**Hosting ARM APIs and AgileConnect in Server.**

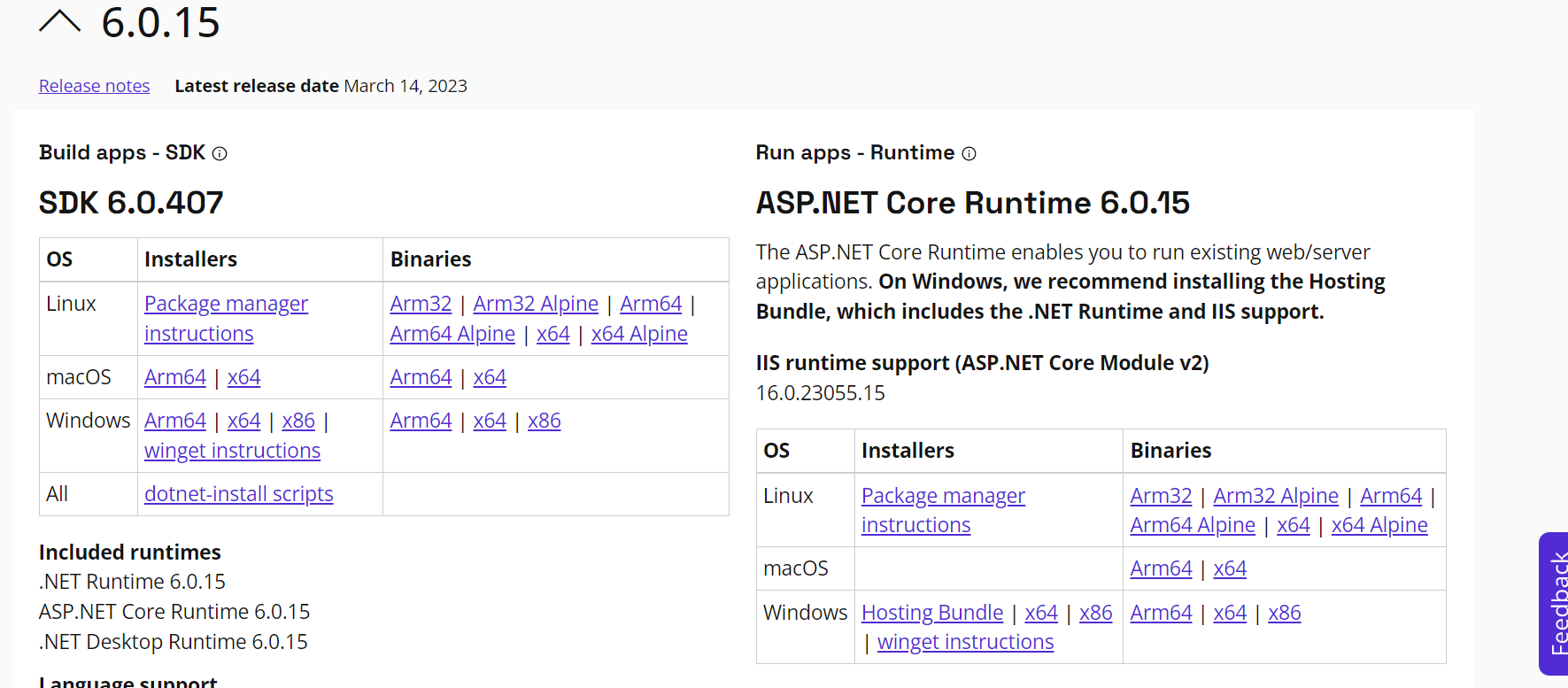
**Pre-requisites:**

1. .NET Core SDK and Hosting bundles.
2. Redis.
3. Rabbitmq with plugin rabbitmq\_delayed\_message\_exchange
4. Postgres.

**Installing .NET Core SDK and Hosting bundles.**

Reference link: <https://dotnet.microsoft.com/en-us/download/dotnet/6.0>

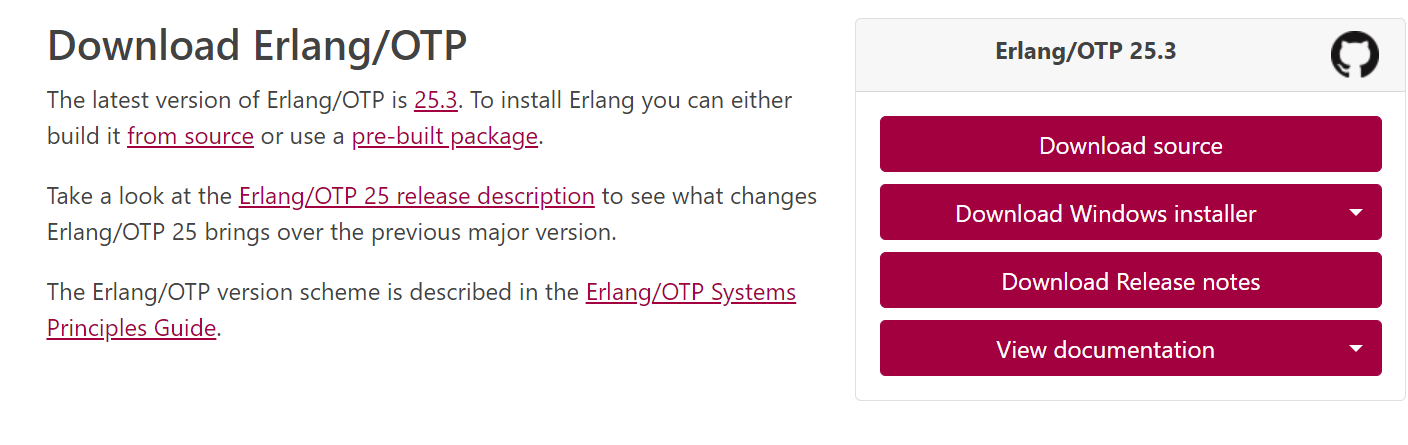
1. Install .NET 6.0 SDK (windows installer).
2. Similarly Install .NET 6.0 hosting bundles for windows.
3. Install ASP.NET Core Runtime (x64) version



**RabbitMQ Installation**

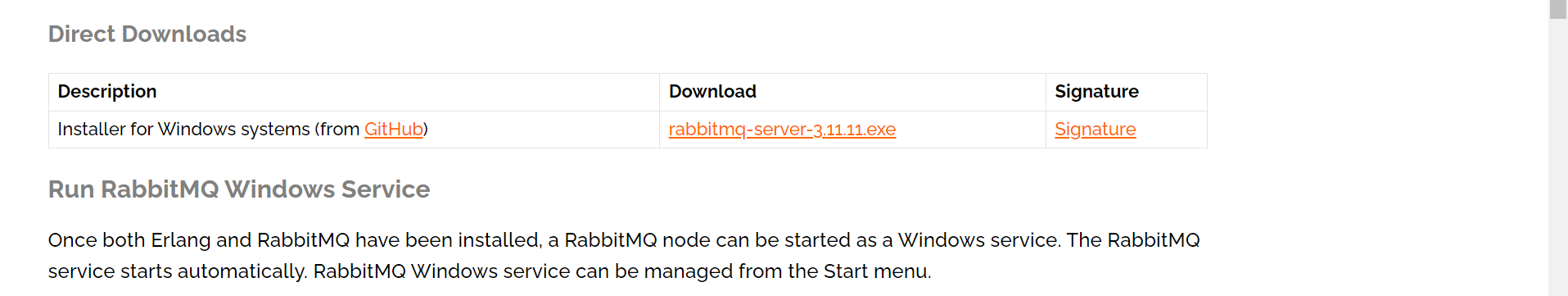
1. Install Erlang:

RabbitMQ is built on the Erlang programming language. Before you can install RabbitMQ, you need to install Erlang. Make sure the Erlang version is compatible with Rabbitmq



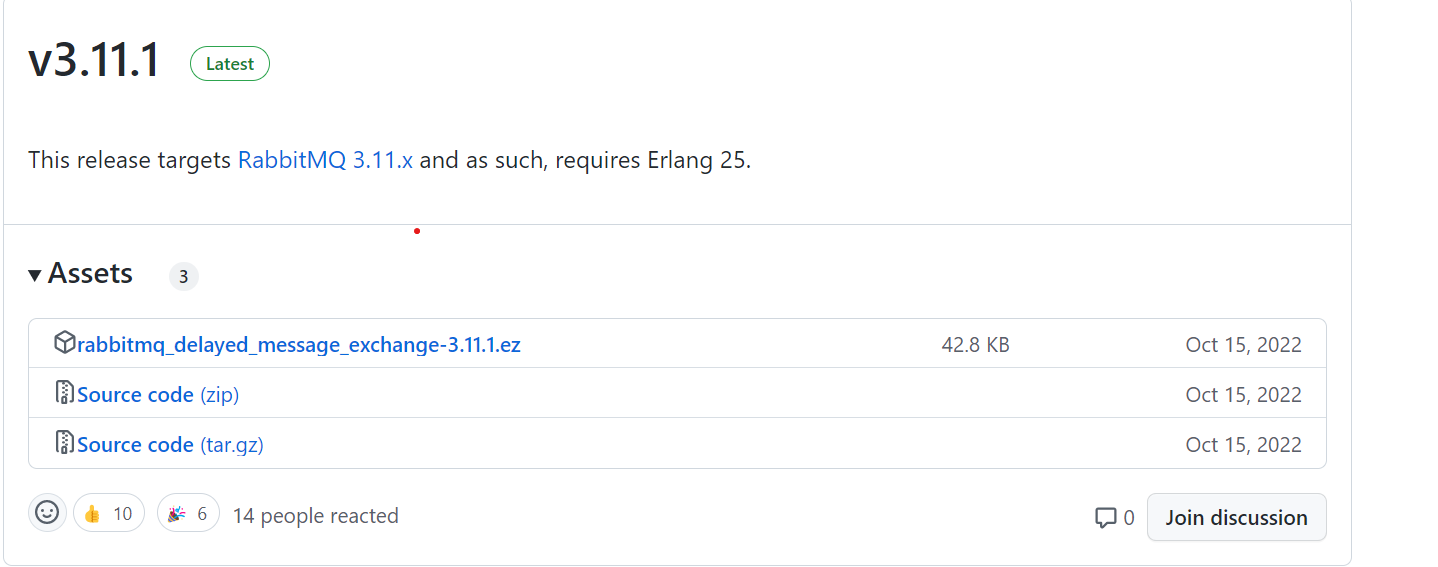
1. Install RabbitMQ:

Once you have installed Erlang, you can download the latest version of RabbitMQ from the official website. Install  [**rabbitmq-server-3.11.7.exe**](https://github.com/rabbitmq/rabbitmq-server/releases/download/v3.11.11/rabbitmq-server-3.11.11.exe) **. OR ABOVE**



1. Download The Delayed Message Exchange plugin : [**https://github.com/rabbitmq/rabbitmq-delayed-message-exchange/releases**](https://github.com/rabbitmq/rabbitmq-delayed-message-exchange/releases)**.**

Download the .ez file from above URL and copy it to RabbitMQ -> Plugins folder in your installation folder.



1. Enable the RabbitMQ Delayed Message Exchange plugin:

The Delayed Message Exchange plugin allows you to delay the delivery of messages to a queue. To enable the plugin, run the following command from the command line from the RabbitMQ’s sbin folder :

**rabbitmq-plugins enable rabbitmq\_delayed\_message\_exchange**

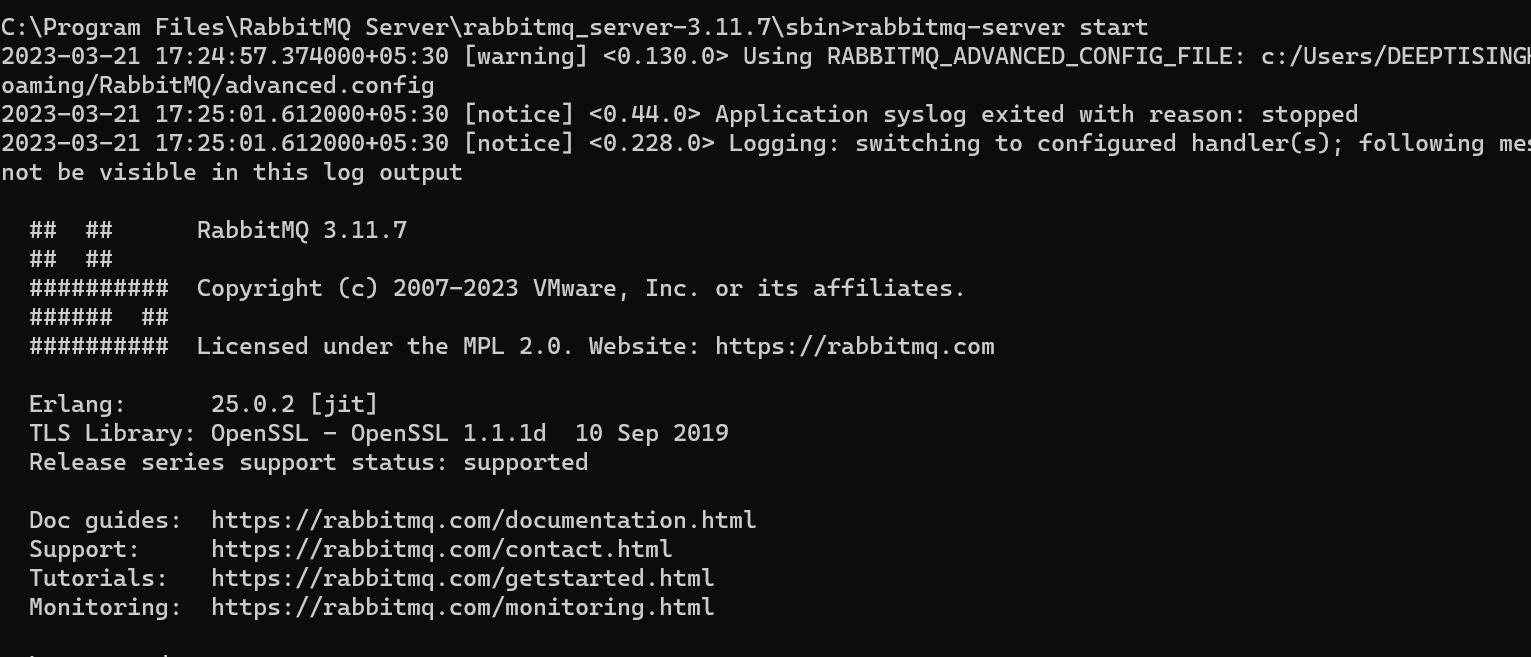
1. Enable management plugin using following command in above cmd:

**rabbitmq-plugins enable rabbitmq\_management**

**Test RabbitMQ installation:**

1. To test if RMQ (RabbitMQ) is successfully installed, you can perform the following steps: Start the RabbitMQ server from the service manager manually or from cmd

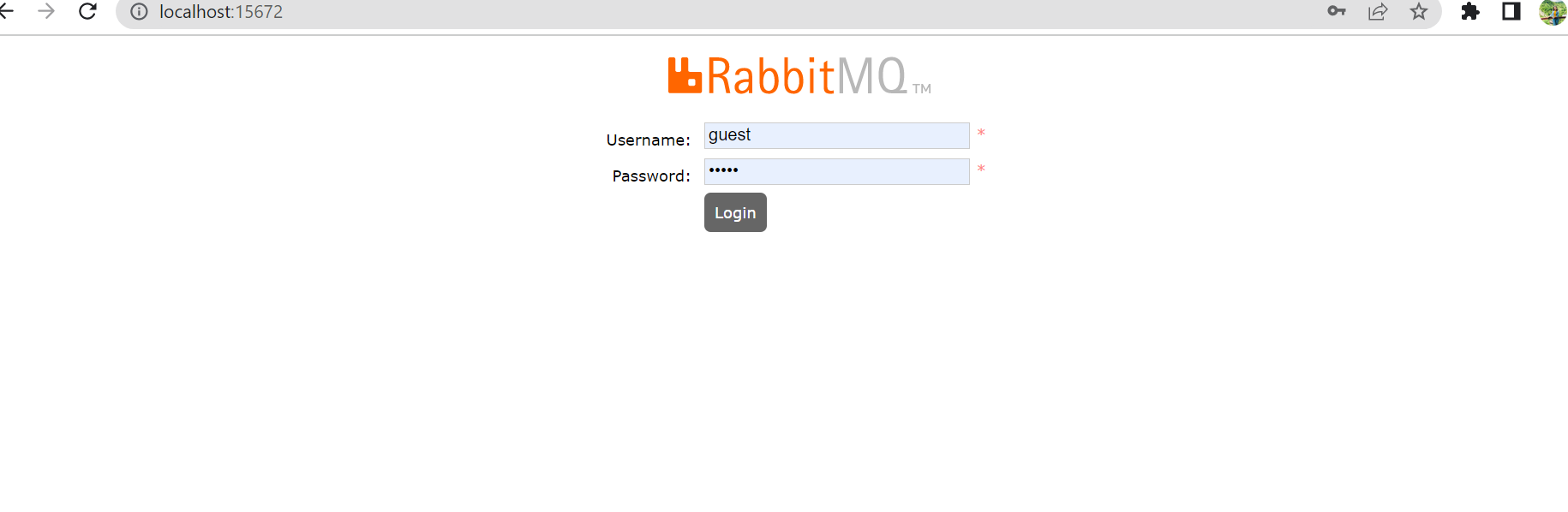
Navigate to the RabbitMQ sbin directory. This is typically located at C:\Program Files\RabbitMQ Server\rabbitmq\_server\_version\sbin.



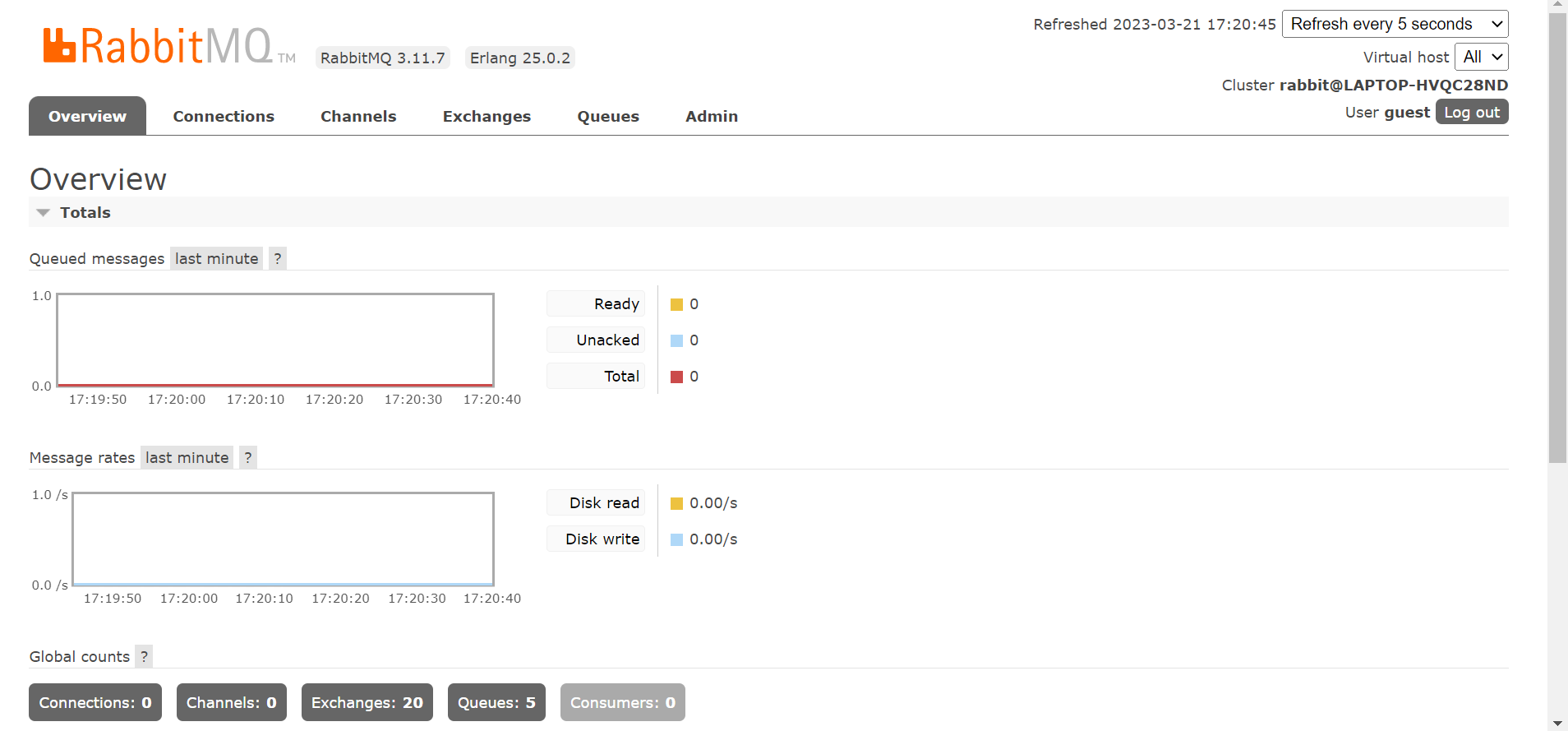
1. Access the RabbitMQ web management interface: [**http://localhost:15672/**](http://localhost:15672/)
2. If RabbitMQ is successfully installed, you should be able to access the web management interface by opening a web browser and navigating to the above URL.
3. Log in to the web management interface:
4. By default, the web management interface requires a username and password to access. The default credentials are:

username: guest

password: guest



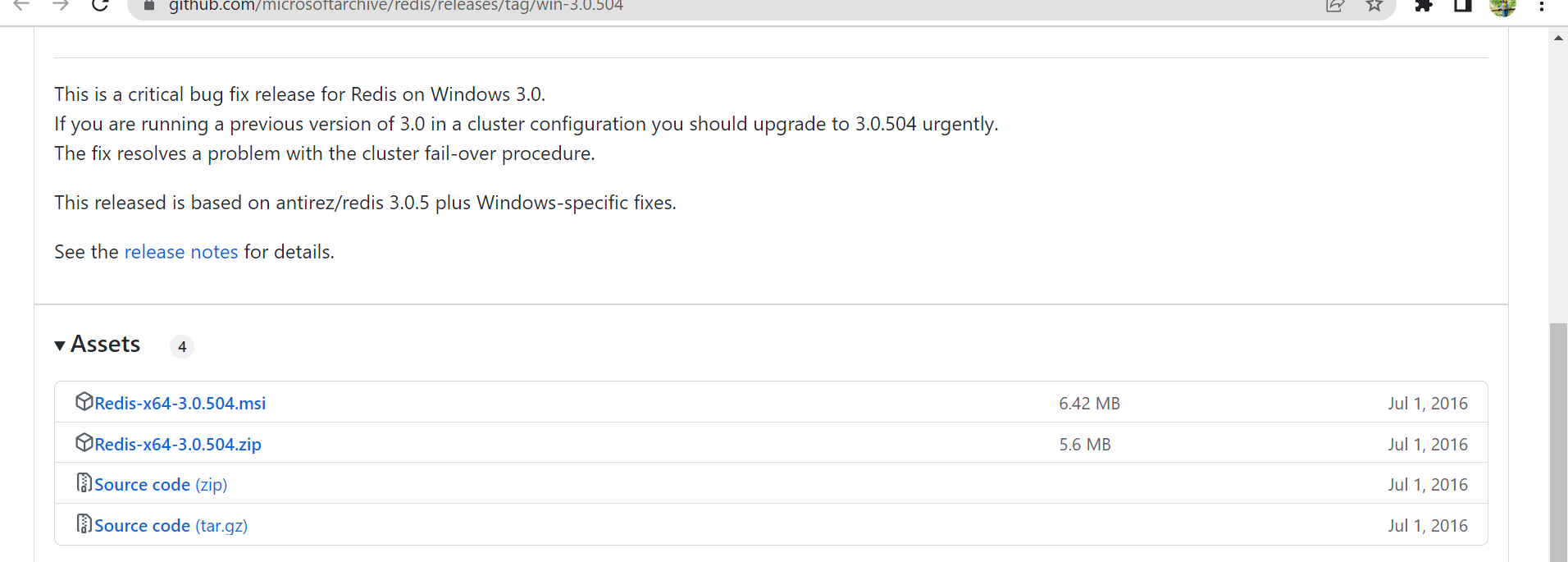
After entering the correct credentials, you should be able to see the RabbitMQ dashboard.



**Redis Installation**

Install Redis Window Installer from :

**https://github.com/microsoftarchive/redis/releases/tag/win-3.0.504**



**Test RabbitMq installation:**

Once you have installed Redis, you can test it to make sure it is working correctly. Here are the steps to test Redis on Windows:

**Postgres Installation**

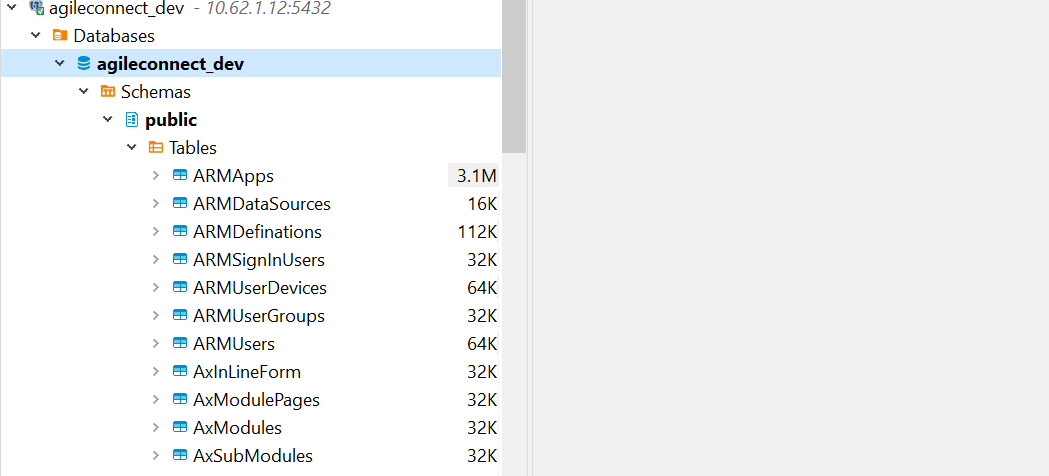
Install Postgres from below link:

**https://www.postgresql.org/download/windows/**

**Next, we need to import agileconnect db, please import db from the following folder.**

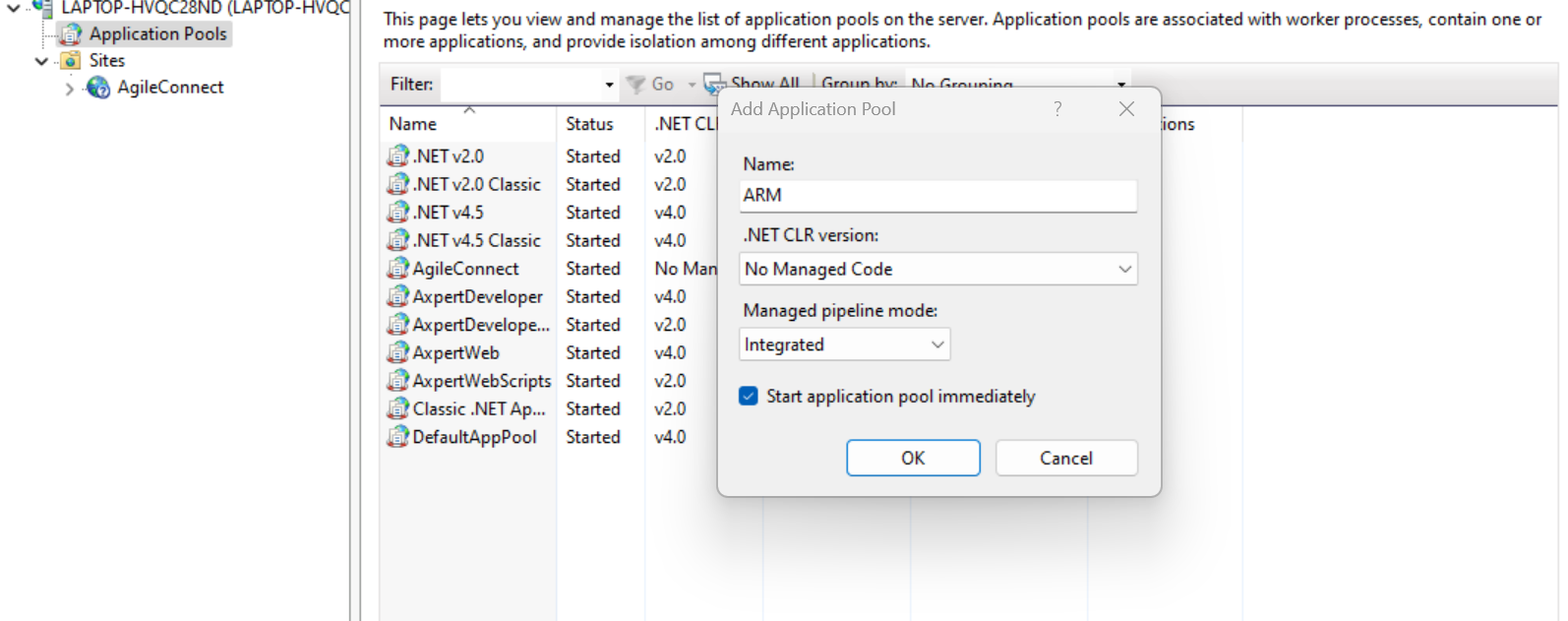
AgileConnect.sql

**After Importing successfully, you can see following schemas and tables:**



**Hosting ARM APIs in IIS**

1. Set up an application pool: To set up an application pool with no managed code, open the IIS Manager, right-click on the "Application Pools" folder, and select "Add Application Pool." Set the .NET Framework version to "No Managed Code" and other settings as required.



1. Create an app : After creating an application pool, you need to create an application to host your app. To create an application, right-click on the site you created in step 2, select "Add Application," and fill out the required fields, such as the application name and physical path.
2. Extract the files from ZIP to above folder.

**appsettings.json configuration:**

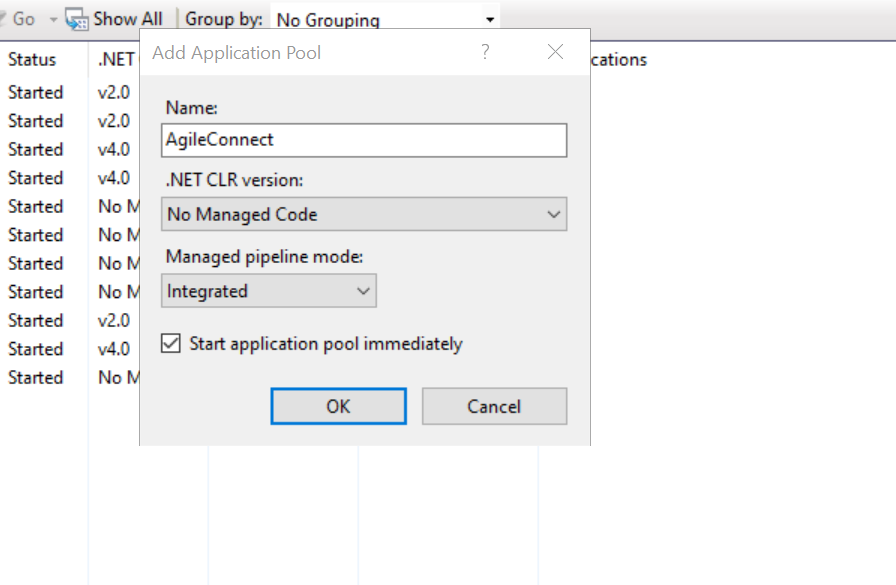
Configure appsettings.json file with proper credentials

1. Test the API:  
   Call the GET URL *<apiurl>/api/v1/ARMAppStatus* in browser or postman

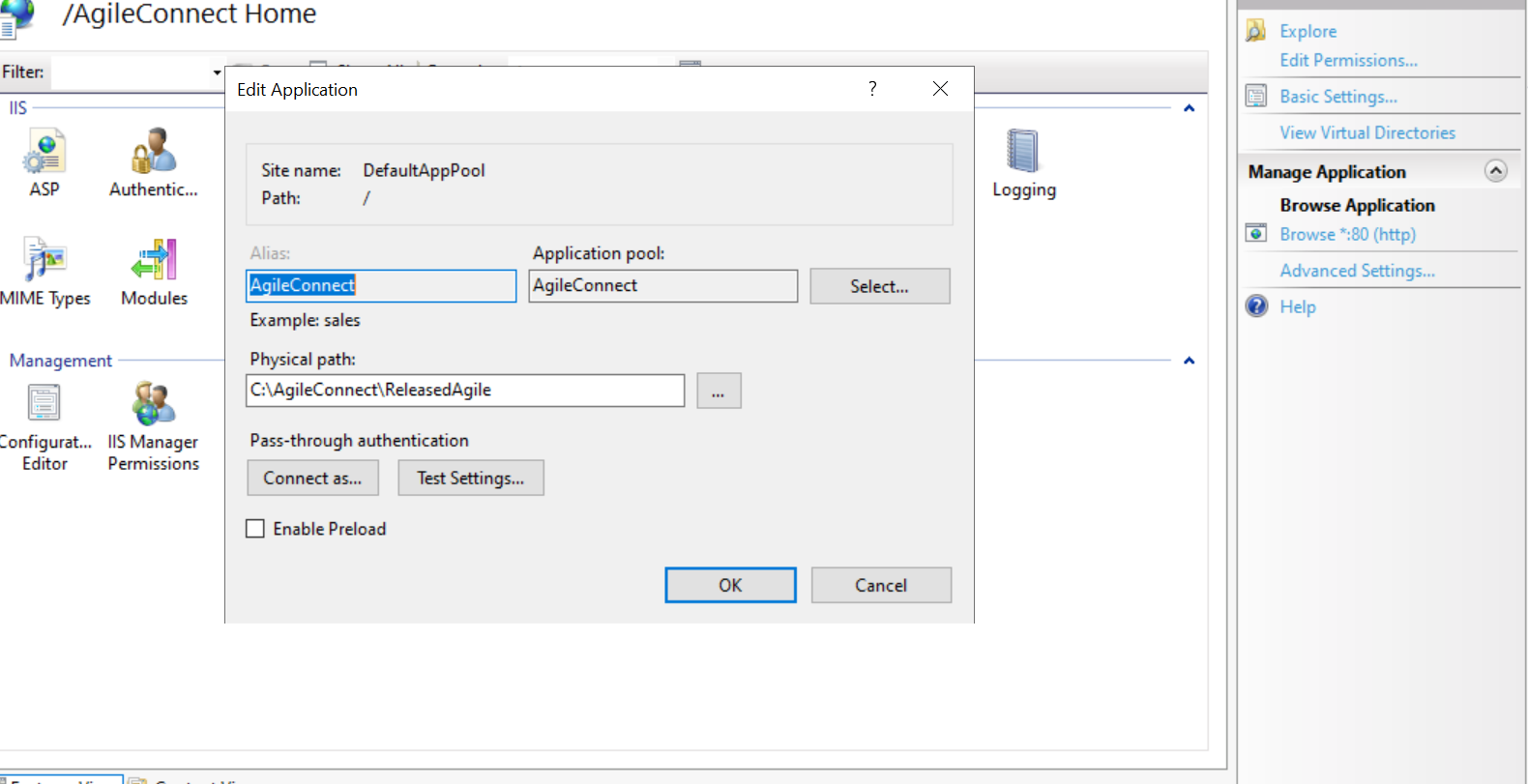
Success Result:  
{"result":{"message":"Application is running successfully","success":true}}

**Hosting AgileConnect in IIS (Optional. Needed for PEG and AxpertFlutter)**

1. Set up an application pool: To set up an application pool with no managed code, open the IIS Manager, right-click on the "Application Pools" folder, and select "Add Application Pool." Set the .NET Framework version to "No Managed Code" and other settings as required.

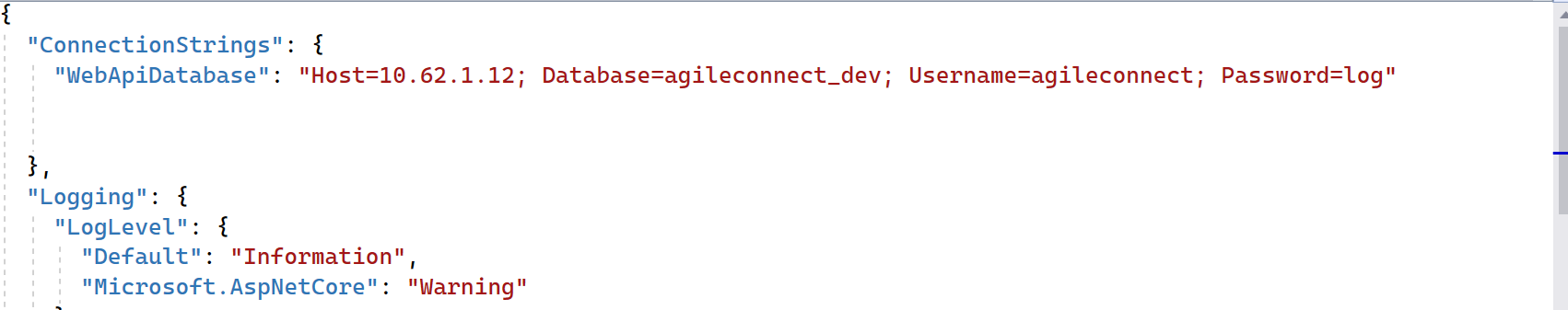


1. Create an app : After creating an application pool, you need to create an application to host your app. To create an application, right-click on the site you created in step 2, select "Add Application," and fill out the required fields, such as the application name and physical path.



1. Extract the files from the ZIP to the above application folder.
2. Similarly we need to update appsettings.json also.

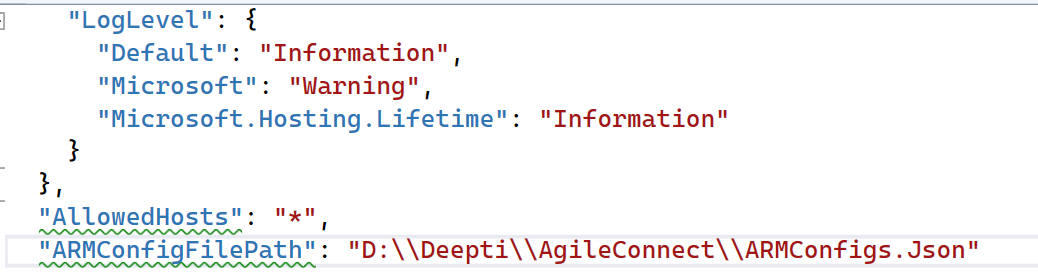
**change connectionstrings :**



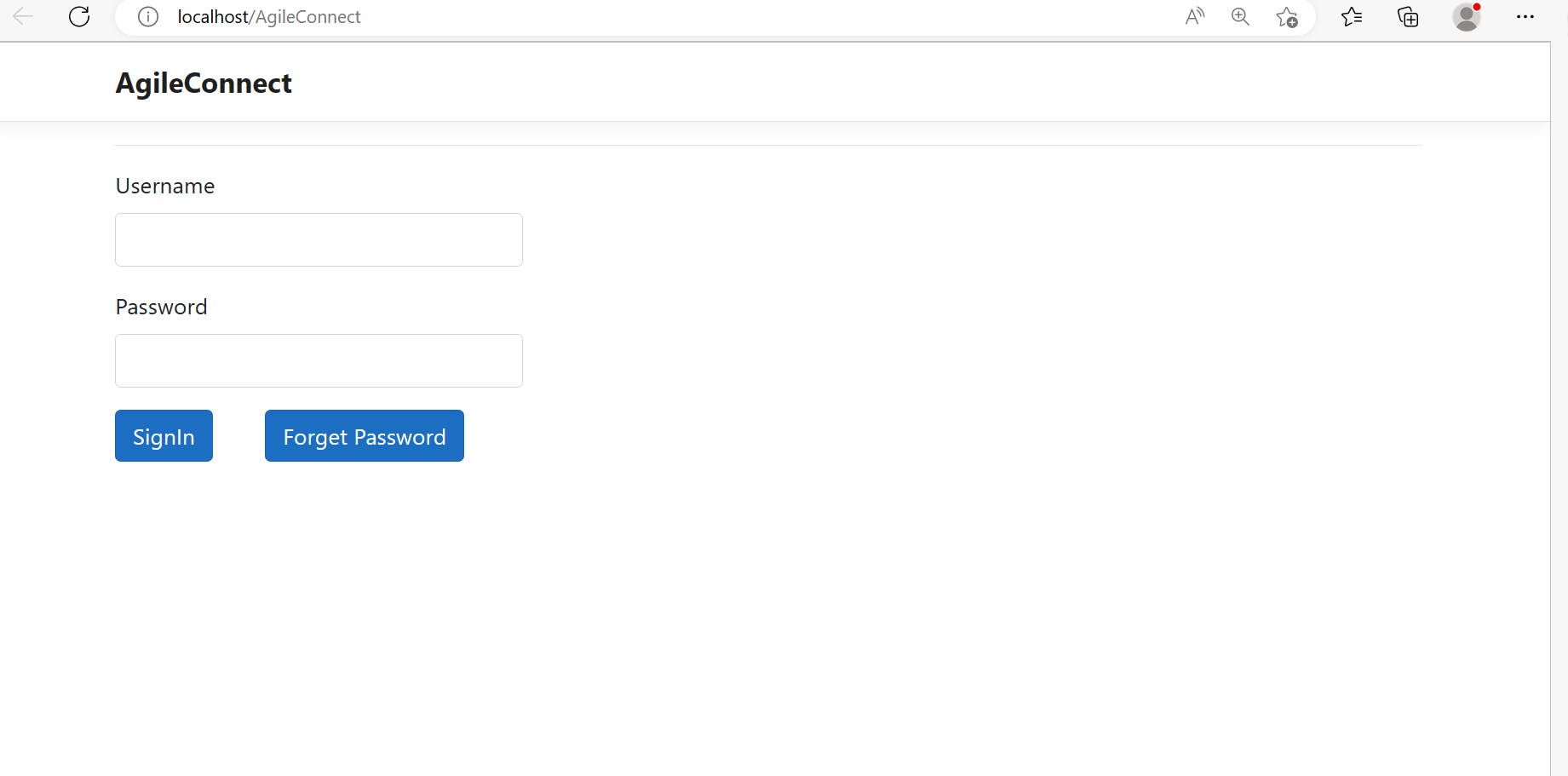
**Also, change Redis configuration accordingly from here:**



**Configure path for ARMConfigFilePath:**



1. Test the app: Finally, you should test your app to ensure it is running correctly. Open a web browser and navigate to the site's URL to access your app.

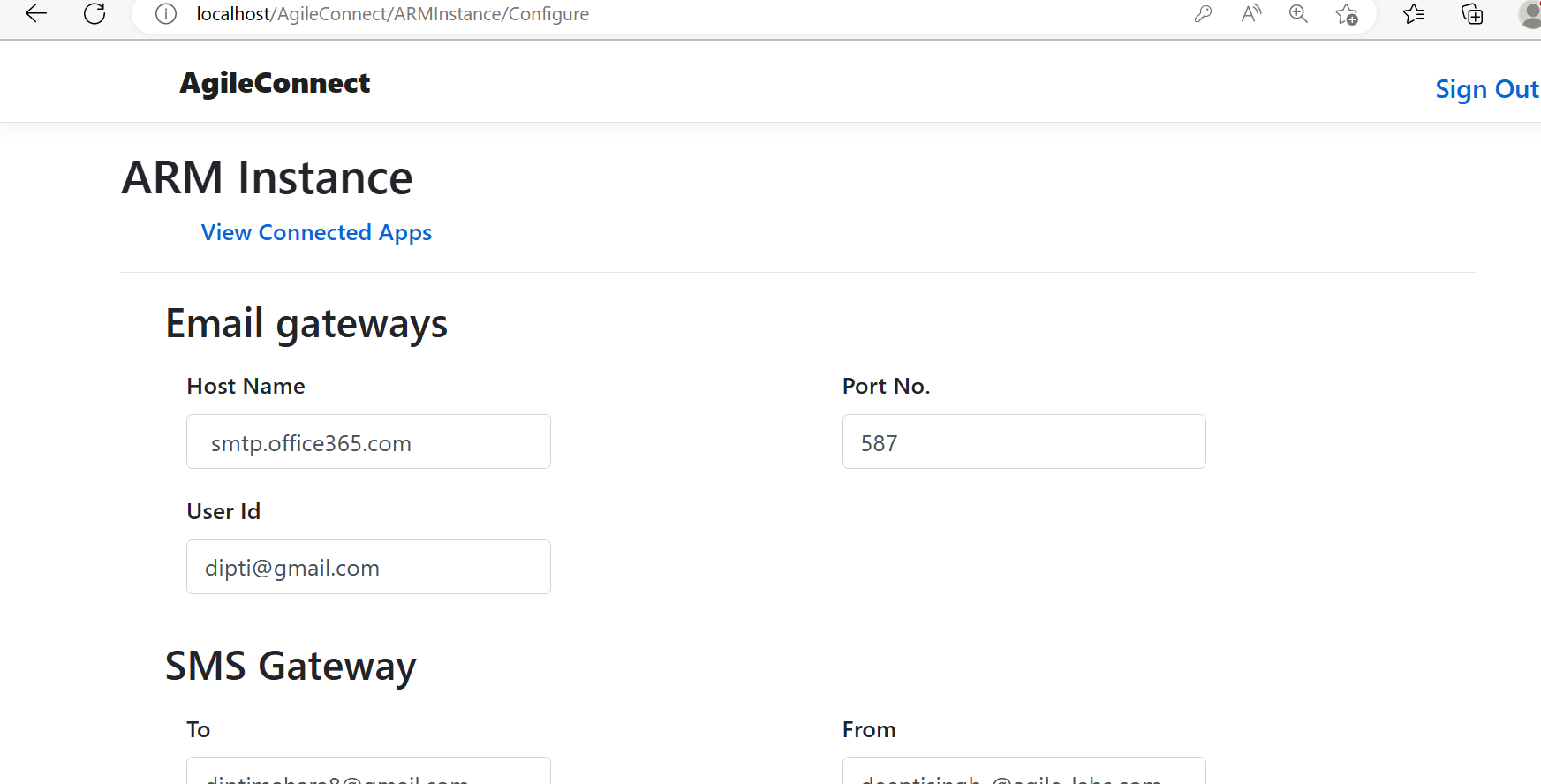


The Username and password of the app is :

Username : admin,

Password : agile.

After Successful login, you will be redirect to ARMInstance Configuration Page :



You can change your Username password from the same ARMInstance page in Admin Details Section:

