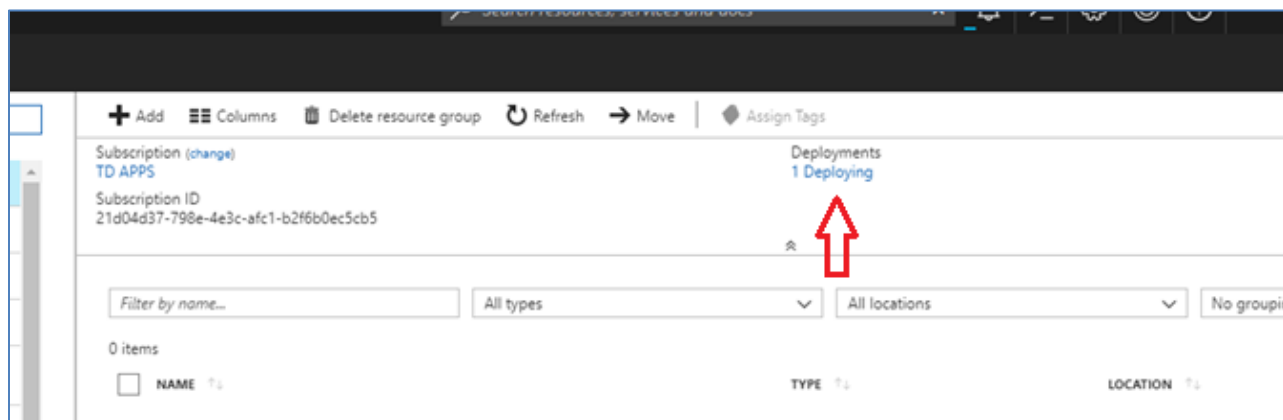
* SQL Database
PLLTEST

Application Insights ☐ On ☒ Off

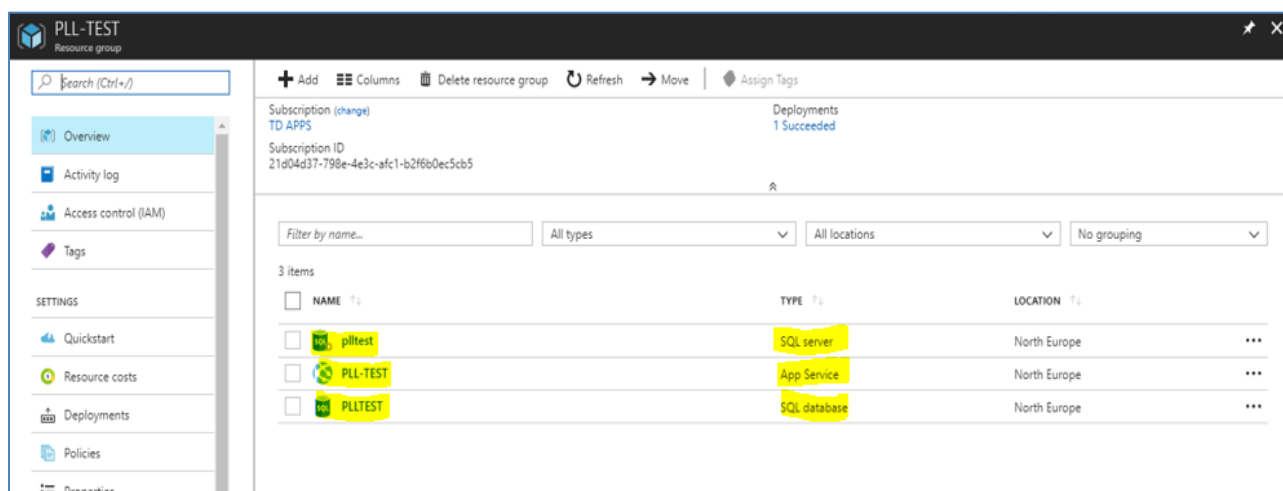
☐ Pin to dashboard

Create Automation options

Note- It will take a while to complete the deployment. A message will be displayed on the top right 'Deploying.'



18. Post deployment you will see all the resources in the list:



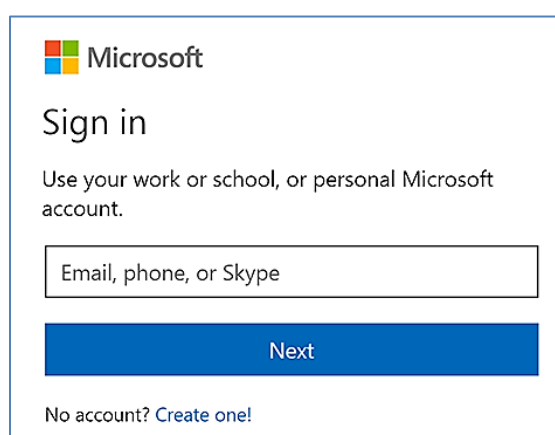
Microsoft Account Creation & Registration

1. Please [create](#) a Microsoft account. If you already have one, skip to step 2.

What is a Microsoft account?

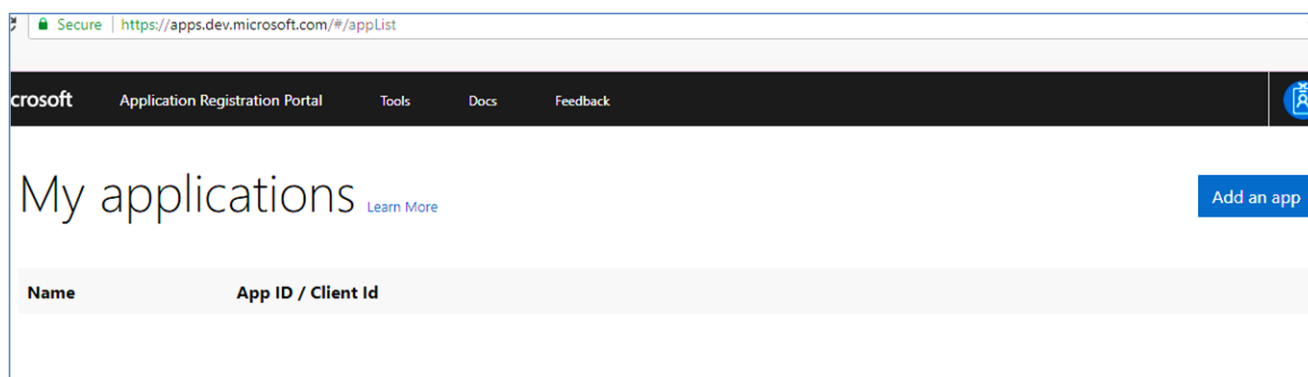
A Microsoft account is what you use to access many Microsoft devices and services. It's the account that you use to sign in to Skype, Outlook.com, OneDrive, Windows Phone, and Xbox LIVE – and it means your files, photos, contacts and settings can follow you securely to any device.

2. Navigate to <https://apps.dev.microsoft.com> and sign in



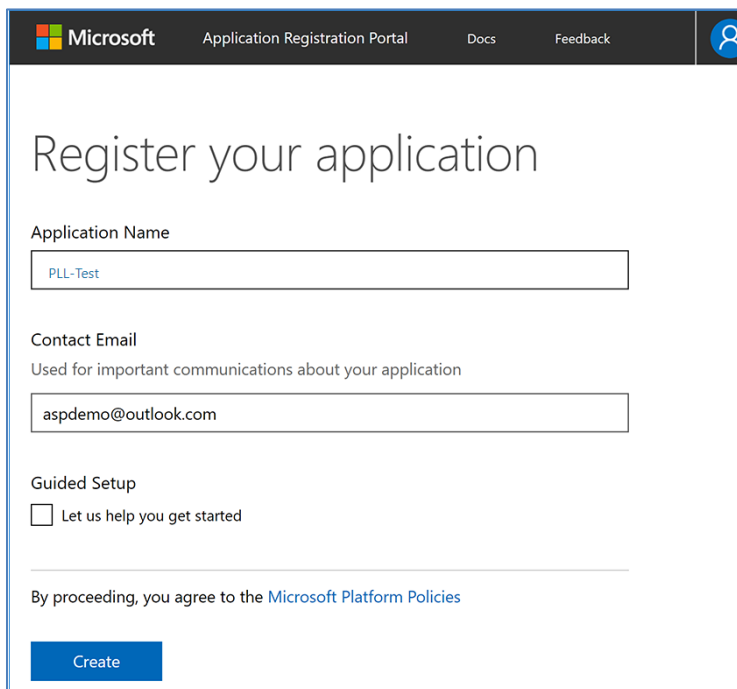
The image shows a Microsoft sign-in page. At the top is the Microsoft logo. Below it is the text "Sign in". Underneath that is the instruction "Use your work or school, or personal Microsoft account." There is a text input field with the placeholder text "Email, phone, or Skype". Below the input field is a blue button labeled "Next". At the bottom of the form is the text "No account? [Create one!](#)".

3. After signing in you are redirected to the **My applications** page:



The image shows a web browser window displaying the "My applications" page on the Microsoft Application Registration Portal. The address bar shows the URL "https://apps.dev.microsoft.com/#/appList". The page has a dark header with the Microsoft logo, "Application Registration Portal", and links for "Tools", "Docs", and "Feedback". The main content area has the heading "My applications" with a "Learn More" link. On the right side of the heading is a blue button labeled "Add an app". Below the heading is a table with two columns: "Name" and "App ID / Client Id". The table is currently empty.

- Click **Add an app** in the upper right corner and enter your **Application Name** and **Contact Email**:



The screenshot shows the 'Register your application' page in the Microsoft Application Registration Portal. The page has a dark header with the Microsoft logo, 'Application Registration Portal', 'Docs', 'Feedback', and a user profile icon. The main content area is white and contains the following fields and options:

- Application Name:** A text input field containing 'PLL-Test'.
- Contact Email:** A text input field containing 'aspdemo@outlook.com'. Below the field is the text 'Used for important communications about your application'.
- Guided Setup:** A checkbox labeled 'Let us help you get started' which is currently unchecked.
- Agreement:** A line of text stating 'By proceeding, you agree to the [Microsoft Platform Policies](#)'.
- Create Button:** A blue button labeled 'Create' at the bottom left.

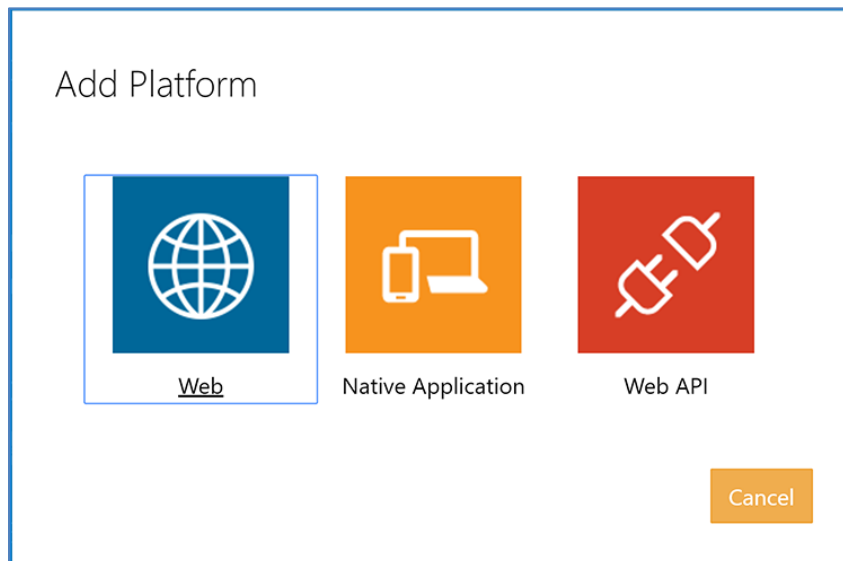
- Click **Create** to continue to the **Registration** page. Provide a **Name** and note the value of the **Application Id**, which you use as `ClientId` later in the webconfig:



The screenshot shows the 'SocialLogins Registration' page in the Microsoft Application Registration Portal. The page has a dark header with the Microsoft logo, 'Application Registration Portal', 'Docs', 'Feedback', and a user profile icon. The main content area is white and contains the following sections and fields:

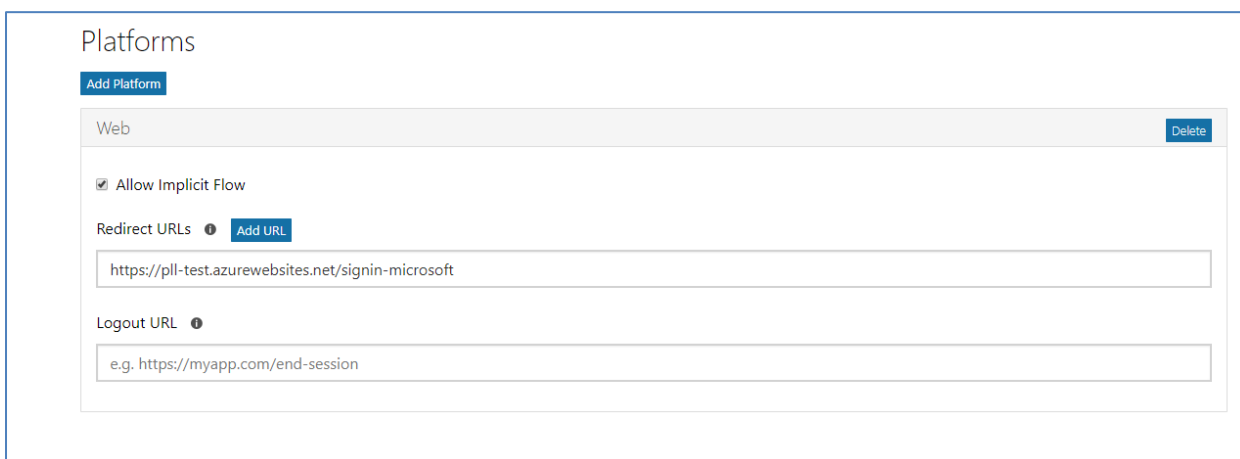
- My applications / SocialLogins:** A breadcrumb trail at the top.
- SocialLogins Registration:** The main heading.
- Help Link:** A link that says 'Click here for help integrating your application with Microsoft.'
- Properties:** A section containing:
 - Name:** A text input field containing 'PLL-Test'.
 - Application Id:** A text field displaying the value '4d519acd-6a8c-4df4-a246-fe5ea741c9db'.
- Application Secrets:** A section with three buttons: 'Generate New Password', 'Generate New Key Pair', and 'Upload Public Key'.
- Platforms:** A section with one button: 'Add Platform'.

6. Tap **Add Platform** in the **Platforms** section and select the **Web** platform:



7. Update the Redirect URL:

Code on the Azure portal: In the new **Web** platform section, enter your development URL with `/signin-microsoft` added, into the **Redirect URLs** field (for example: <https://pll-test.azurewebsites.net/signin-microsoft>). The Microsoft authentication scheme configured later in this tutorial will automatically handle requests at `/signin-microsoft` route to implement the OAuth flow:



Platforms

[Add Platform](#)

Platform	Actions
Web	Delete

☒ Allow Implicit Flow

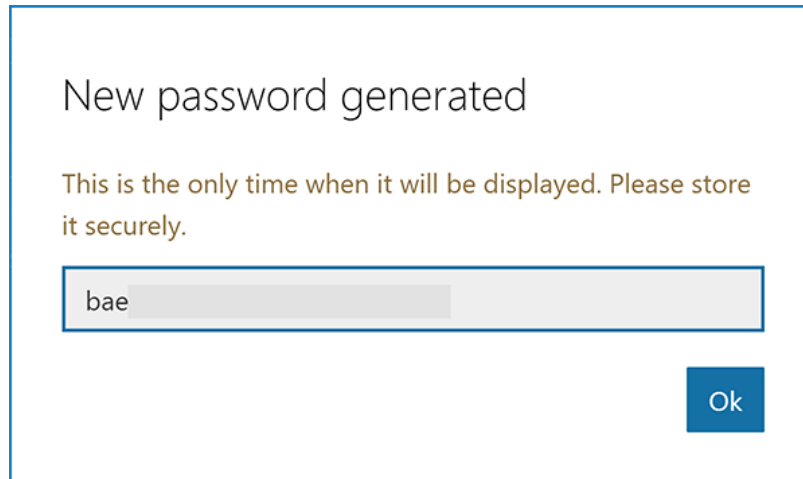
Redirect URLs [Add URL](#)

Logout URL [Add URL](#)

8. Click **Add URL** to ensure the URL was added.
9. Fill out any other application settings if necessary and tap **Save** at the bottom of the page to save changes to app configuration.

Note- Store the Microsoft application ID and Password

10. **Application Id** is displayed on the **Registration** page.
11. Tap **Generate New Password** in the **Application Secrets** section. This displays a box where you can copy the application password:



Link sensitive settings like Microsoft **Application ID** and **Password** to your application configuration.

Congratulations now your application is registered!

Code Deployment

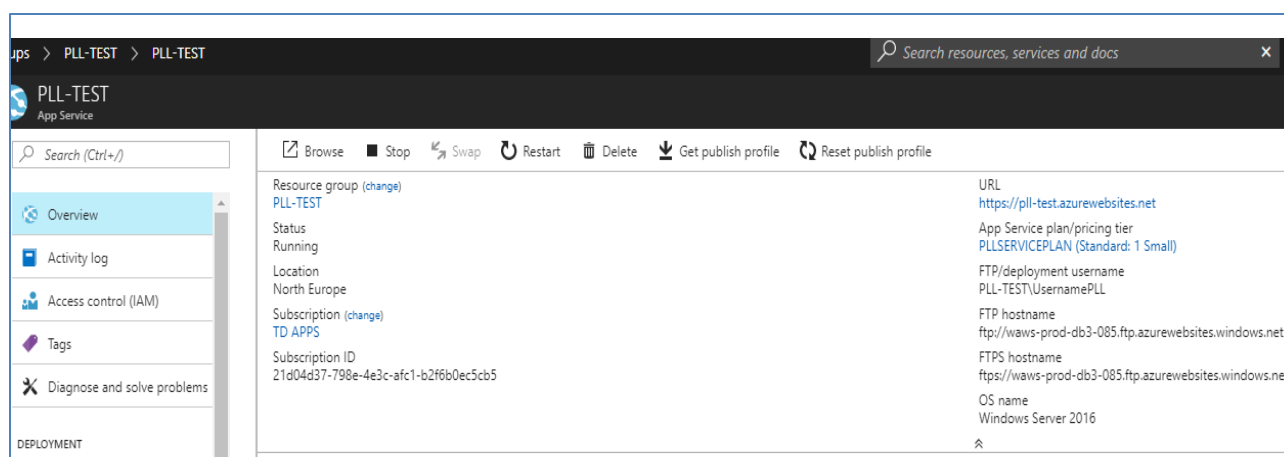
You have already been provided with the complied/raw code that you can use for the deployment. Below are the prerequisites to accomplish this task.

1. Update the web-config file.
2. Upload the code on the FTP
3. Run DB scripts.

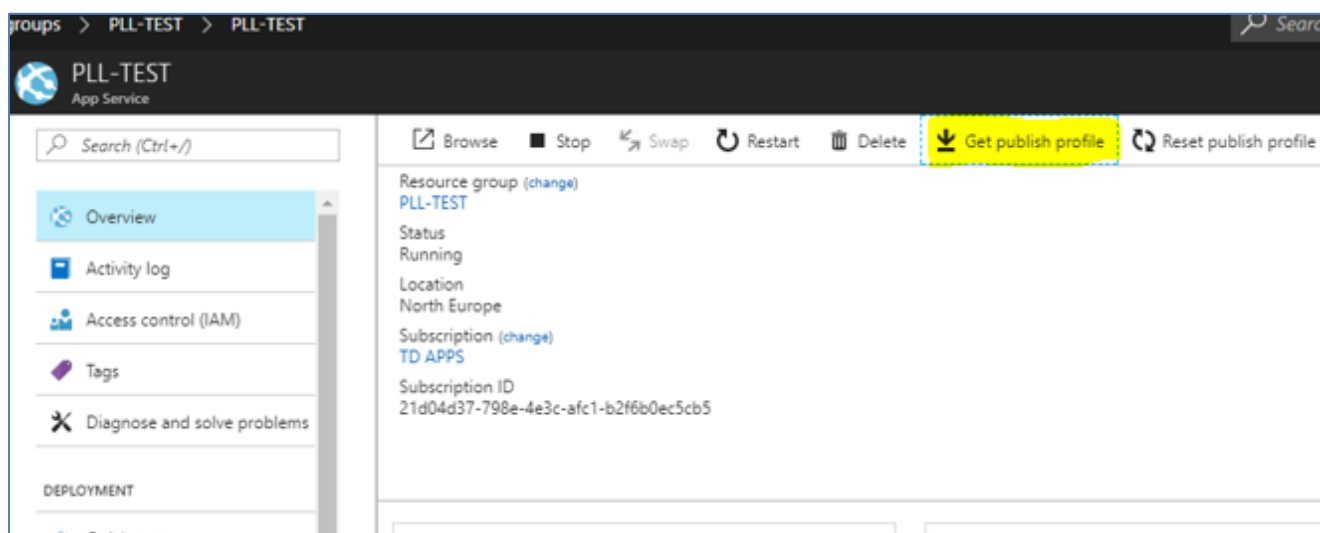
Update the Web-Config file.

Before we start updating the Web Config, publish the profile.

1. Open your Azure portal. Click on **Resource Groups**. Open your resource. Click on the PLL-TEST App Service



2. Click on **Get Publish Profile**. This will download the publish profile of the application that contains the secure information of the application.



NOTE: The Publish Profile format can be difficult to read and/or understand. We recommend you open it with [xmlbeautifier](#). Copy and paste the data into the box and then click on the 'beautify now' button.

XML «Beautifier»

paste or upload an ugly looking XML document and make it suddenly looking beautiful

NOTE: do not submit documents with sensitive information - this is a server side tool.

beautify now!

[formatting options](#) | [upload xml](#) | [xml tree view](#) | [xsd from xml](#) | **NEW!** [json beautifier](#)

After beautification you will get much better format that is readable:

```
<publishData>

  <publishProfile profileName="pll-test - Web Deploy" publishMethod="MSDeploy" publishUrl="pll-
test.scm.azurewebsites.net:443" msdeploySite="pll-test" userName="$pll-test"
userPWD="7moFdGByiJkbiREnBdbmqNjlbiylBtK7vHT3vd2H4ao1ymJvQGLWoJQuaZa7"
destinationAppUrl="http://pll-test.azurewebsites.net" SQLServerDBConnectionString="Data Source=tcp:pll-
karan-server.database.windows.net,1433;Initial Catalog=pll-karanDB;User Id=karan@pll-karan-
server.database.windows.net;Password=Welcome1;" mySQLDBConnectionString=""
hostingProviderForumLink="" controlPanelLink="http://windows.azure.com" webSystem="WebSites"
targetDatabaseEngineType="sqlazuredatabase" targetServerVersion="Version100">

    <databases>

      <add name="defaultConnection" connectionString="Data Source=tcp:pll-karan-
server.database.windows.net,1433;Initial Catalog=pll-karanDB;User Id=karan@pll-karan-
server.database.windows.net;Password=W**c*m*1;" providerName="System.Data.SqlClient" type="Sql"
targetDatabaseEngineType="sqlazuredatabase" targetServerVersion="Version100" />

    </databases>

  </publishProfile>

  <publishProfile profileName="pll-test - FTP" publishMethod="FTP" publishUrl="ftp://waws-prod-db3-
o85.ftp.azurewebsites.windows.net/site/wwwroot" ftpPassiveMode="True" userName="pll-test" $pll-test"
userPWD="7moFdGByiJkbiREnBdbmqNjlbiylBtK7vTT3vd2H7ao1ymJvQGLWoJQuaZa7"
destinationAppUrl="http://pll-test.azurewebsites.net" SQLServerDBConnectionString="Data Source=tcp:pll-
karan-server.database.windows.net,1433;Initial Catalog=pll-karanDB;User Id=karan@pll-karan-
server.database.windows.net;Password=Welcome1;" mySQLDBConnectionString=""
hostingProviderForumLink="" controlPanelLink="http://windows.azure.com" webSystem="WebSites"
targetDatabaseEngineType="sqlazuredatabase" targetServerVersion="Version100">

    <databases>

      <add name="defaultConnection" connectionString="Data Source=tcp:pll-karan-
server.database.windows.net,1433;Initial Catalog=pll-karanDB;User Id=karan@pll-karan-
server.database.windows.net;Password=Welcome1;" providerName="System.Data.SqlClient" type="Sql"
targetDatabaseEngineType="sqlazuredatabase" targetServerVersion="Version100" />

    </databases>

  </publishProfile>

</publishData>
```

3. Please take note of the following values from the above file- highlighted in yellow.
 - a. From following tag<databases>
 - Data Source= tcp:pll-karan-server.database.windows.net.
 - User ID = karan@pll-karan-server.database.windows.net
 - Password = Welcome1
 - Initial Catalog=pll-karanDB
 - b. From publishMethod in tag<publishProfile> -
 - publishUrl=ftp://waws-prod-db3-o85.ftp.azurewebsites.windows.net/site/wwwroot
 - userName="pll-test\pll-test"
 - userPWD="7moFdGBYiJkbiREnBdbmqNjlbiylBtK7vHT3vdzH4ao1ymJvQGLW**
*****"

Note: The above are examples only and the information in your data may differ

4. Open Web-Config File, available in the code given to you. Right click on the file and choose edit with notepad/notepad++
5. Look for tag named <ConnectionStrings>:

```
<connectionStrings>
  <add name="PLLConnection" connectionString="Server=tcp:pll-karan-server.database.windows.net;Initial Catalog=pll-karanDB;Persist Security Info=False;User ID=karan@pll-karan-server.database.windows.net;Password=Welcome1;MultipleActiveResultSets=False;Encrypt=True;TrustServerCertificate=False;Connection Timeout=30;" providerName="System.Data.SqlClient" />
  <add name="PrivateLabelLiteDataEntities" connectionString="metadata=res://*/DataEntities.PLLDataModel.csdl|res://*/DataEntities.PLLDataModel.ssdl|res://*/DataEntities.PLLDataModel.msl;provider=System.Data.SqlClient;provider connection string="data source=tcp:pll-karan-server.database.windows.net;initial catalog=pll-karanDB;User ID=karan@pll-karan-server.database.windows.net;Password=Welcome1;MultipleActiveResultSets=False;Encrypt=True;TrustServerCertificate=False;Connection Timeout=30;"" providerName="System.Data.EntityClient" />
</connectionStrings>
```

In connection string you have to update the following things from the points you noted down in 3(a) step.

- a. Server = It is a Data Source, so copy the data source here. i.e. "tcp:pll-karan-server.database.windows.net"
- b. Initial Catalog = pll-karanDB
- c. User ID=karan@pll-karan-server.database.windows.net
- d. Password= W**c*m*1

NOTE: You have to update these same 4 things again in connection string itself. Look carefully these 4 points (Data source, initial catalog, userid, password) come again in the next 2 lines.

6. Check <appSettings> and update the fields as per requirement:

```
<appSettings>
  <add key="webpages:Version" value="3.0.0.0" />
  <add key="webpages:Enabled" value="false" />
  <add key="ClientValidationEnabled" value="true" />
  <add key="UnobtrusiveJavaScriptEnabled" value="true" />
  <add key="ClientId" value=" " />
  <add key="RedirectUri" value="http://localhost:56619/" />
  <add key="Tenant" value="common" />
  <add key="Authority" value="https://login.microsoftonline.com/{01/v2.0" />
  <!-- Microsoft Account Integration Setting-->
  <add key="MicrosoftClientId" value=" " />
  <add key="MicrosoftClientSecret" value=" " />
  <add key="AllowedResellers" value=" " />
  <add key="NotificationEmails" value=" " />
  <add key="NotificationEmailFrom" value=" " />
  <add key="PageSize" value="20" />
  <add key="MSVendorCatalogIntervalMin" value="180" />
  <add key="EnableSecureUrlScheme" value="true" />
  <add key="UpdateVendorCatalogIntervalHours" value="10" />
</appSettings>
```

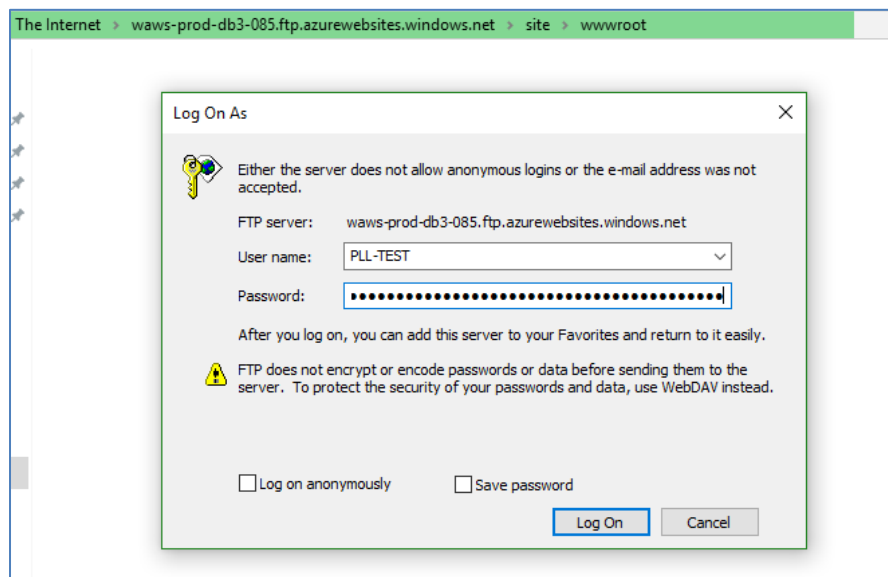
- **"MicrosoftClientId"** is the AppId or Client Id got after registering your application on the Microsoft site.
- **"MicrosoftClientSecret"** is the password given after registering your application on the Microsoft site.
- **"AllowedResellers"** are the users with admin rights. Add them according to your requirements.
- **"NotificationEmails"** are the emails which will be triggered from the app if there are any actions performed on the SDK app.
- **"NotificationEmailFrom"** is the email from which it will be triggered.

7. Mail Setting can also be changed if required.

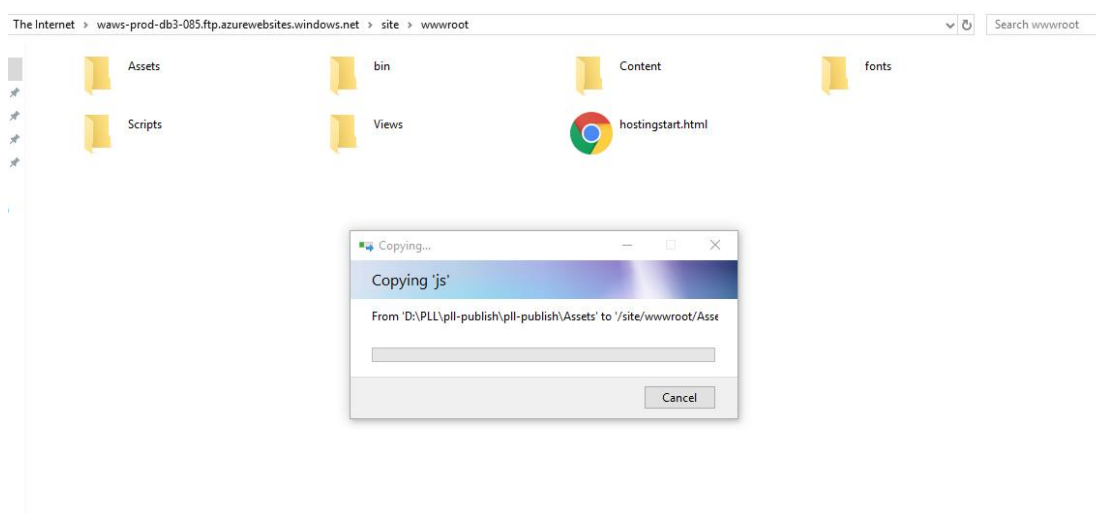
```
</customErrors>
</system.web>
<system.net>
  <mailSettings>
    <smtp>
      <network host="outlook.office365.com" port="587" userName=" " password=" " />
    </smtp>
  </mailSettings>
</system.net>
<system.webServer>
```

Upload code on FTP

- Look for the FTP details you noted down in step 3(b).
 - publishUrl=<ftp://waws-prod-db3-o85.ftp.azurewebsites.windows.net/site/wwwroot>
 - userName="pll-test\pll-test"
 - userPWD="7moFdGBYiJkbiREnBdbmqNjlbIyIBtK7vHT3vdzH4ao1ymJvQGLW***
*****"



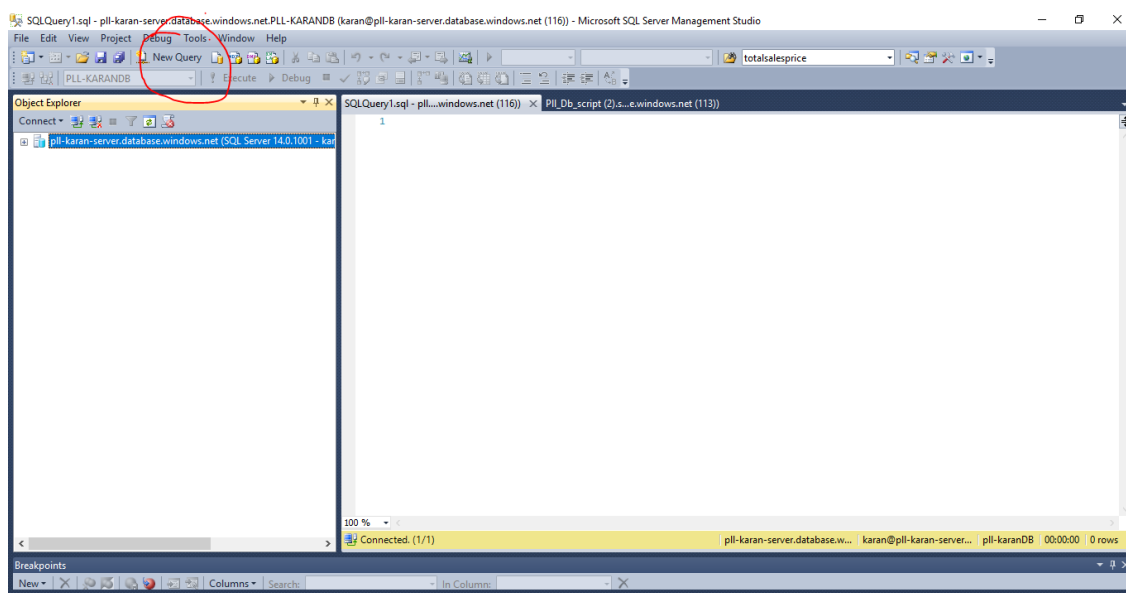
- Copy and paste the code provided to you on FTP, this will take a few minutes. Do not cancel the copying or restart/sleep/shutdown your system while copying.



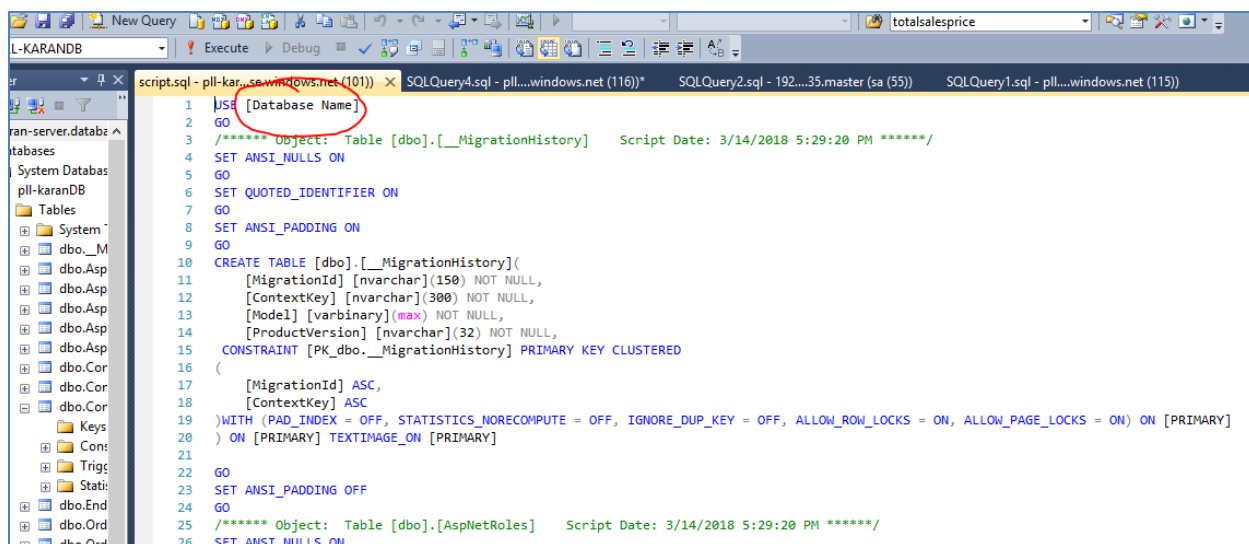
Run DB scripts.

Please open MS-SQL server on your local system.

1. Enter the credentials:
 - a. Server name = It is a Data Source, so copy the data source here. i.e. "tcp:pll-karan-server.database.windows.net"
 - b. Login - It is the User ID => karan@pll-karan-server.database.windows.net
 - c. Password= W**c*m*1
2. How to Run the Script:
 - a. Now click on "New Query".



- b. Change the database from Master to your database.
 - Please open the Scripts folder shared with you and open the script "Database_Schema" in notepad. Copy and paste that into your new query section.
 - c. Change the database name in script file. Enter your database name in the very first line of script.



- d. Press F5.
- e. Again click on 'New Query'
- f. Open the Scripts folder and copy the script "Config_data" in your new query section.
- g. Repeat step 'c'.

3. Update "Config_data" Script with required data to run the application.

You will see dummy values as 'abcd', 'URL', 'Coming Soon' etc. in the config data however you must be provided with the following values beforehand by Tech Data. Once 'client_secret'

- 'client_id'
- 'SOIN'
- 'ResellerId'
- 'AllowedResellers'

Once recieved edit the below query before executing it.

```
USE [DatabaseName]
GO
SET IDENTITY_INSERT [dbo].[Configs] ON

INSERT [dbo].[Configs] ([Id], [Key], [Value], [Type]) VALUES (7, N'client_secret', N'abcd', N'Auth')
INSERT [dbo].[Configs] ([Id], [Key], [Value], [Type]) VALUES (8, N'TokenOAuth', N'url', N'Auth')
INSERT [dbo].[Configs] ([Id], [Key], [Value], [Type]) VALUES (9, N'client_id', N'abcd', N'Auth')
INSERT [dbo].[Configs] ([Id], [Key], [Value], [Type]) VALUES (10, N'SOIN', N'abcd', N'Auth')
INSERT [dbo].[Configs] ([Id], [Key], [Value], [Type]) VALUES (11, N'grant_type', N'client_credentials', N'Auth')
INSERT [dbo].[Configs] ([Id], [Key], [Value], [Type]) VALUES (12, N'CustomerSearchURL', N'url', N'API URL')
INSERT [dbo].[Configs] ([Id], [Key], [Value], [Type]) VALUES (13, N'OrderSearchUrl', N'url', N'API URL')
INSERT [dbo].[Configs] ([Id], [Key], [Value], [Type]) VALUES (14, N'OrderModifyUrl', N'url', N'API URL')
INSERT [dbo].[Configs] ([Id], [Key], [Value], [Type]) VALUES (15, N'OrderDetailUrl', N'url', N'API URL')
INSERT [dbo].[Configs] ([Id], [Key], [Value], [Type]) VALUES (16, N'ModifyAddOnUrl', N'url', N'API URL')
INSERT [dbo].[Configs] ([Id], [Key], [Value], [Type]) VALUES (17, N'ResellerId', N'abcd', N'Reseller Info')
INSERT [dbo].[Configs] ([Id], [Key], [Value], [Type]) VALUES (18, N'VendorListUrl', N'url', N'API URL')
INSERT [dbo].[Configs] ([Id], [Key], [Value], [Type]) VALUES (19, N'VendorCatalogSearchUrl', N'url', N'API URL')
INSERT [dbo].[Configs] ([Id], [Key], [Value], [Type]) VALUES (21, N'ResellerName', N'Tech Data 2', N'Reseller Info')
INSERT [dbo].[Configs] ([Id], [Key], [Value], [Type]) VALUES (22, N'ApplicationName', N'Pl1 Test App', N'App Info')
INSERT [dbo].[Configs] ([Id], [Key], [Value], [Type]) VALUES (23, N'AllowedResellers', N'abcd', N'Reseller Info')
SET IDENTITY_INSERT [dbo].[Configs] OFF

GO
SET IDENTITY_INSERT [dbo].[SiteContent] ON

INSERT [dbo].[SiteContent] ([Id], [Key], [Value]) VALUES (1, N'TermsAndConditions', N'Coming Soon.')
SET IDENTITY_INSERT [dbo].[SiteContent] OFF
```

- Please enter valid URL (dummy value written as 'url')

```
USE [DatabaseName]
GO
SET IDENTITY_INSERT [dbo].[Configs] ON

INSERT [dbo].[Configs] ([Id], [Key], [Value], [Type]) VALUES (7, N'client_secret', N'abcd', N'Auth')
INSERT [dbo].[Configs] ([Id], [Key], [Value], [Type]) VALUES (8, N'TokenOAuth', N'url', N'Auth')
INSERT [dbo].[Configs] ([Id], [Key], [Value], [Type]) VALUES (9, N'client_id', N'abcd', N'Auth')
INSERT [dbo].[Configs] ([Id], [Key], [Value], [Type]) VALUES (10, N'SOIN', N'abcd', N'Auth')
INSERT [dbo].[Configs] ([Id], [Key], [Value], [Type]) VALUES (11, N'grant_type', N'client_credentials', N'Auth')
INSERT [dbo].[Configs] ([Id], [Key], [Value], [Type]) VALUES (12, N'CustomerSearchURL', N'url', N'API URL')
INSERT [dbo].[Configs] ([Id], [Key], [Value], [Type]) VALUES (13, N'OrderSearchUrl', N'url', N'API URL')
INSERT [dbo].[Configs] ([Id], [Key], [Value], [Type]) VALUES (14, N'OrderModifyUrl', N'url', N'API URL')
INSERT [dbo].[Configs] ([Id], [Key], [Value], [Type]) VALUES (15, N'OrderDetailUrl', N'url', N'API URL')
INSERT [dbo].[Configs] ([Id], [Key], [Value], [Type]) VALUES (16, N'ModifyAddOnUrl', N'url', N'API URL')
INSERT [dbo].[Configs] ([Id], [Key], [Value], [Type]) VALUES (17, N'ResellerId', N'abcd', N'Reseller Info')
INSERT [dbo].[Configs] ([Id], [Key], [Value], [Type]) VALUES (18, N'VendorListUrl', N'url', N'API URL')
INSERT [dbo].[Configs] ([Id], [Key], [Value], [Type]) VALUES (19, N'VendorCatalogSearchUrl', N'url', N'API URL')
INSERT [dbo].[Configs] ([Id], [Key], [Value], [Type]) VALUES (21, N'ResellerName', N'Tech Data 2', N'Reseller Info')
INSERT [dbo].[Configs] ([Id], [Key], [Value], [Type]) VALUES (22, N'ApplicationName', N'Pl1 Test App', N'App Info')
INSERT [dbo].[Configs] ([Id], [Key], [Value], [Type]) VALUES (23, N'AllowedResellers', N'abcd', N'Reseller Info')
SET IDENTITY_INSERT [dbo].[Configs] OFF

GO
SET IDENTITY_INSERT [dbo].[SiteContent] ON

INSERT [dbo].[SiteContent] ([Id], [Key], [Value]) VALUES (1, N'TermsAndConditions', N'Coming Soon.')
SET IDENTITY_INSERT [dbo].[SiteContent] OFF
```

- To add terms and conditions, please add your required terms in place of 'coming soon':

```
USE [DatabaseName]
GO
SET IDENTITY_INSERT [dbo].[Configs] ON

INSERT [dbo].[Configs] ([Id], [Key], [Value], [Type]) VALUES (7, N'client_secret', N'abcd', N'Auth')
INSERT [dbo].[Configs] ([Id], [Key], [Value], [Type]) VALUES (8, N'TokenOAuth', N'url', N'Auth')
INSERT [dbo].[Configs] ([Id], [Key], [Value], [Type]) VALUES (9, N'client_id', N'abcd', N'Auth')
INSERT [dbo].[Configs] ([Id], [Key], [Value], [Type]) VALUES (10, N'SOIN', N'abcd', N'Auth')
INSERT [dbo].[Configs] ([Id], [Key], [Value], [Type]) VALUES (11, N'grant_type', N'client_credentials', N'Auth')
INSERT [dbo].[Configs] ([Id], [Key], [Value], [Type]) VALUES (12, N'CustomerSearchURL', N'url', N'API URL')
INSERT [dbo].[Configs] ([Id], [Key], [Value], [Type]) VALUES (13, N'OrderSearchUrl', N'url', N'API URL')
INSERT [dbo].[Configs] ([Id], [Key], [Value], [Type]) VALUES (14, N'OrderModifyUrl', N'url', N'API URL')
INSERT [dbo].[Configs] ([Id], [Key], [Value], [Type]) VALUES (15, N'OrderDetailUrl', N'url', N'API URL')
INSERT [dbo].[Configs] ([Id], [Key], [Value], [Type]) VALUES (16, N'ModifyAddOnUrl', N'url', N'API URL')
INSERT [dbo].[Configs] ([Id], [Key], [Value], [Type]) VALUES (17, N'ResellerId', N'abcd', N'Reseller Info')
INSERT [dbo].[Configs] ([Id], [Key], [Value], [Type]) VALUES (18, N'VendorListUrl', N'url', N'API URL')
INSERT [dbo].[Configs] ([Id], [Key], [Value], [Type]) VALUES (19, N'VendorCatalogSearchUrl', N'url', N'API URL')
INSERT [dbo].[Configs] ([Id], [Key], [Value], [Type]) VALUES (21, N'ResellerName', N'Tech Data 2', N'Reseller Info')
INSERT [dbo].[Configs] ([Id], [Key], [Value], [Type]) VALUES (22, N'ApplicationName', N'Pl1 Test App', N'App Info')
INSERT [dbo].[Configs] ([Id], [Key], [Value], [Type]) VALUES (23, N'AllowedResellers', N'abcd', N'Reseller Info')
SET IDENTITY_INSERT [dbo].[Configs] OFF

GO
SET IDENTITY_INSERT [dbo].[SiteContent] ON

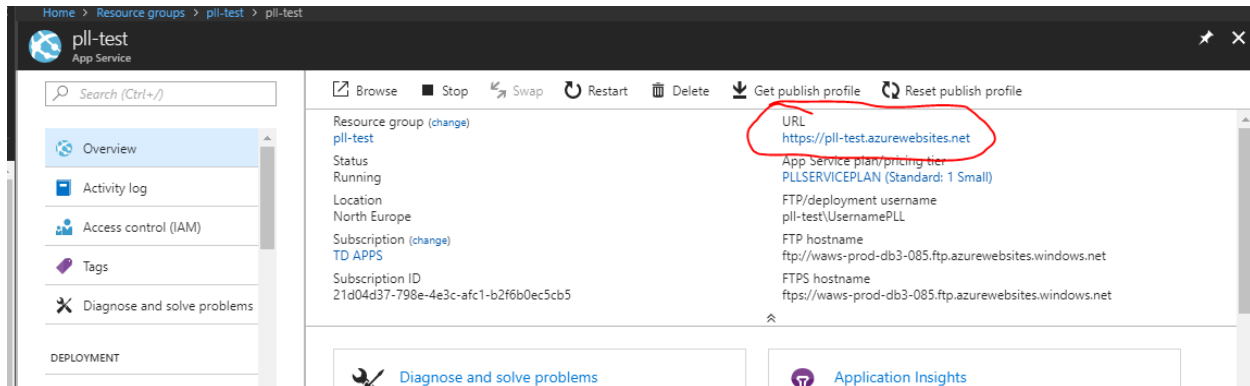
INSERT [dbo].[SiteContent] ([Id], [Key], [Value]) VALUES (1, N'TermsAndConditions', N'Coming Soon.')
SET IDENTITY_INSERT [dbo].[SiteContent] OFF
```

- 'ApplicationName' is customizable.
- Press F5.

App Configuration	
1. Reseller id	(Already Provided by Tech Data)
2. SOIN	(Provided by Tech Data)
3. Client_id	(Provided by Tech Data)
4. Client Secret	(Provided by Tech Data)
5. E-mail Server Configuration (for seat change notification)	E-mail & Password (Reseller should provide it at the time of hosting)

RUN the website.

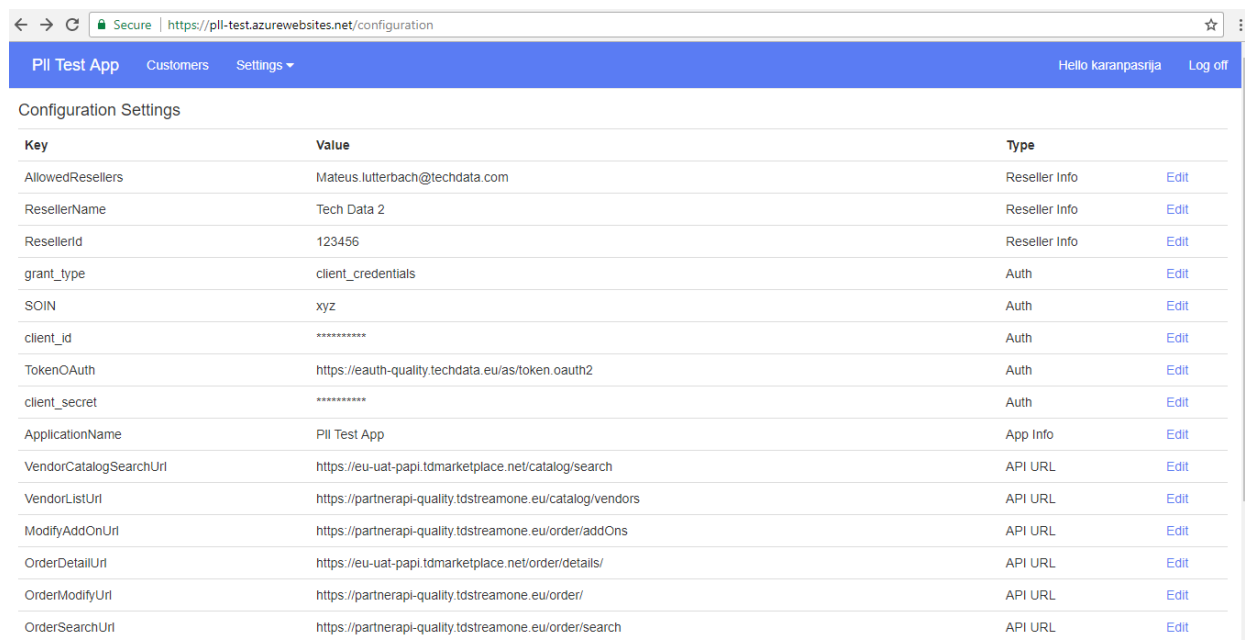
- Again open your Azure portal. Open your resource. Now open app service and look for URL.



- Hit the URL. Your application is hosted on Azure now.

Do the following mandatory changes:-

1. The URL you opened in the last step. Add “/configuration” to the end of the URL .
2. The following page will then open



The screenshot shows a web browser window with the URL <https://pii-test.azurewebsites.net/configuration>. The page title is "PII Test App" and the user is logged in as "Hello karanpasrija". The page displays "Configuration Settings" with a table of configuration keys, values, and types. Each row has an "Edit" link.

Key	Value	Type	
AllowedResellers	Mateus.lutterbach@techdata.com	Reseller Info	Edit
ResellerName	Tech Data 2	Reseller Info	Edit
ResellerId	123456	Reseller Info	Edit
grant_type	client_credentials	Auth	Edit
SOIN	xyz	Auth	Edit
client_id	*****	Auth	Edit
TokenOAuth	https://eauth-quality.techdata.eu/as/token.oauth2	Auth	Edit
client_secret	*****	Auth	Edit
ApplicationName	PII Test App	App Info	Edit
VendorCatalogSearchUri	https://eu-uat-papi.tdmartplace.net/catalog/search	API URL	Edit
VendorListUri	https://partnerapi-quality.tdstreamone.eu/catalog/vendors	API URL	Edit
ModifyAddOnUri	https://partnerapi-quality.tdstreamone.eu/order/addOns	API URL	Edit
OrderDetailUri	https://eu-uat-papi.tdmartplace.net/order/details/	API URL	Edit
OrderModifyUri	https://partnerapi-quality.tdstreamone.eu/order/	API URL	Edit
OrderSearchUri	https://partnerapi-quality.tdstreamone.eu/order/search	API URL	Edit

Now Your application is ready to go

