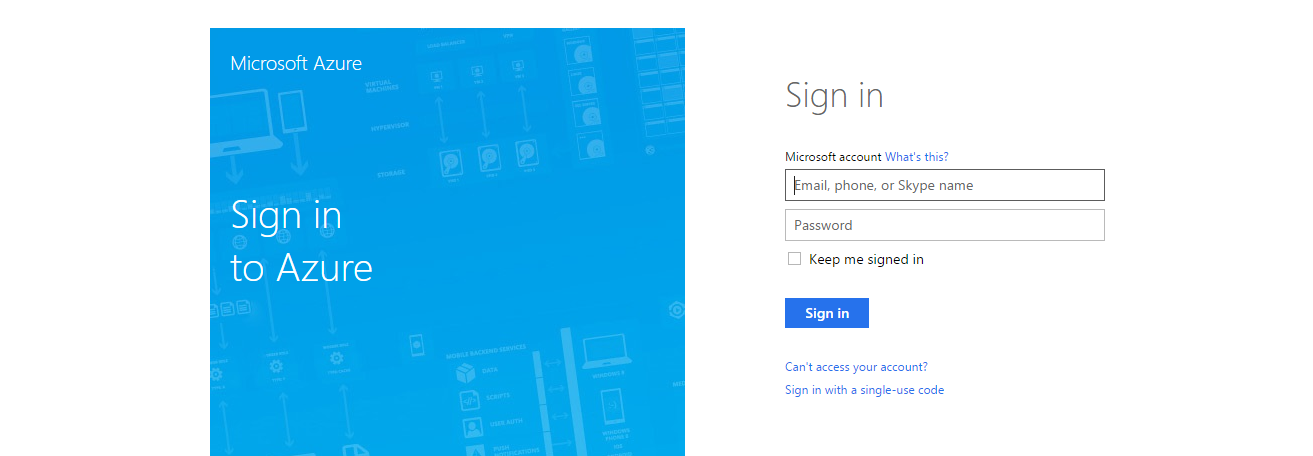
Azure portal

If you want to publish your project into azure, please follow the steps which are given below.

Step1- [https://**portal**.**azure**.com/](https://portal.azure.com/)

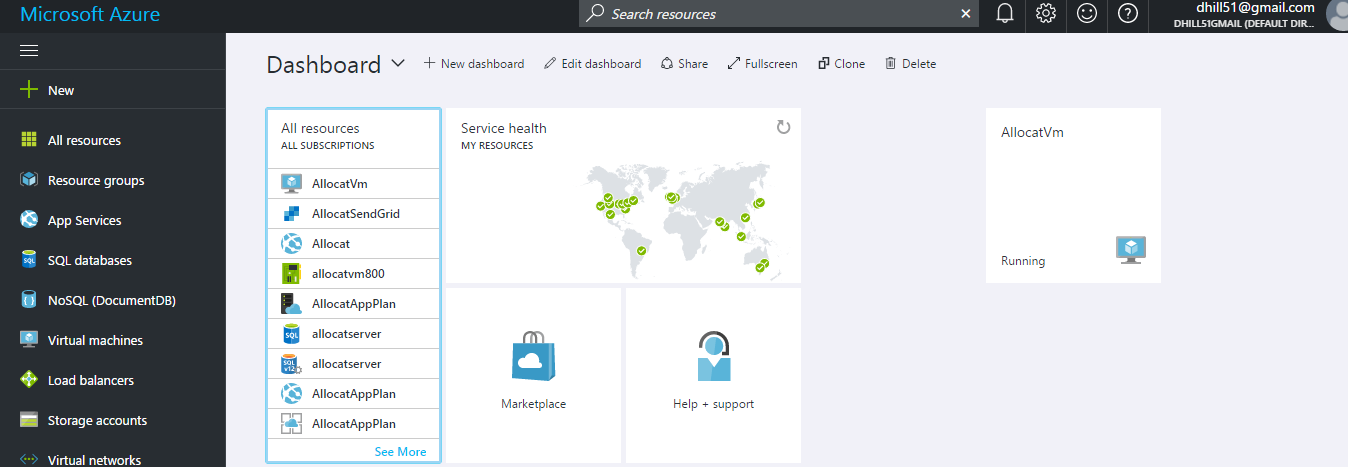
Step-2- If you are registered, Please Enter EmailId and Password.



And press sign in button.

Step2.1- If you are not registered, press “signup now” button which is in the bottom of the page and please enter all information and get registered with azure.

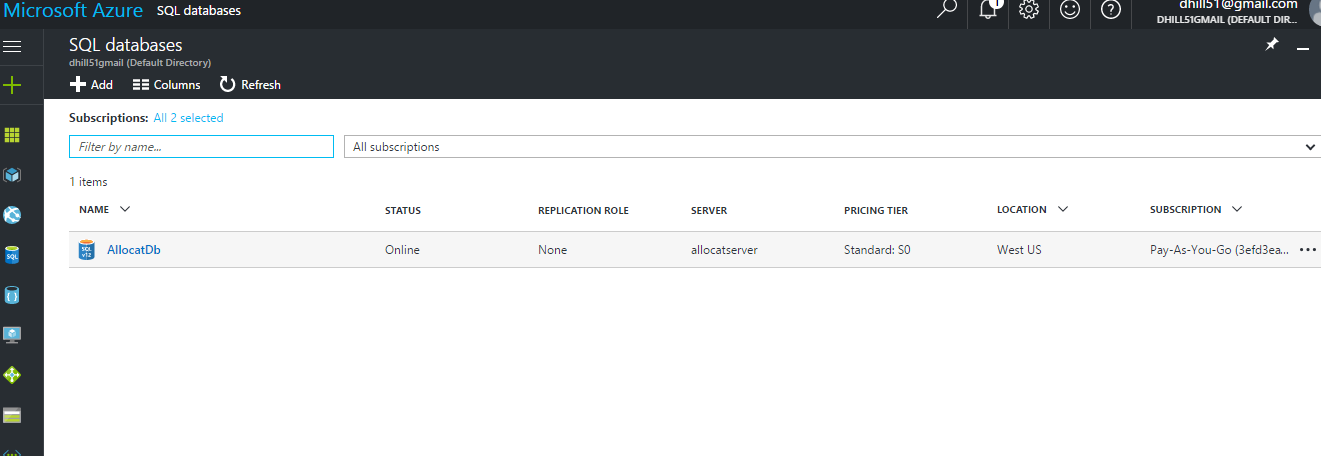
Step3- Now, you will be redirected to azure dashboard.



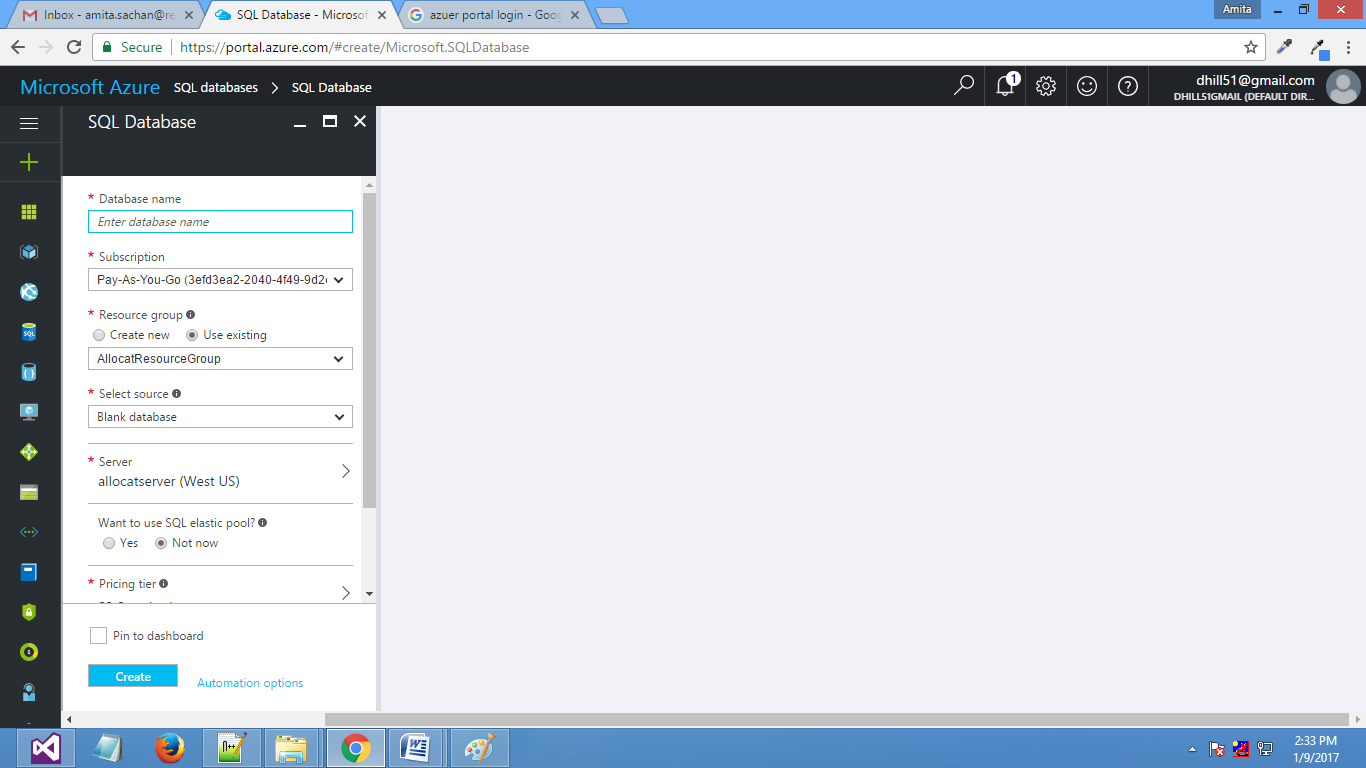
**Azure Database**

After sign in, you can see Sql Databases menu in left pane.

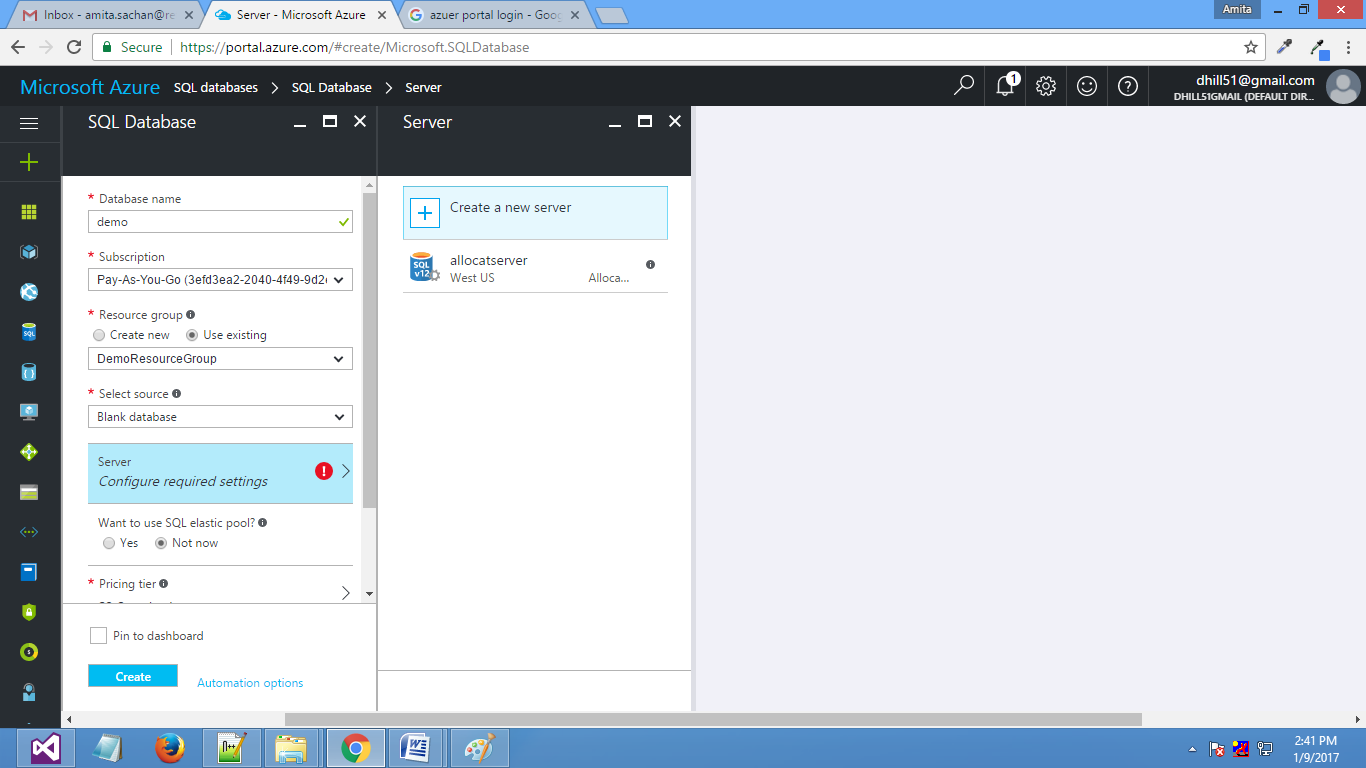
**Step1-** when you click sql database, it will open a new window where you can see “add” button on top.



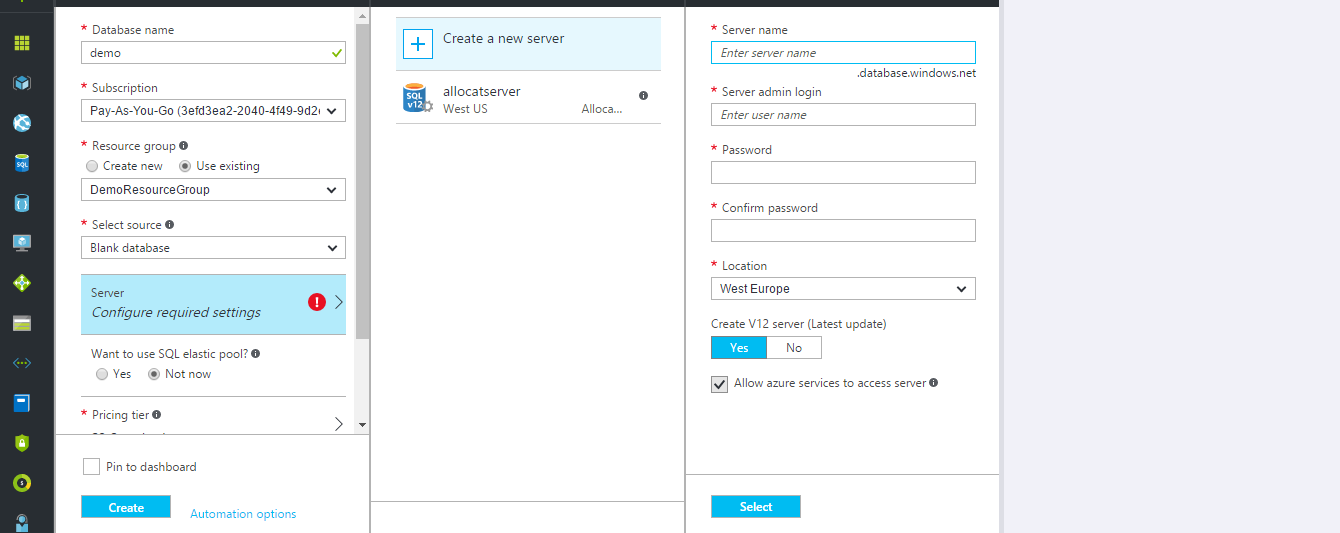
**Step2-** when you press add button. A window will appear.

****

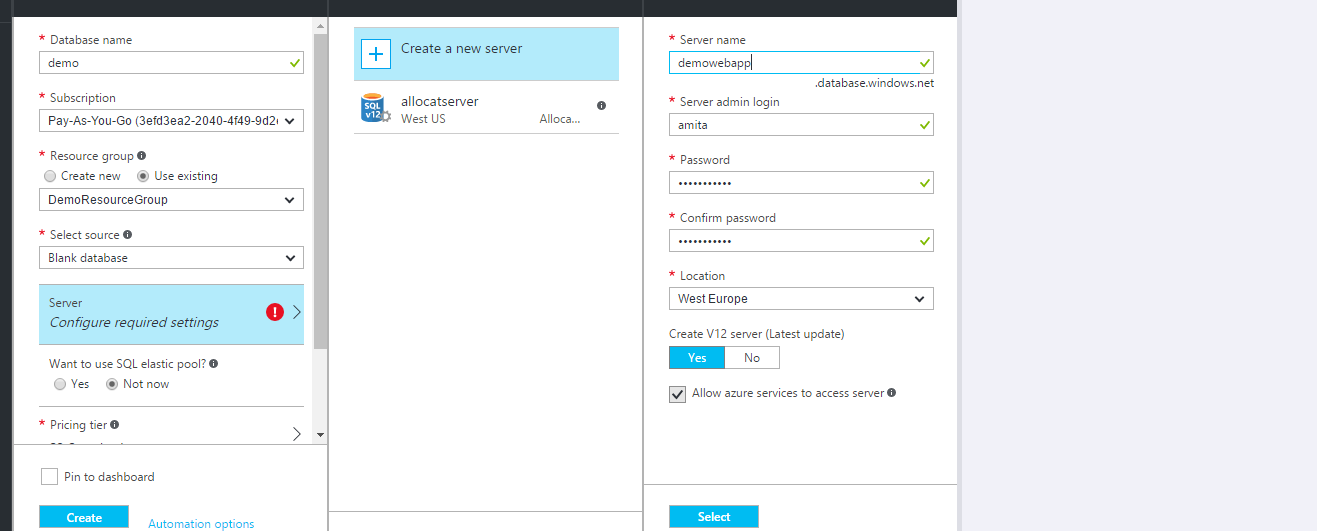
**Step3-** Enter database name, subscription (pay-as-you-go), resource group (we already created resource group so select it from use existing), select blank database from select source, now click on server, it will open a new window.

****

**Step-4** Now click on create a new server button; it will open a new window.

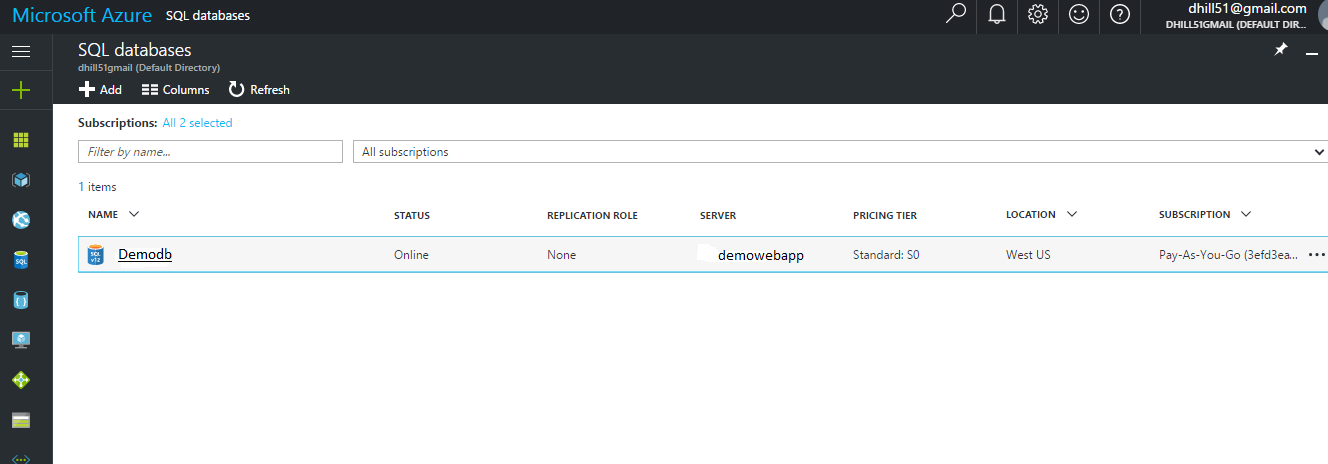
****

**Step-5** Enter server name, admin login, password. (Note down server name, adminlogin, password).

****

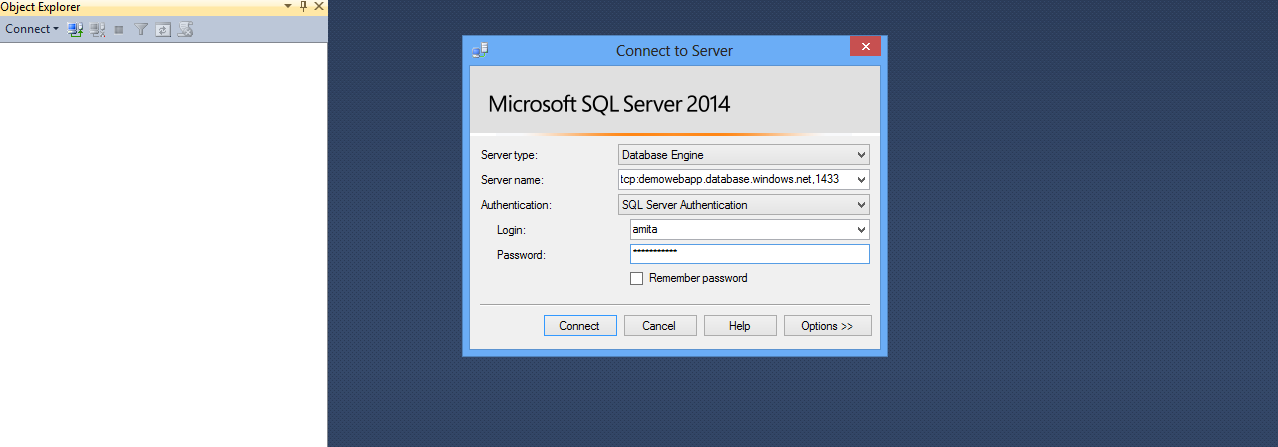
Press select button and then choose pricing tier and press create button.

Step6- Now you will be redirected to sql databases dashboard page. And here you can see that your database is created.

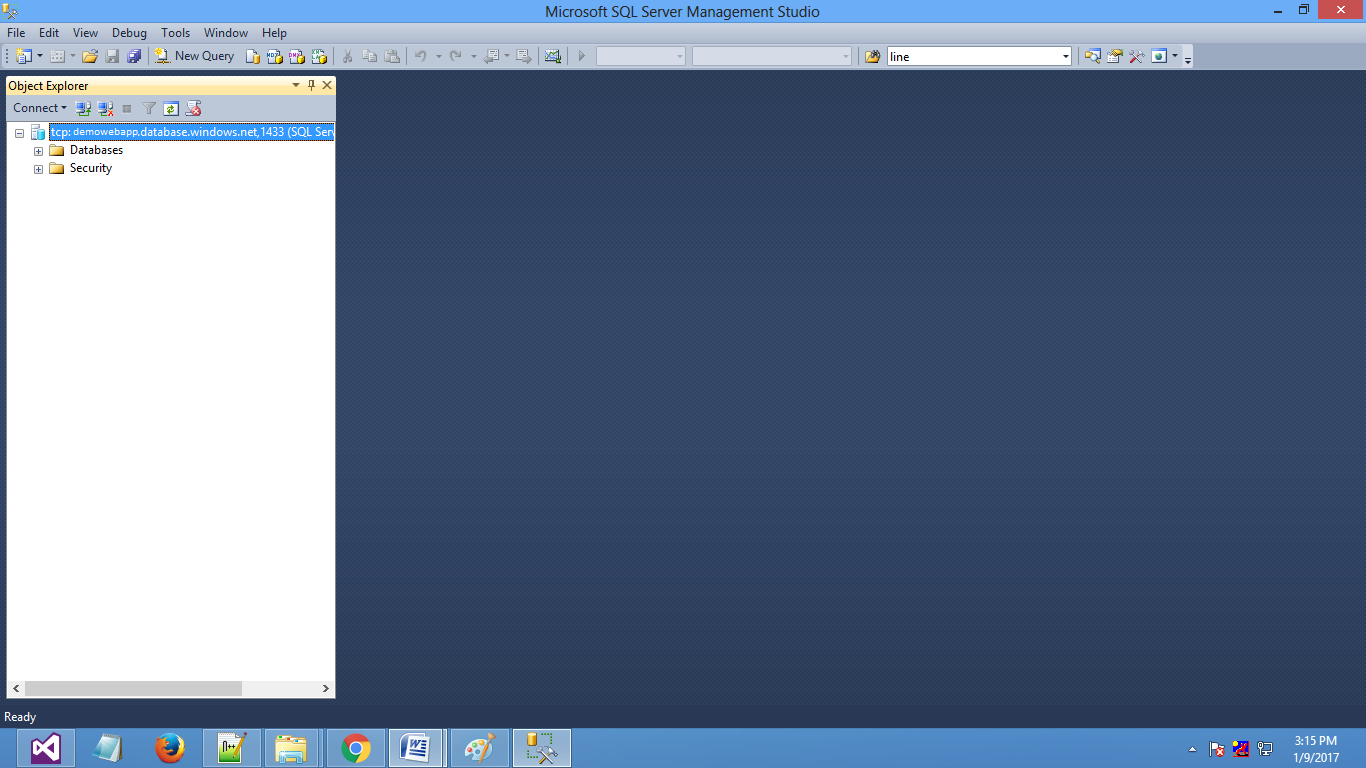
 Your database server is created now.

Step7- when you click on Demodb, it will open a new window,in this window you can see “show database connection string” when you click on this, you can see your server name, user name and password.

Step8- Open sql management studio and use these credentials to connect with live server.



And press connect button. A screen should open like this-

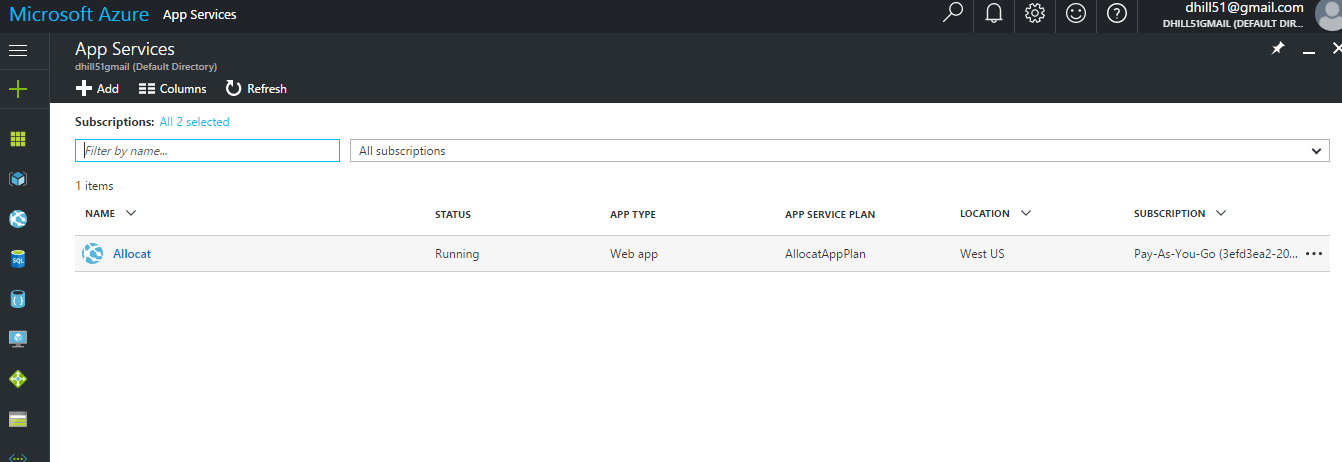


Step9- Open query window in sql management studio and write create database command in query window and execute it. It will create database in your management studio.

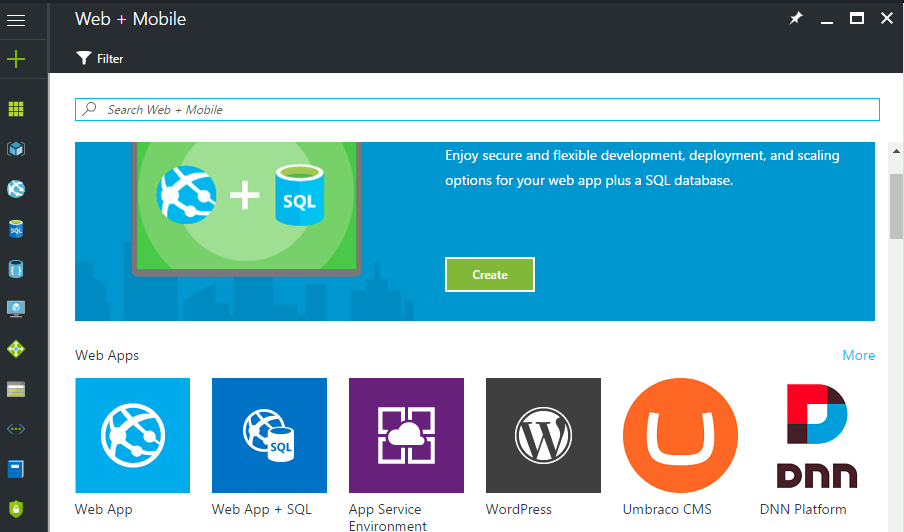
Step-10 Modify database and use it. And now your database is live.

**Azure Web apps**

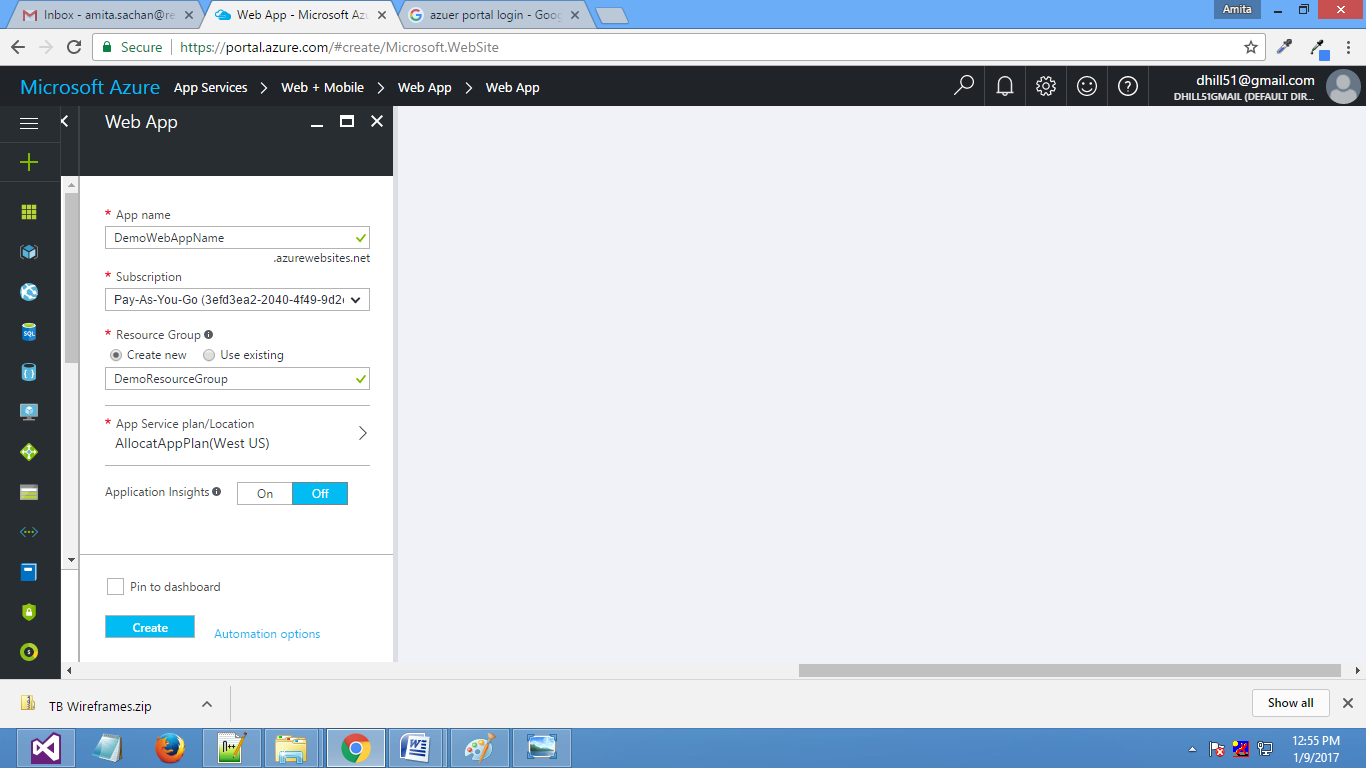
Step1- In the left pane, you can see “app services” menu. When you click “app services” then a new window will open.



You can see “Add” button on top. For creating a new app service, click add button. A new window will open.



Select web app and press create button.A new window will open.

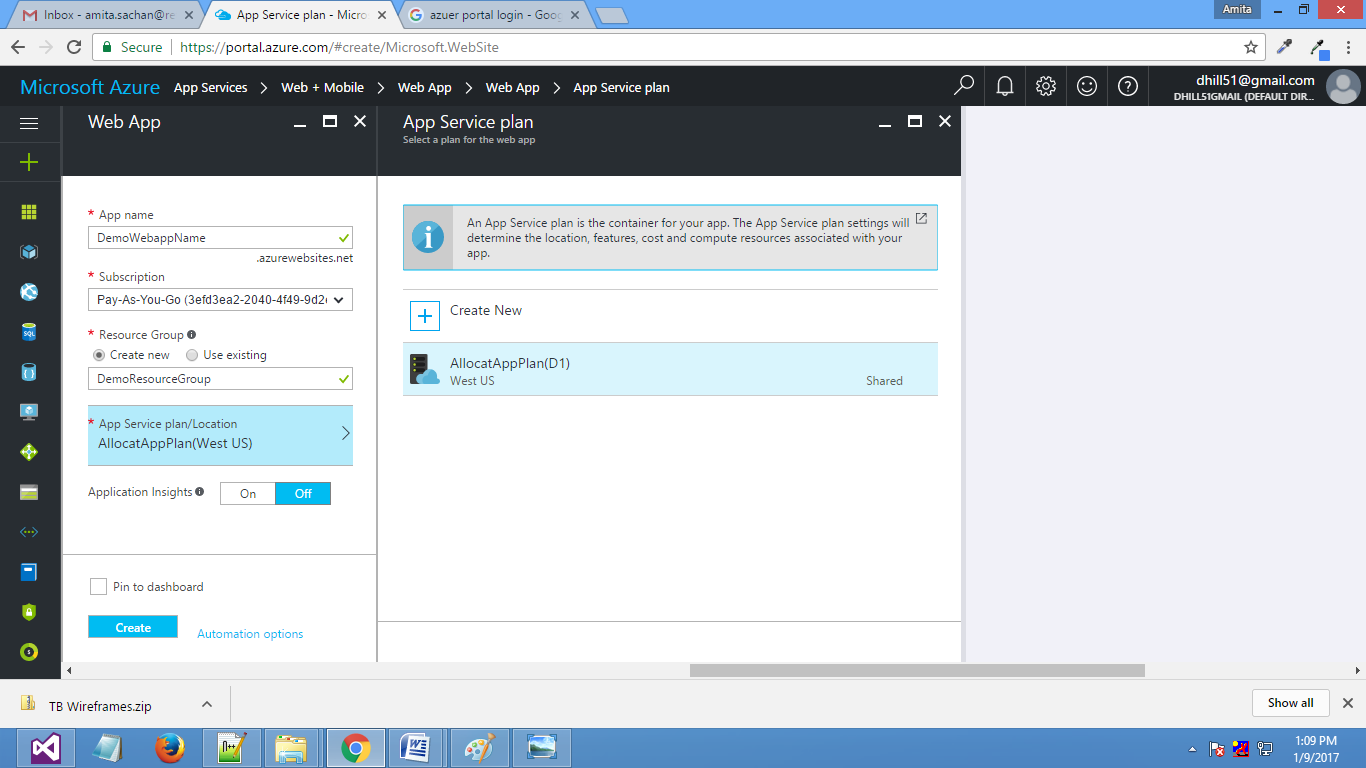


Enter web app name.

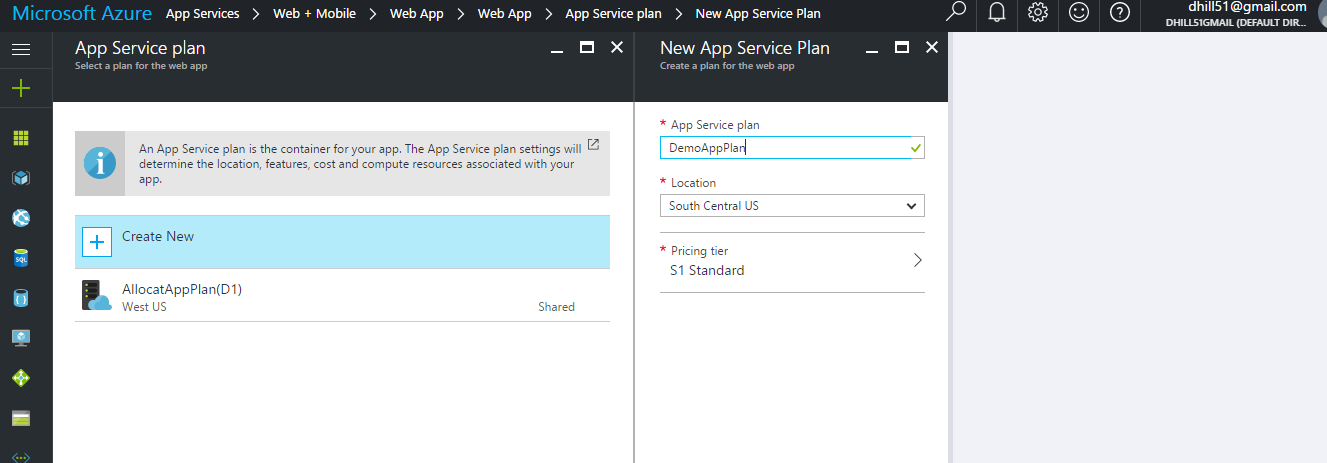
Subscription is always pay as you go.

In the resource group, select create new radio button and enter resource group name.

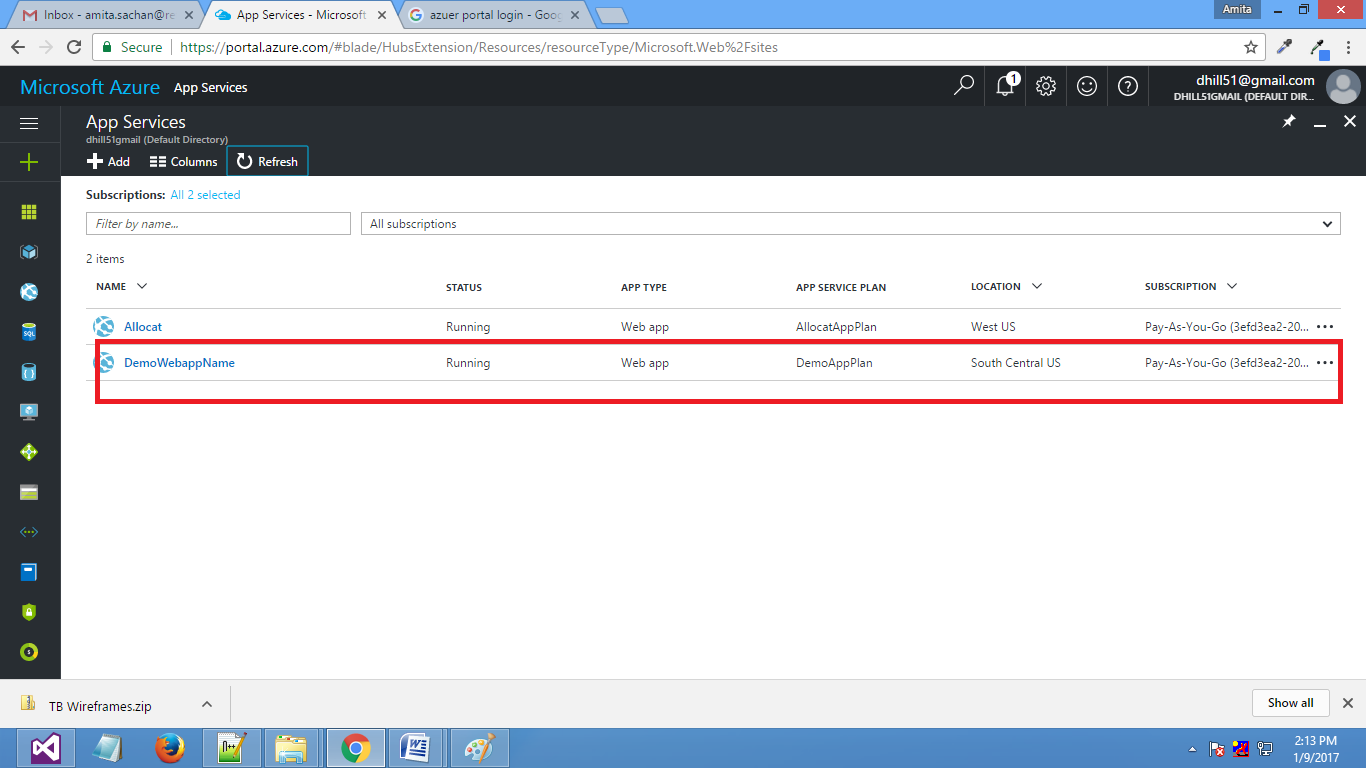
When you click App service plan, it will open a new window in which you can see “create new” button.



When you click create new button,A new window will appear.



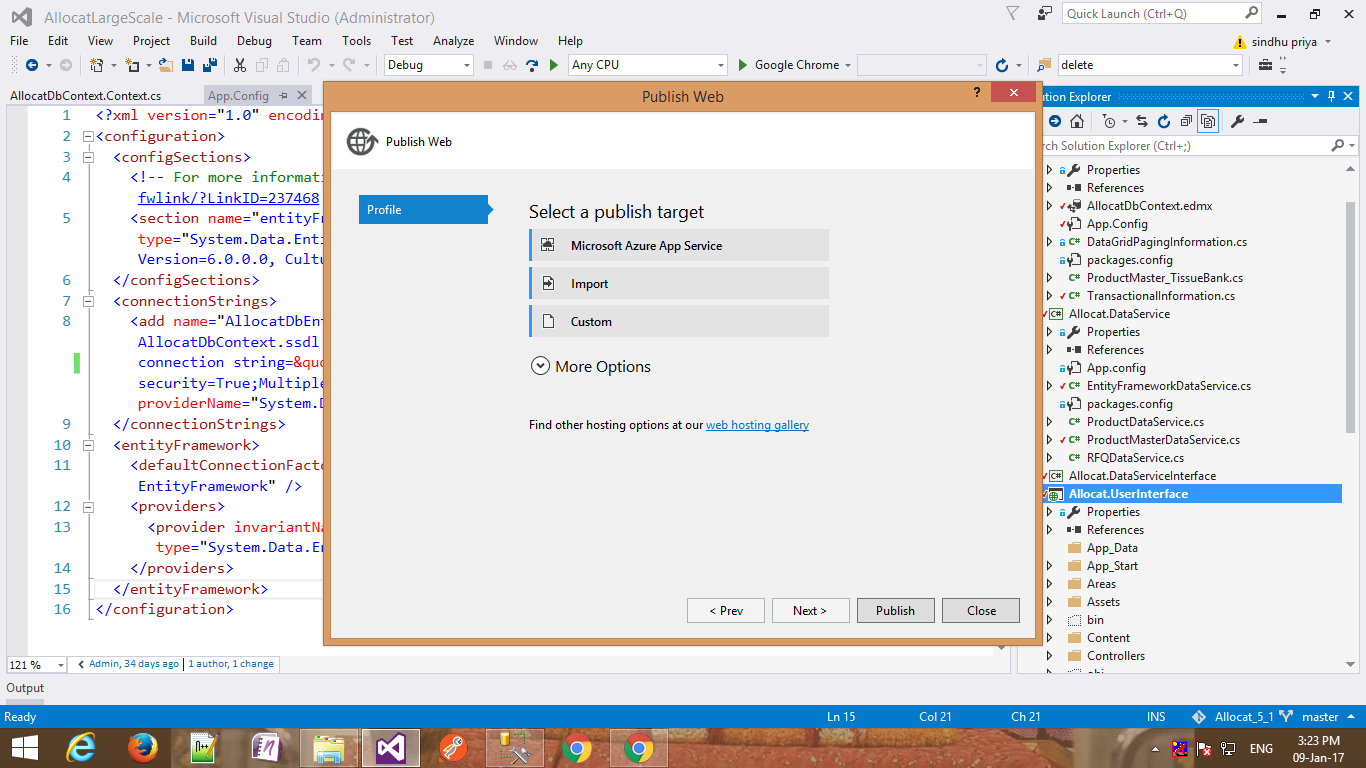
Enter App service plan name, location, pricing tier and press ok and then create. Now you will be redirected to app services window where you can see your newly created app service.



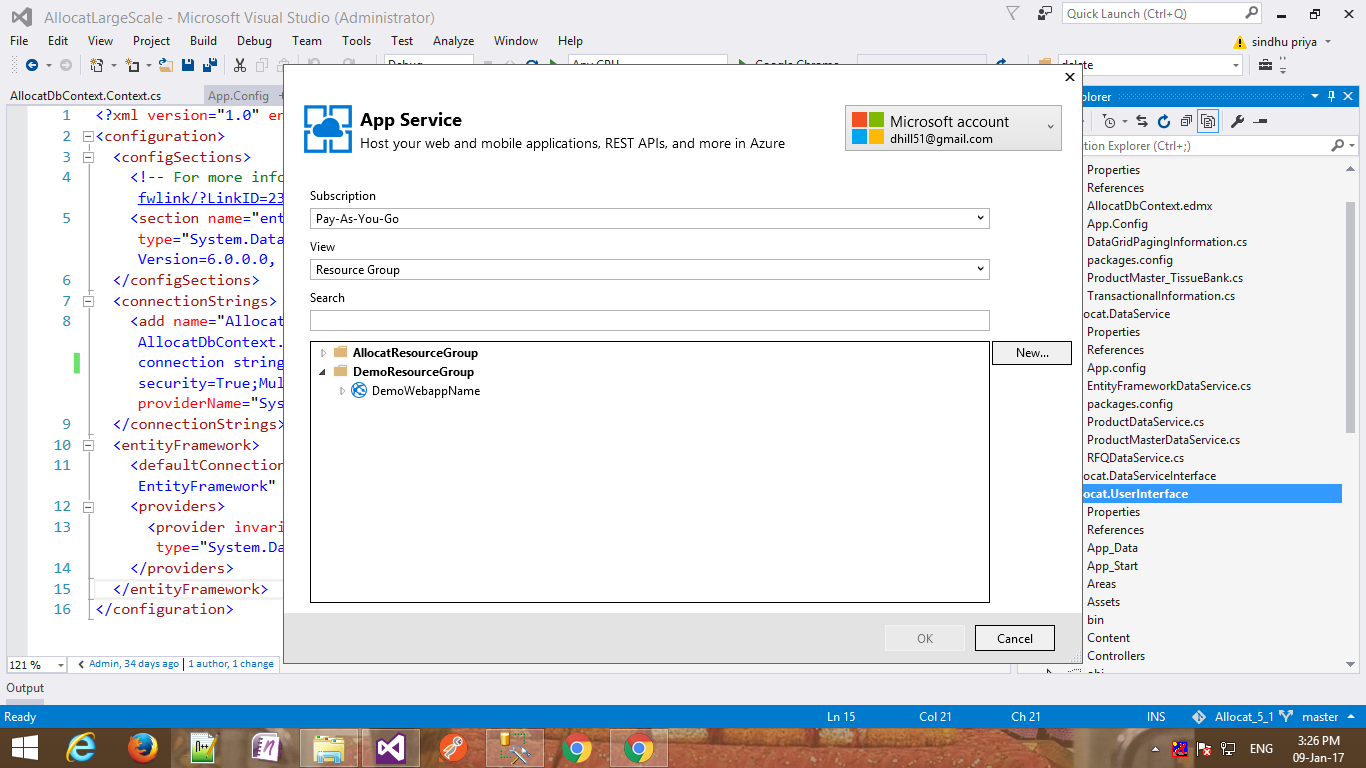
Now your app service is created.

Step2- Open your project in visual studio and right click on your project folder and choose publish.

Step4- after doing publish, you can see screen like this-

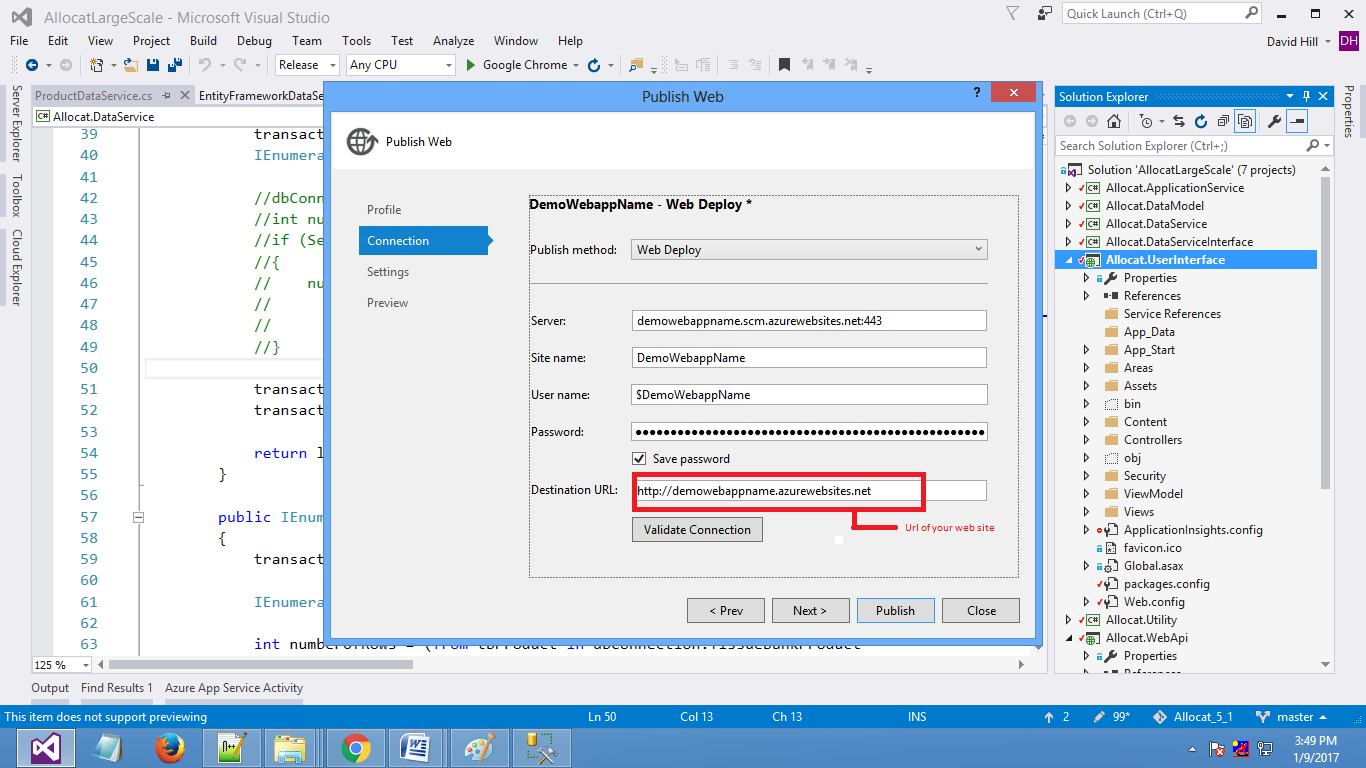


Step4- click Microsoft azure app service, it will open a new screen. After entering your account details you can see your email id in right side.



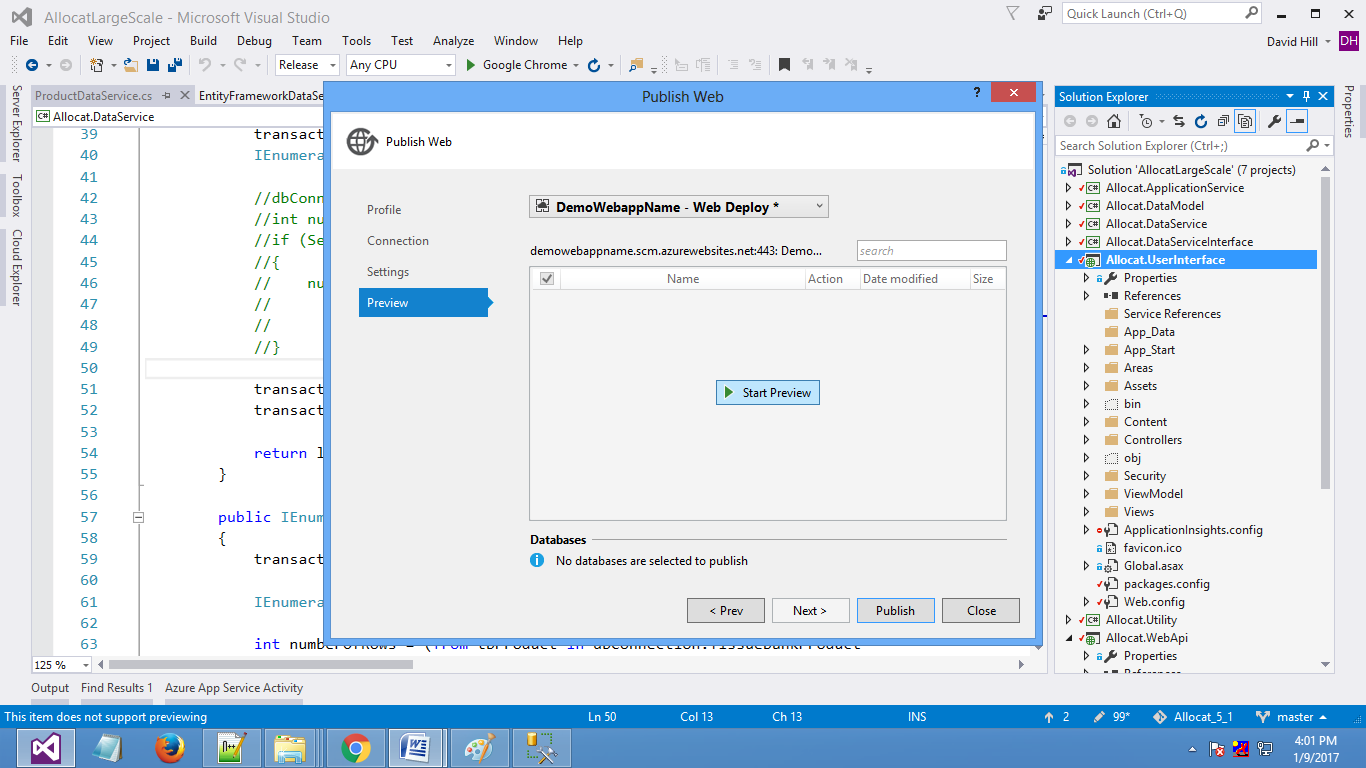
You can see demoResourceGroup and demoWebappName, select demowebappname and press ok.

Step5- After selecting ok, you can see connection menu fields are automatically filled with all details.



Step6- validation connection validates your connection with server and produces result in green tick sign.

Step7- click next button to proceed, it will open a new window.

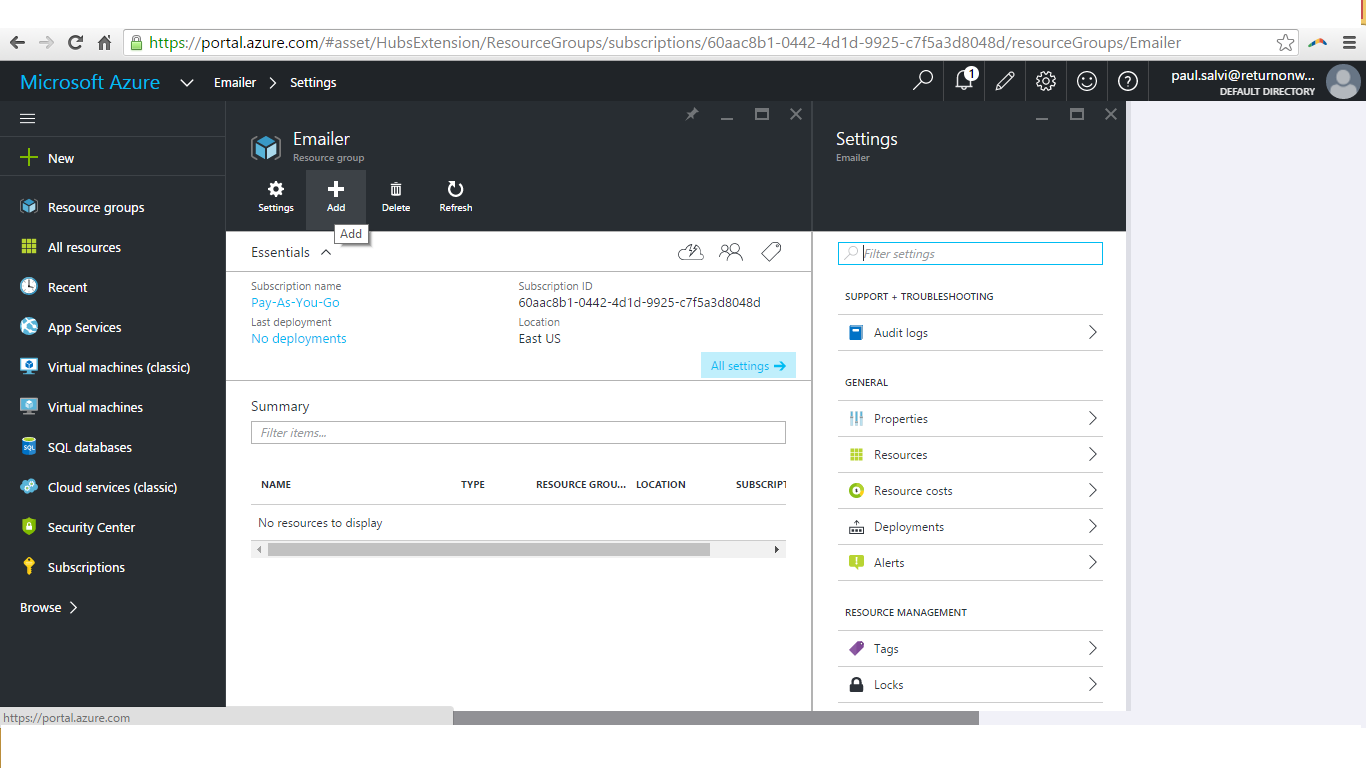


Step8- When you click start preview, it will show all your project data which is going to publish in server. And then press publish button in the bottom of page. It will publish your project in live server.

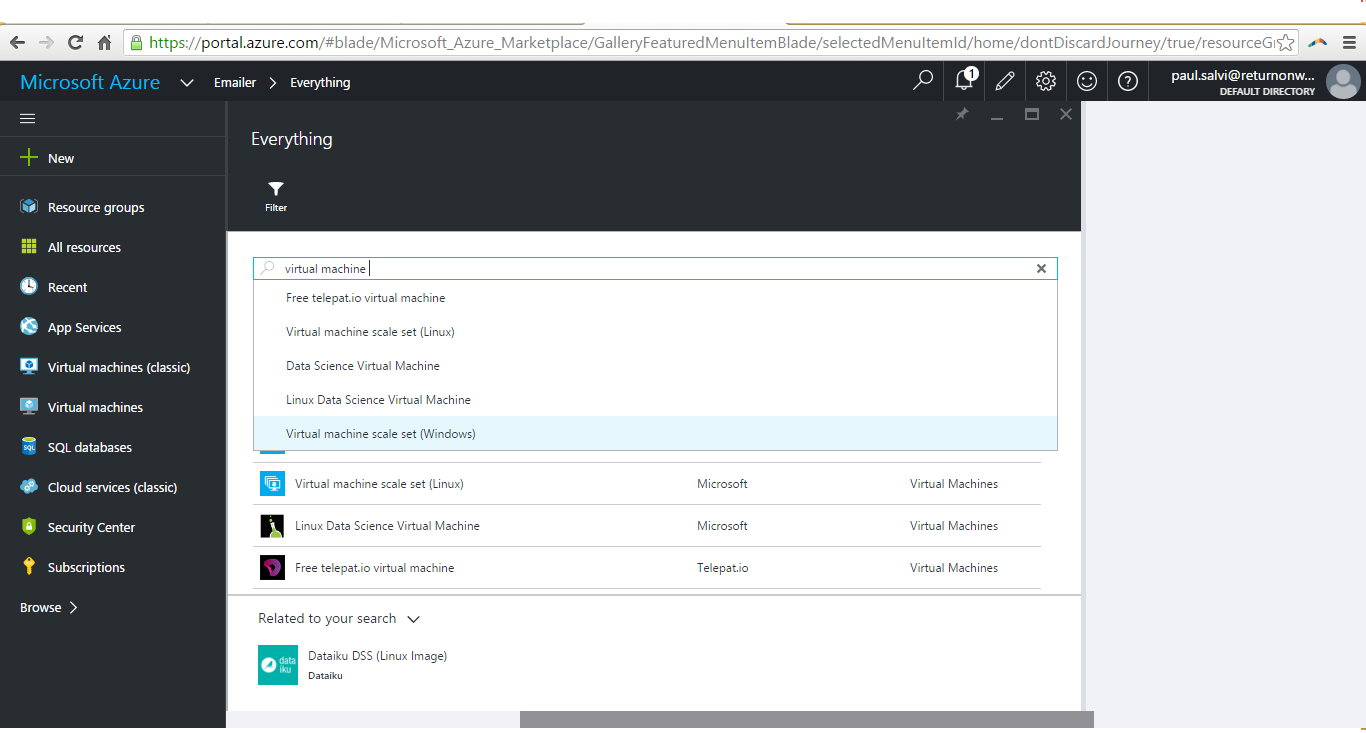
Step9- You can see URL on browser.

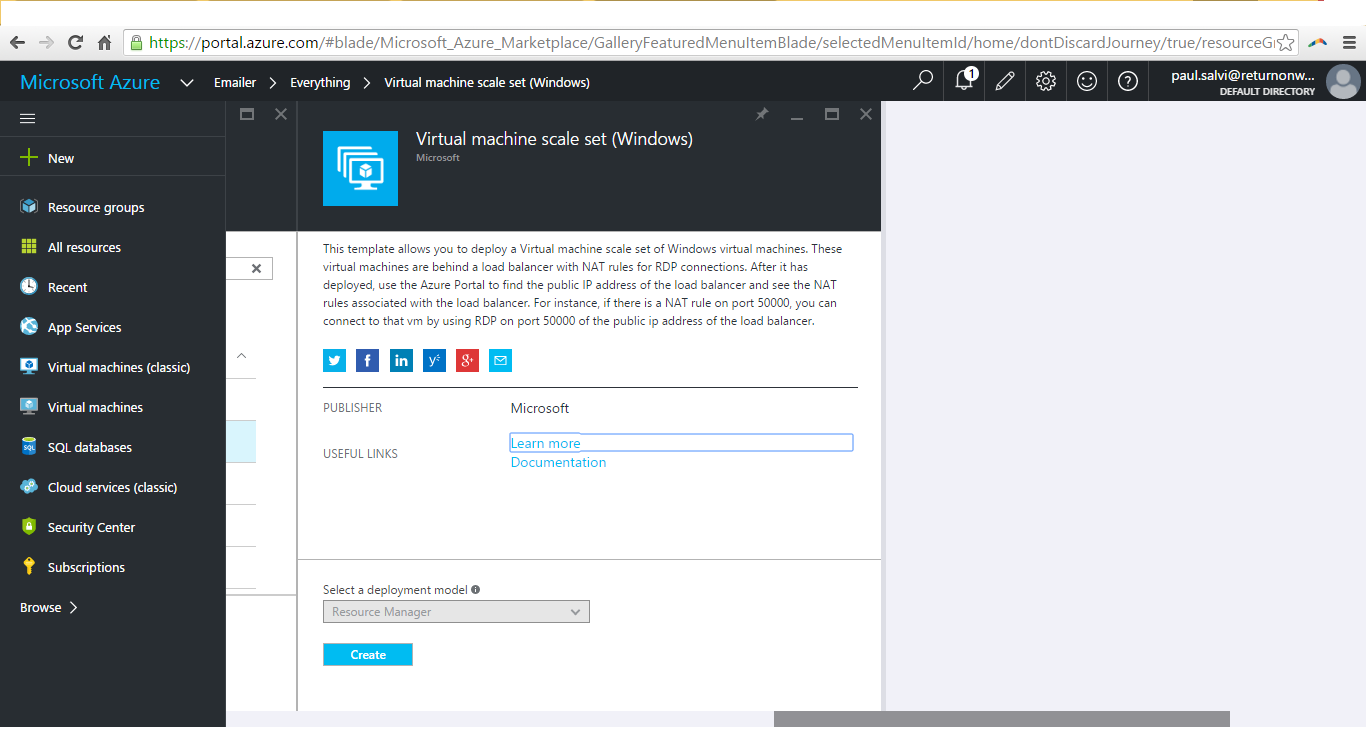
**Virtual Machine**

1. Now we can add virtual machine to resource group.
2. Click on newly created resource group.
3. lick on add button to add new resource

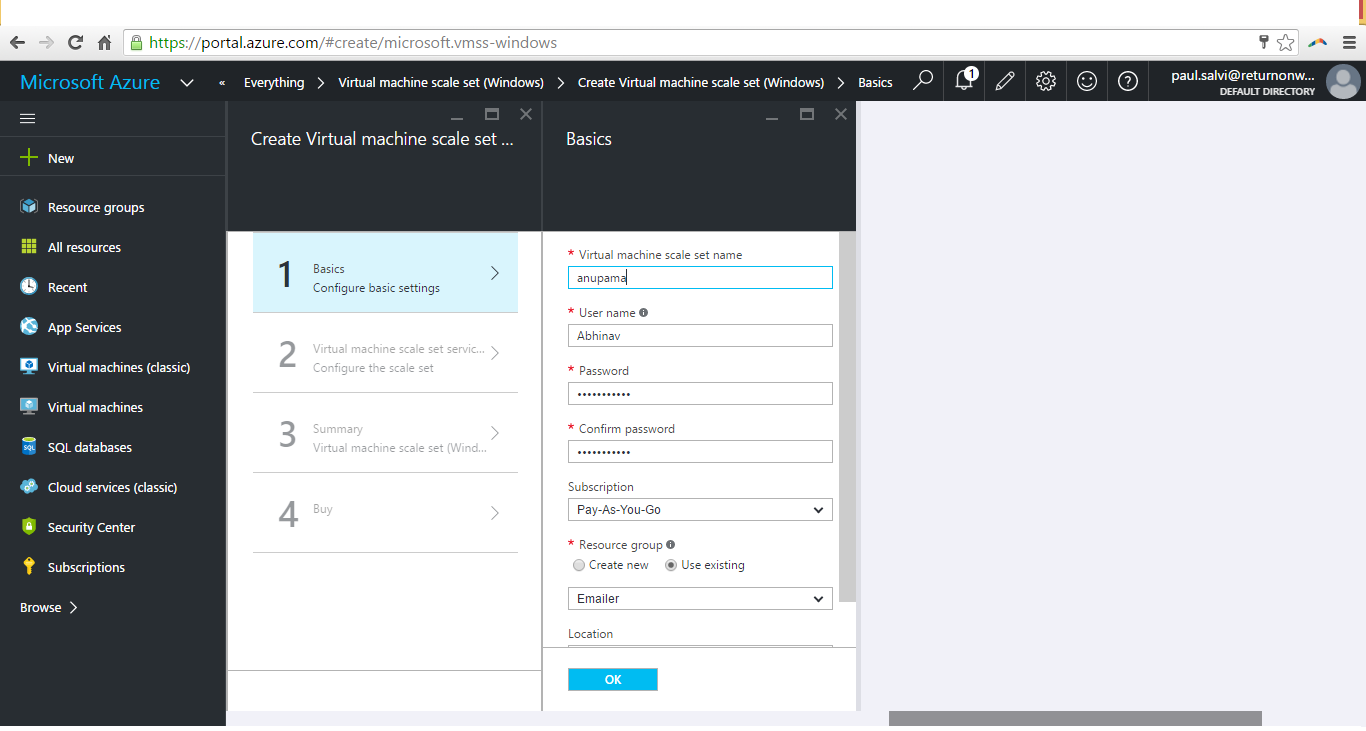
****

1. In search tab writes **Virtual Machine.** in dropdown select **virtual machine scale set (windows)**

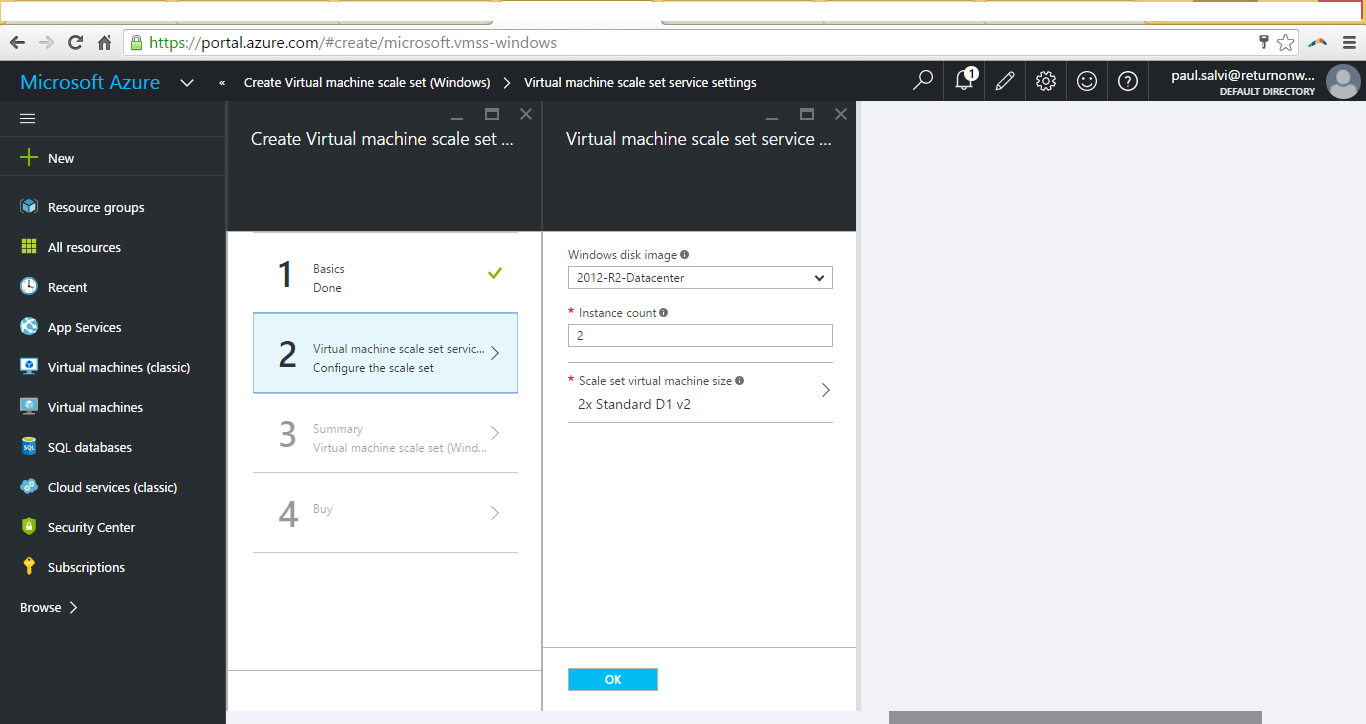
****

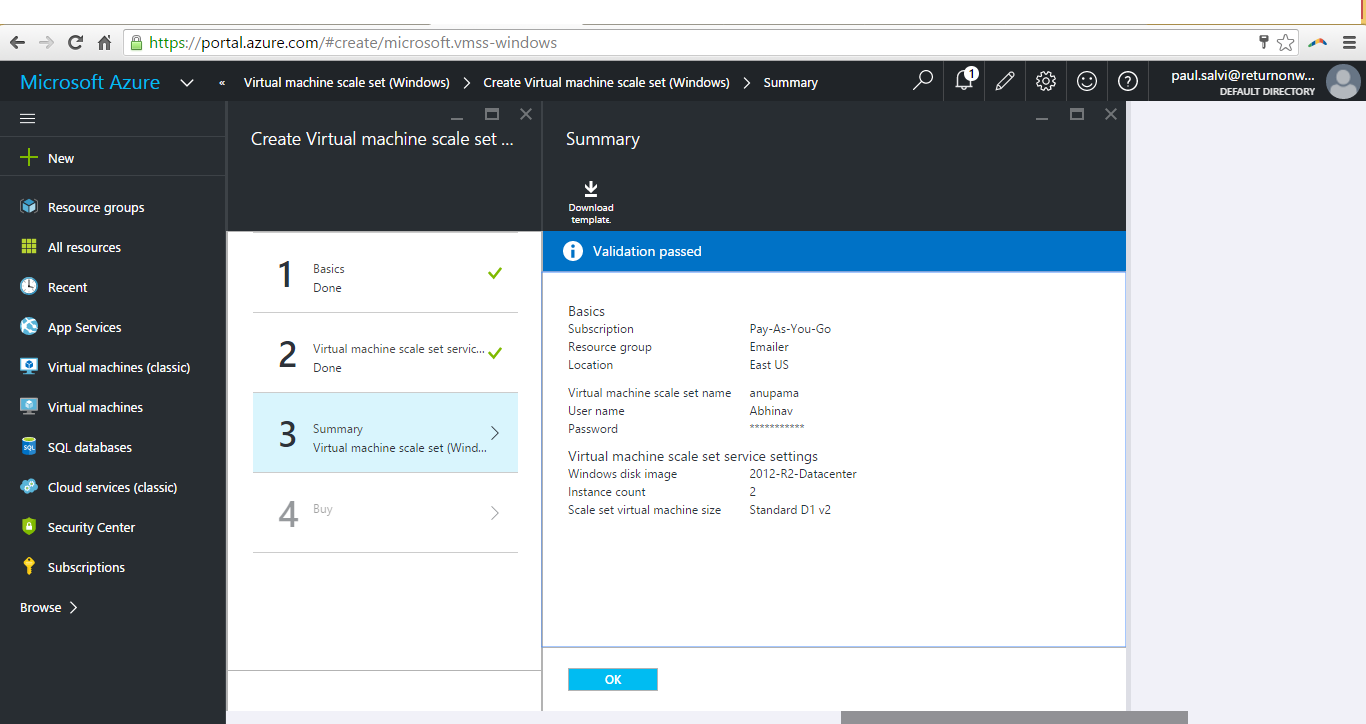
****

1. Create virtual machine scale set.

****

1. Configure scale set

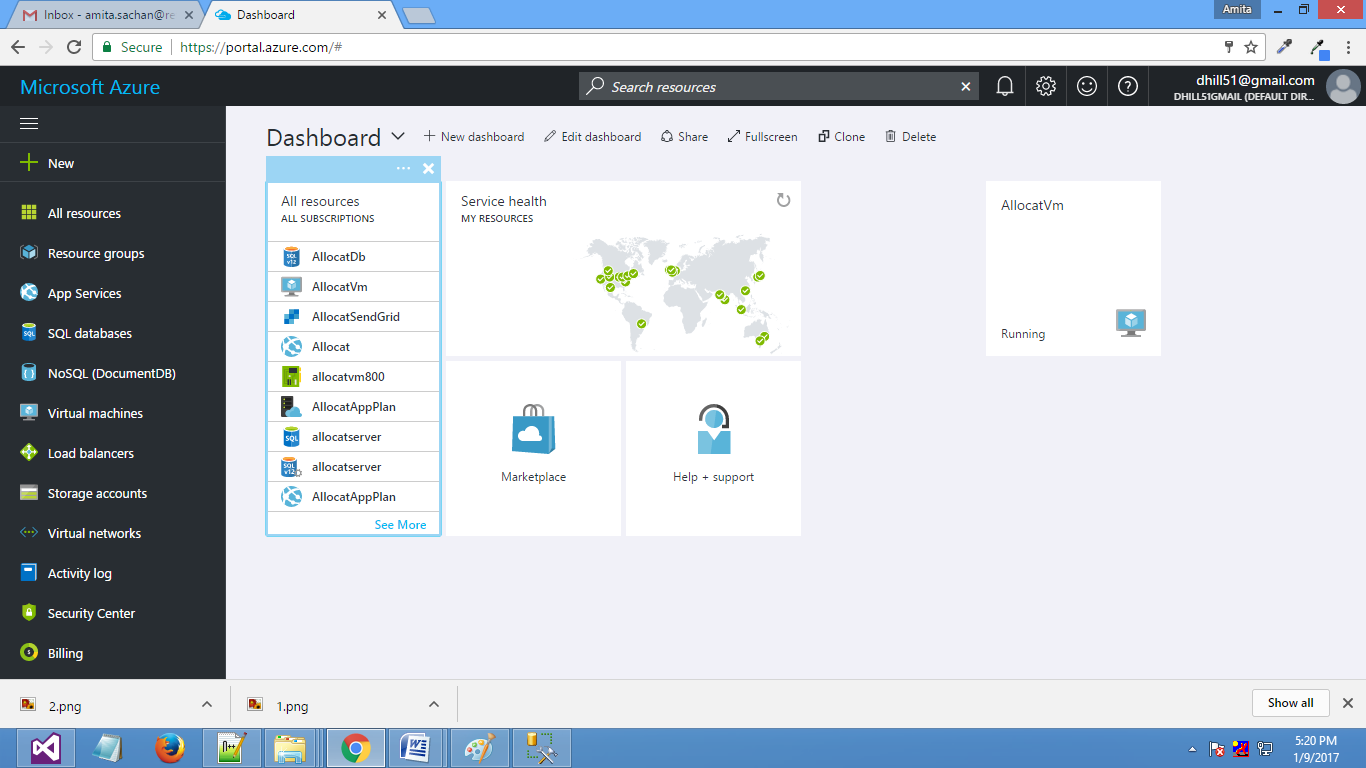
****

****

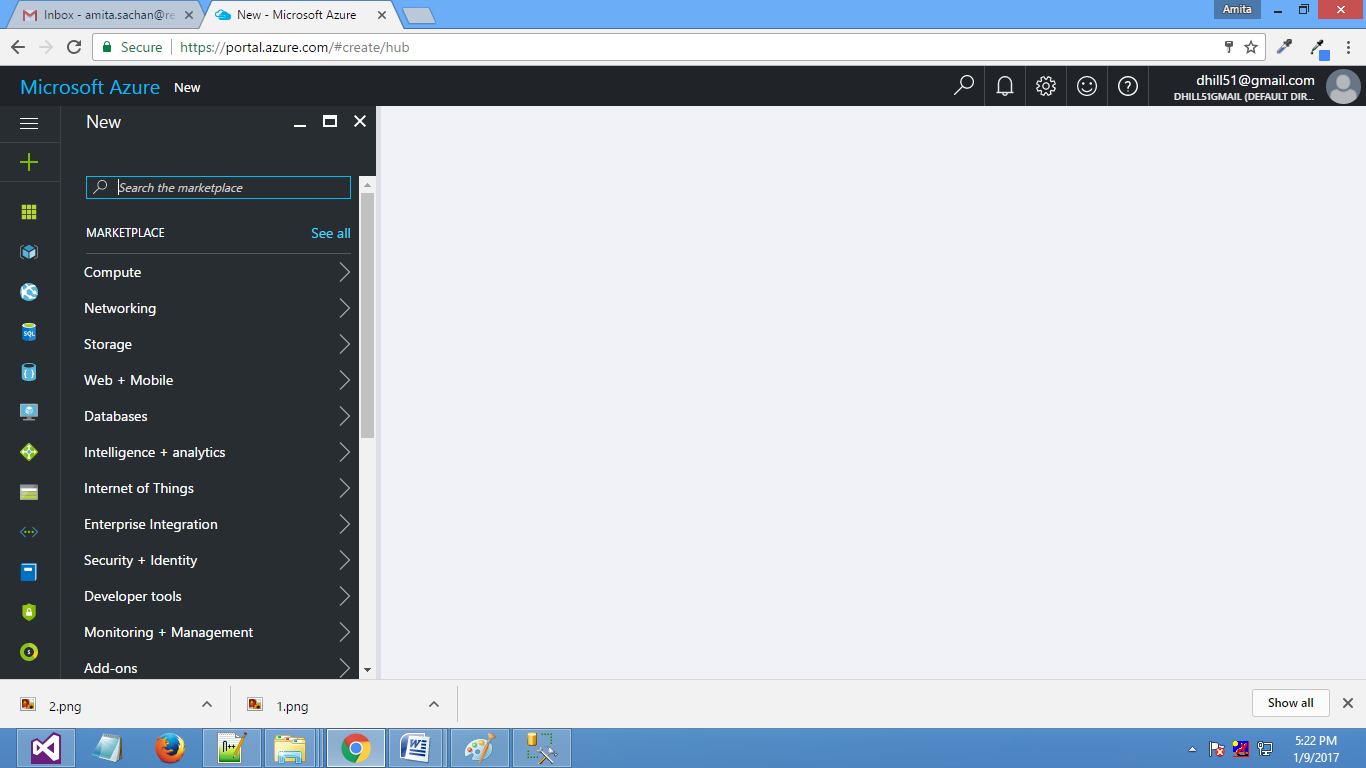
1. for step4 buy appropriate plan according to you are need.

**SendGrid**

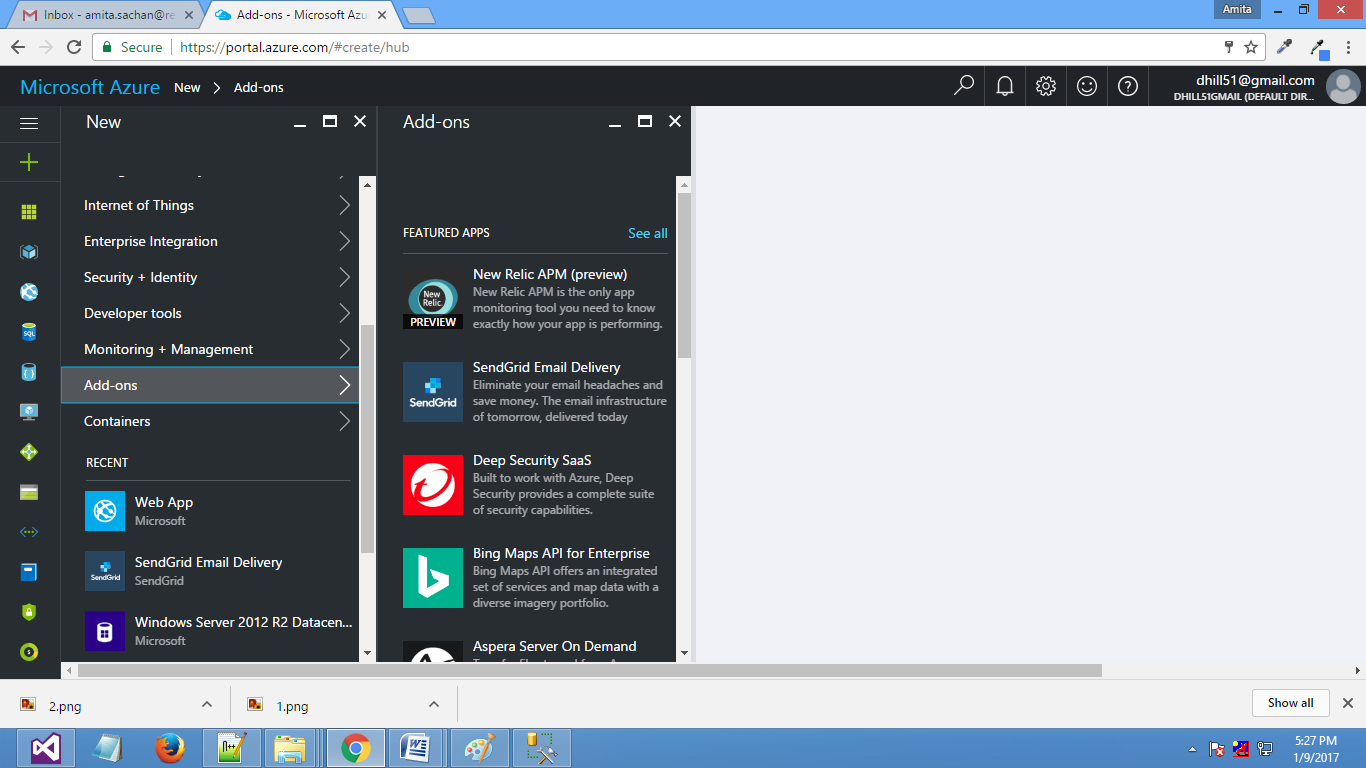
**Step-1** After login, you can see “+New” menu in the left pane.

****

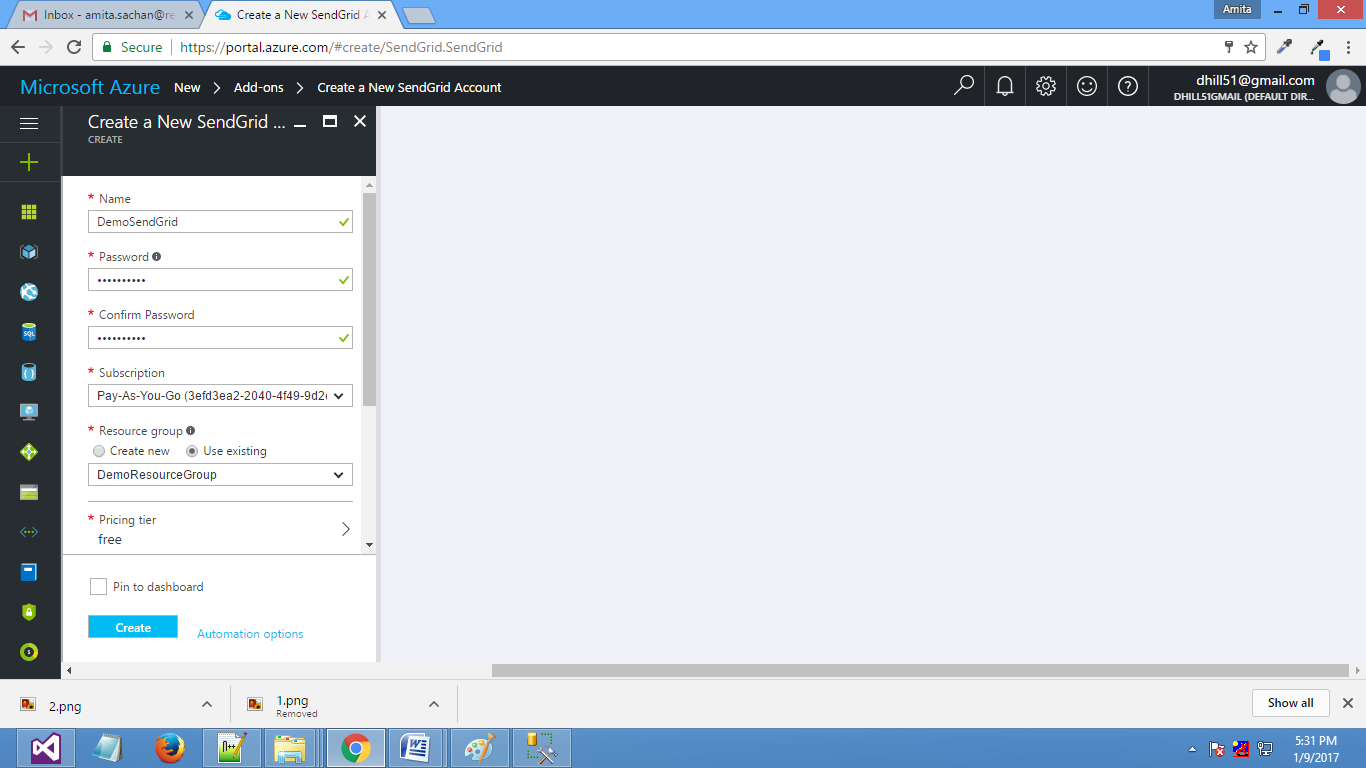
**Step-2** When you click “+ new”, you cans see “Add ons” in the bottom

****

When you click on Add ons, it will open a list. In which you must have to select “SendGridEmailWithDelievery”.

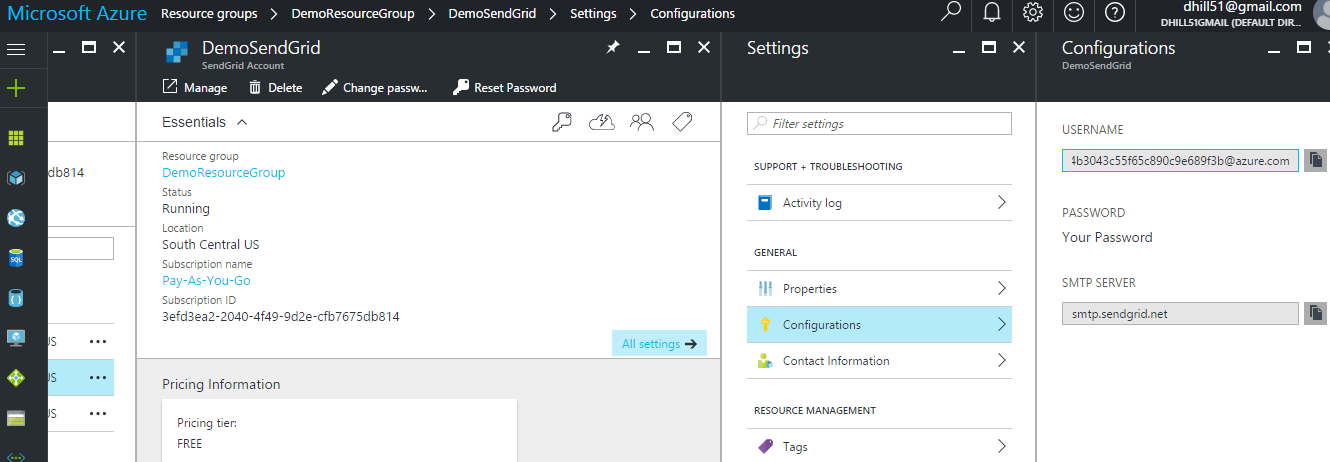


Step-3 After clicking sendgrid email delivery, a new window will open.



Step5- Press create button.

Step6- Go to all resources and select send grid which you have just created, it will display configuration menus. After clicking this, you can see server and user name and use it in your code for sending mail.



**MSMQMailService**

Overview – there are 4 applications, used in whole solution.

1. Netmester.Mail.Serialization – used for serialization
2. MSMQMailService – Needs to be installed on server, which sends email after retrieving from message queue
3. MailSenderLibrary – Sends emails to Message queue
4. TestApplication – Consumer

Steps are – Following steps needs to be performed in order to activate email functionality :

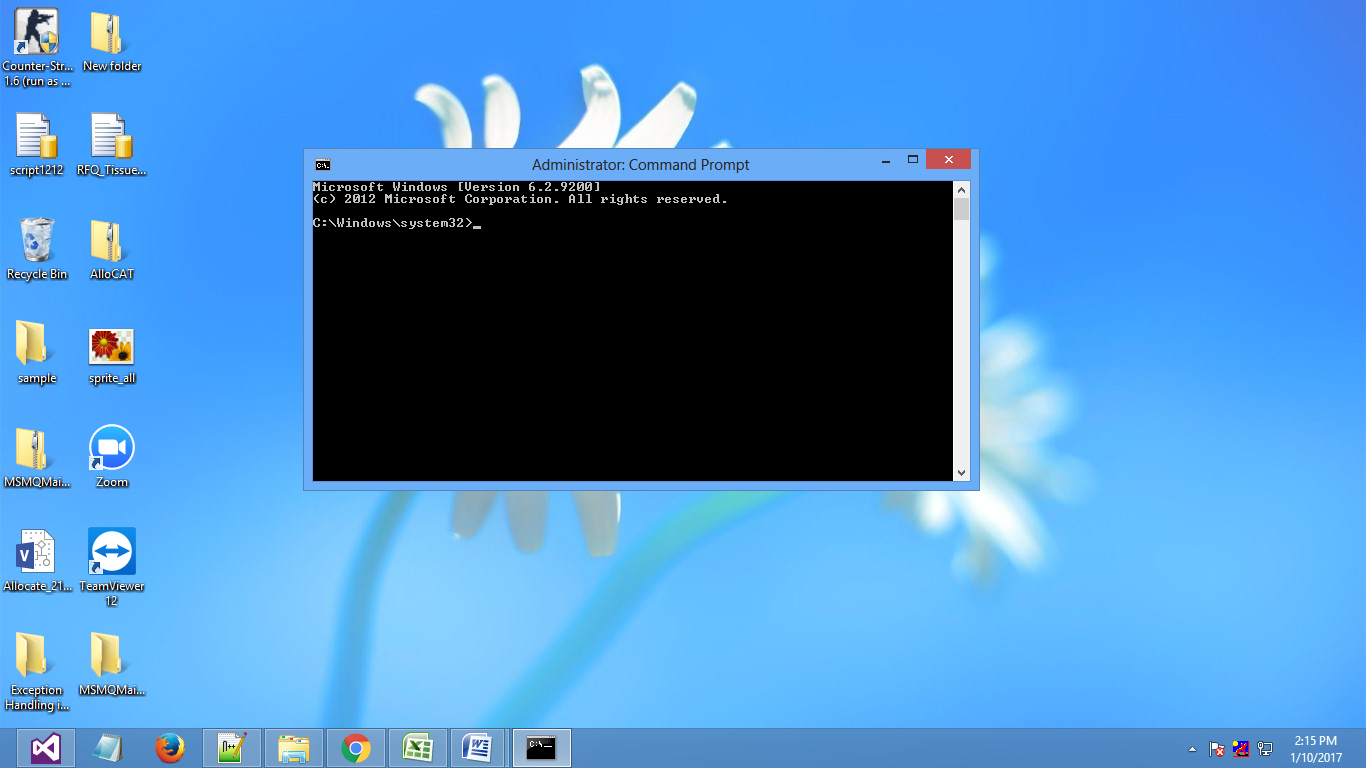
1. Create message queue

* Right click on “My Computer”
* Click on “Manage”
* Expand Services and Application
* Now you can see “Message Queuing”
* Expand “Private Queue”, click on “New Private queue”

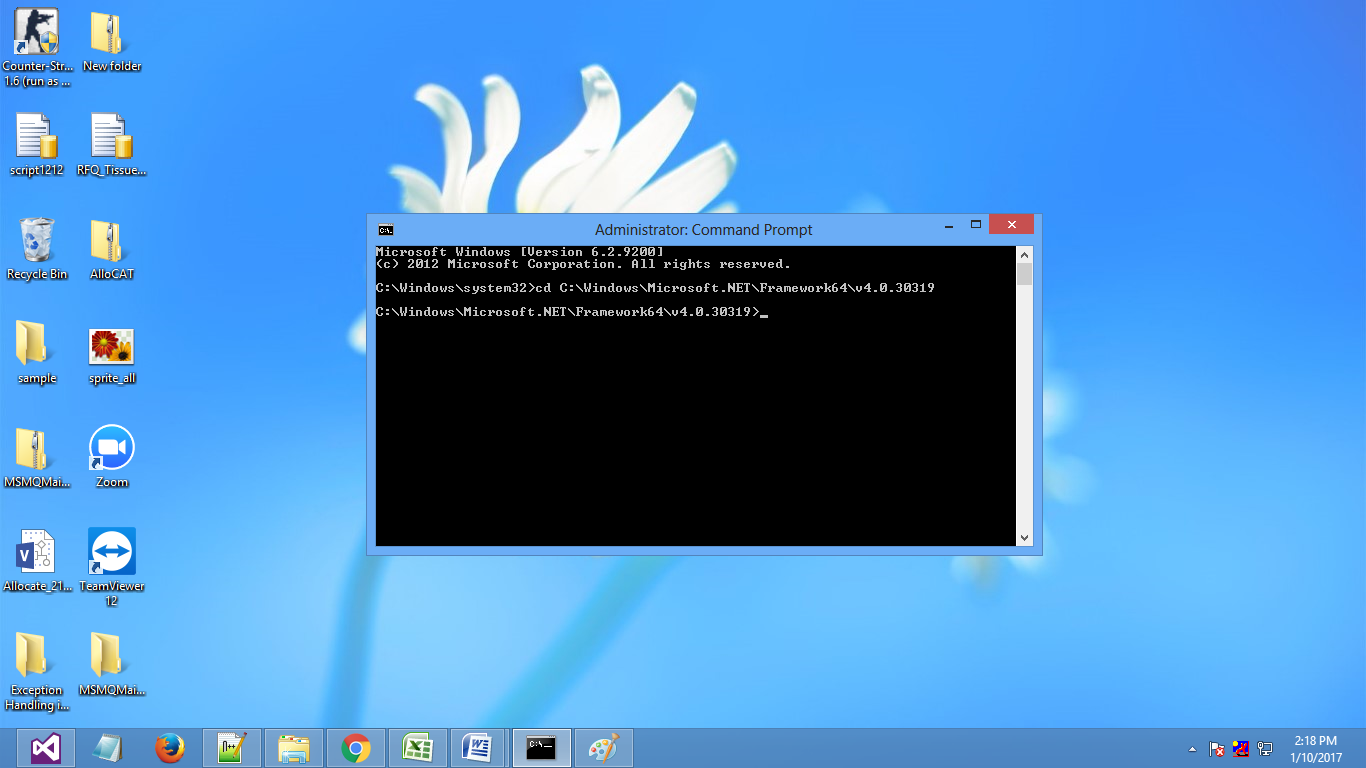
2. Install the service 'MSMQMailService' using installutil.exe on server

**Installing Windows Service**

**Step1**- open command prompt and run it as admin. You can see window like –

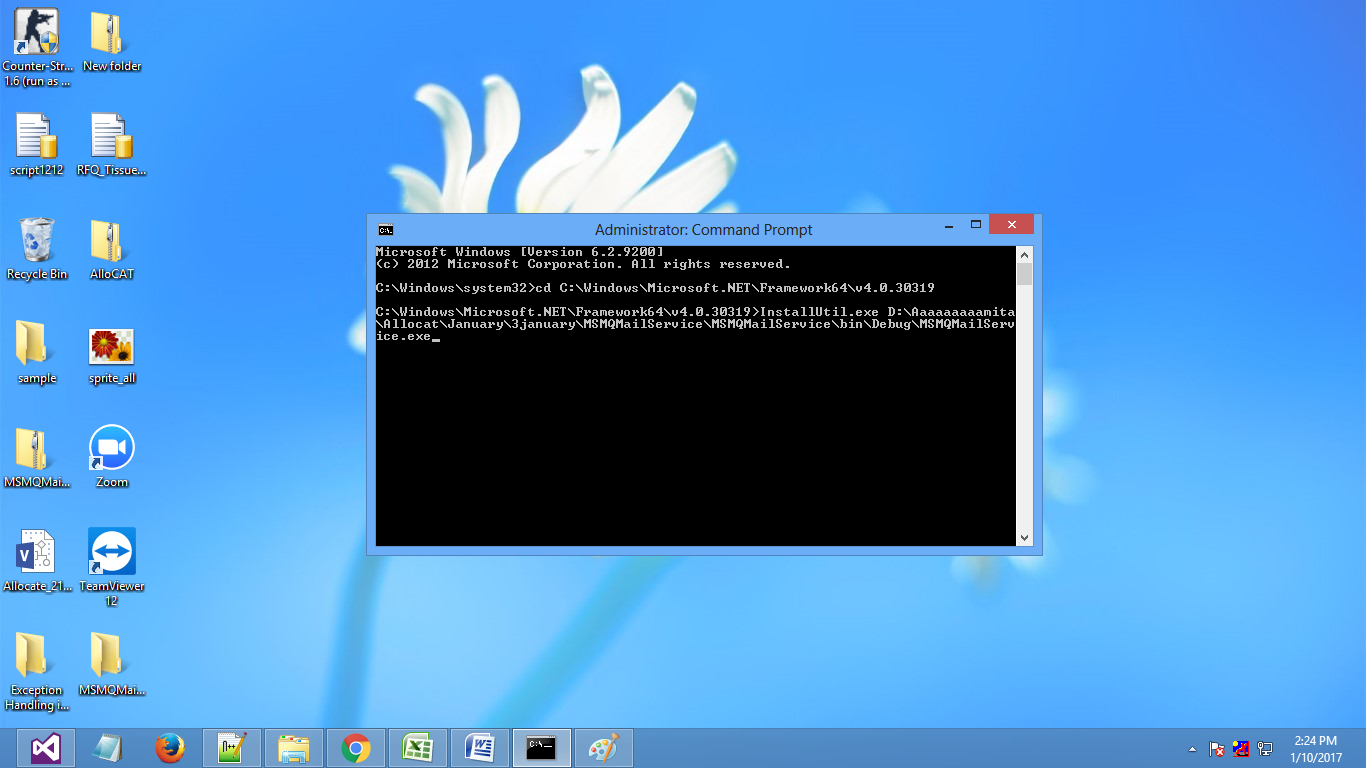


**Step2-** Now writes cd C:\Windows\Microsoft.NET\Framework64\v4.0.30319 command in cmd and press enter. You can see your directory path is changed now.

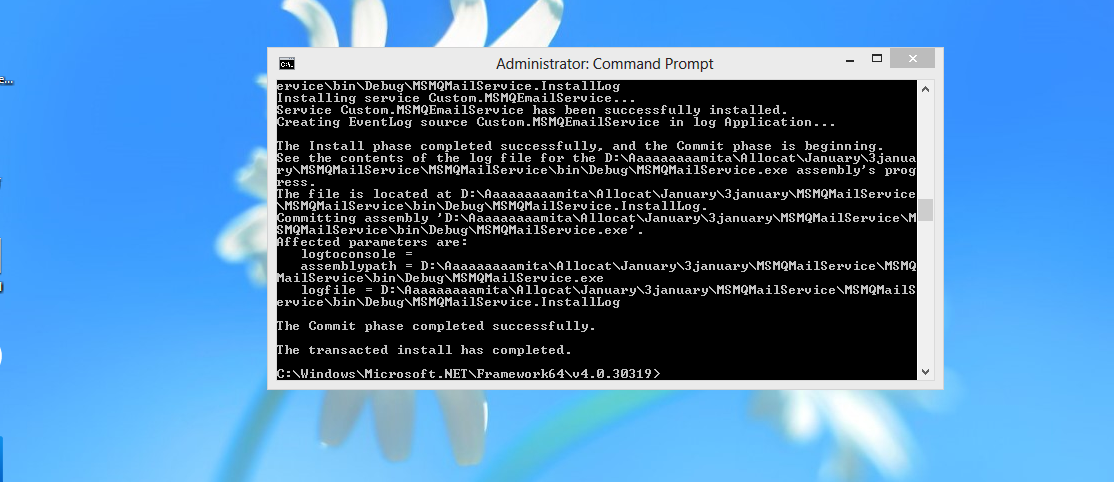
****

**Step3**- Now open your windows service in file explorer, go to bin>debug>MSMQMailService.exe and copy the path like this-“D:\Aaaaaaaaamita\Allocat\January\3january\MSMQMailService\MSMQMailService\bin\Debug\MSMQMailService.exe”

**Step4-** then write installutil.exe D:\Aaaaaaaaamita\Allocat\January\3january\MSMQMailService\MSMQMailService\bin\Debug\MSMQMailService.exe .

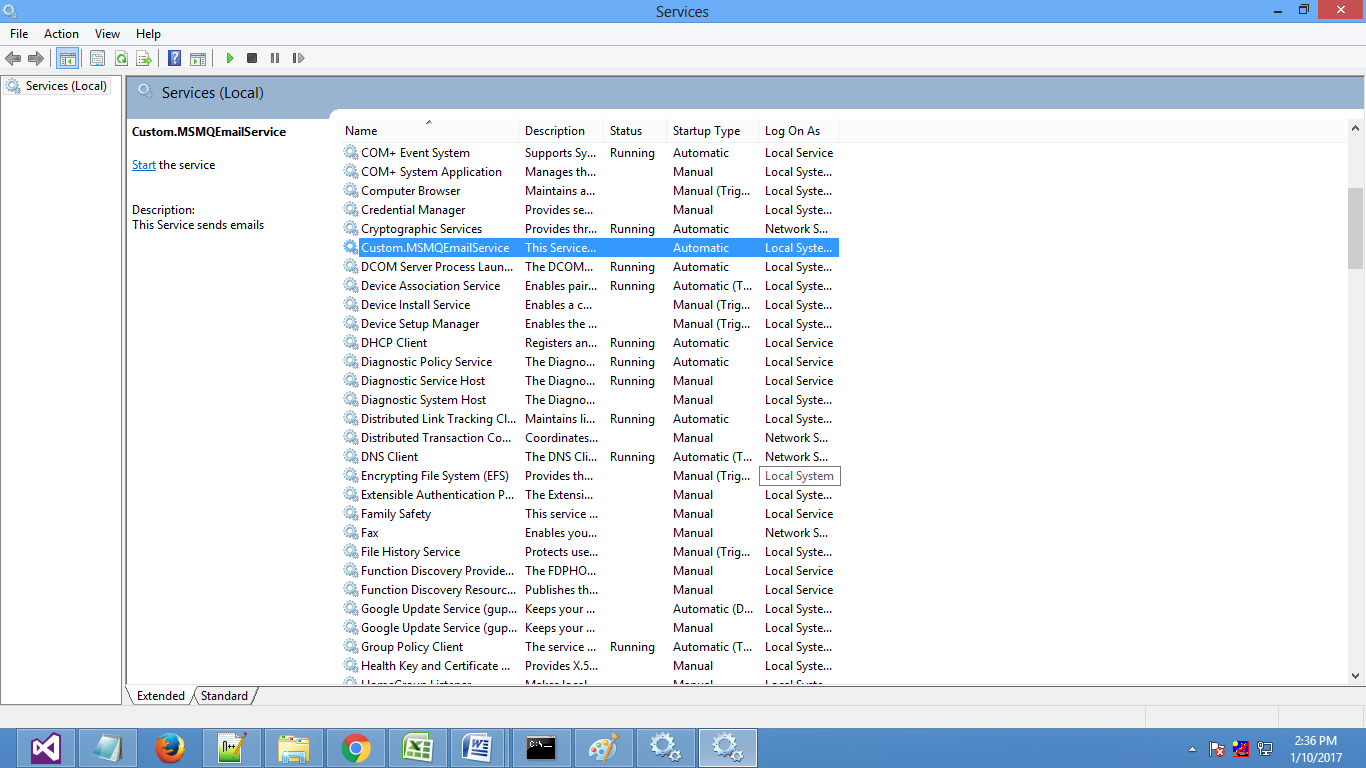
****

**Step5-** Pressenter. Now you can see installing process of windows service in your machine.

****

**Step6-** windows service is installed in your system successfully. Now open services and check your windows service in services.Open services by-

1. Press windows+R
2. Write services.msc



Step7- To uninstall windows service follows the command.

**installutil.exe –u D:\Aaaaaaaaamita\Allocat\January\3january\MSMQMailService\MSMQMailService\bin\Debug\MSMQMailService.exe**

**Step-8** Sometimes if installation is done but it wont work, in that case use this command to delete in command prompt. write –

**sc delete “Custom.MSMQEMailService” and press enter.**

1. **Netmester.Mail.Serialization** – used for serialization
2. **MSMQMailService** – This Service sends email after retrieving from message queue

Things which are important. Don’t change-

**App.config –**

<add key="QueueName" value =".\Private$\EmailQueue"></add>

Please don’t change the Name Of EMailQueue. If you will change “EmailQueue” name then you have to change it in the test application also.

<mailSettings>

<smtp deliveryMethod="Network" from="xx@gmail.com ">

<network defaultCredentials="false" enableSsl="true" port="587" host="SMTP.SENDGRID.NET" userName="azure\_e1993754403deeafed749635d75e8e77@azure.com" password="Allocat@12345678@"/>

</smtp>

</mailSettings>

1. MailSenderLibrary – Sends emails to Message queue-

This library takes whole email body from test application via method **QueueMessage(); and sends to** MSMQMailService

**Test Application-**

I have created an console application.

In **the App config** file**-** I have added

<add key="EmailTurnOn" value="true"/>

<add key="EmailTo" value="amita.sachan@returnonweb.com" />

<add key="EmailCC" value="sindhu.t@returnonweb.com" />

<add key ="EmailBcc" value="sindhu.t@returnonweb.com"/>

<!--Email seetings for exceptions -->

<add key ="FailureEmailTo" value="amita.sachan@returnonweb.com" />

<add key ="FailureEmailCC" value="sindhu.t@returnonweb.com" />

<add key ="FailureEmailBcc" value="sindhu.t@returnonweb.com"/>

<add key ="EmailFrom" value="sachan.amita33@gmail.com" />

Under <EmailSettings>

**configurationReader.cs**

**First I declared all variables.**

public readonly string EmailTurnOn;

public readonly string EmailTo;

public readonly string EmailCC;

public readonly string EmailBcc;

public readonly string FailureEmailTo;

public readonly string FailureEmailCC;

public readonly string FailureEmailBcc;

public readonly string EmailFrom;

**then I added**

var emailSection = (System.Collections.Specialized.NameValueCollection)ConfigurationManager.GetSection("EmailSettings");

this syntax reads all keys and values which are in app.config file. Then I have stored all values in these variables-

EmailTurnOn = emailSection["EmailTurnOn"];

EmailTo = emailSection["EmailTo"]; This is same as in App file

EmailCC = emailSection["EmailCC"];

EmailBcc = emailSection["EmailBcc"];

**Consumer.cs**- Always add these two libraries in reference-

using Microsoft.Practices.EnterpriseLibrary.Logging;

using Microsoft.Practices.EnterpriseLibrary.Common;

I use Use this class which reads all values of configuration reader-

ConfigurationReader configurationReader = new ConfigurationReader(); it reads all values of configurationReaders.

Must use this statement-

EmailSender objEmailSender = new EmailSender(@".\Private$\EmailQueue"); don’t change the parameter in emailsender.

Then write your mail function and message body and then add

**objEmailSender.QueueMessage(objMailMessage);**

**This QueueMessage() belongs to MailSenderLibrary and sends message.**