WorkshopPLUS

Microsoft Azure Infrastructure as a Service (IaaS)

Introduction to Azure Role Based Access Control (RBAC)

Student Lab Manual

V1.8, June 1, 2016

Information in this document is subject to change without notice. The example companies, organizations, products, people, and events depicted herein are fictitious. No association with any real company, organization, product, person or event is intended or should be inferred. Complying with all applicable copyright laws is the responsibility of the user. Without limiting the rights under copyright, no part of this document may be reproduced, stored in or introduced into a retrieval system, or transmitted in any form or by any means (electronic, mechanical, photocopying, recording, or otherwise), or for any purpose, without the express written permission of Microsoft Corporation.

Microsoft may have patents, patent applications, trademarked, copyrights, or other intellectual property rights covering subject matter in this document. Except as expressly provided in any written license agreement from Microsoft, the furnishing of this document does not give you any license to these patents, trademarks, copyrights, or other intellectual property.

© 2014 Microsoft Corporation. All rights reserved.

Microsoft, MS-DOS, MS, Windows, Windows NT, MSDN, Active Directory, BizTalk, SQL Server, SharePoint, Outlook, PowerPoint, FrontPage, Visual Basic, Visual C++, Visual J++, Visual InterDev, Visual SourceSafe, Visual C#, Visual J#,  and Visual Studio are either registered trademarks or trademarks of Microsoft Corporation in the U.S.A. and/or other countries.

Other product and company names herein may be the trademarks of their respective owners.

Contents

[Introduction to Azure Role Based Access Control (RBAC) 4](#_Toc452559884)

[Prerequisites 4](#_Toc452559885)

[Exercise 1: Adding users to the default AAD tenant 5](#_Toc452559886)

[Task 1 – Adding users and groups 5](#_Toc452559887)

[Task 2 - Creating new users in your AAD tenant 6](#_Toc452559888)

[Exercise 2: Deploy a Web application using ARM 10](#_Toc452559889)

[Task 1 – Create the Azure Resource Group project 10](#_Toc452559890)

[Task 2 – Deploy to Azure 10](#_Toc452559891)

[Task 3 – Confirm Deployment of Web App in Azure 14](#_Toc452559892)

[Exercise 3: Setup Users using RBAC 18](#_Toc452559893)

[Exercise 4: Test Users using RBAC permissions 24](#_Toc452559894)

[Task 1 – Confirm AAD Admin group permissions 24](#_Toc452559895)

[Task 2 – Confirm AAD SQL Dev group permissions 24](#_Toc452559896)

[Task 3 – Confirm AAD Web Dev permissions 25](#_Toc452559897)

[Task 4 – Confirm AAD Web Reader permissions 25](#_Toc452559898)

# Introduction to Azure Role Based Access Control (RBAC)

In this lab, you will setup an environment that will allow you to test RBAC for users within your own Azure Active Directory (AAD) tenant.

You'll learn:

* How to add users and groups to the default AAD directory
* How to deploy a pre-existing web application to an Azure Resource Group using Visual Studio template deployment
* How to setup RBAC for the users in your AAD tenant

### Prerequisites

The following is required to complete this hands-on lab:

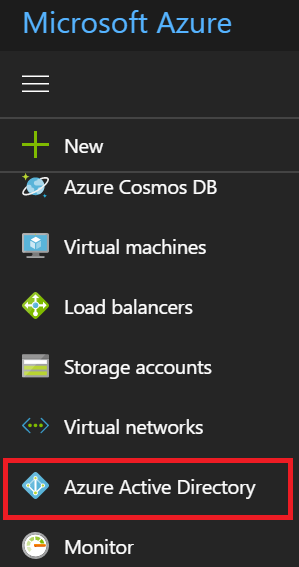
* Microsoft V[isual Studio 2015 Professional or Enterprise with Update 1](http://www.microsoft.com/visualstudio/)
* [Microsoft Azure SDK for .NET (VS 2015) - 2.](http://www.microsoft.com/windowsazure/sdk/)9.1 or later
* A Microsoft Azure subscription
* **A pre-existing Azure ARM storage account**

## Exercise 1: Adding users to the default AAD tenant

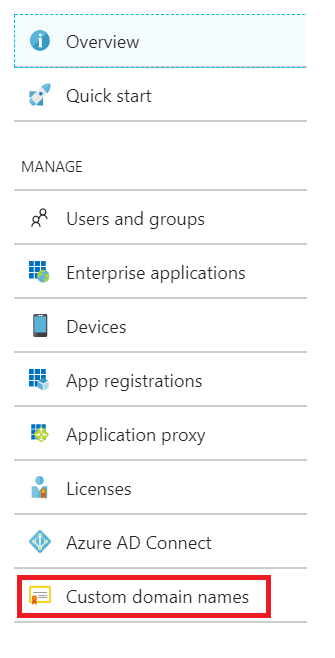
For each Azure subscription, there is a default Azure Active Directory tenant. Depending on who created your subscription, you may or may not have access to add users and groups the tenant. This exercise assumes you have the appropriate access. Prior to adding a new user, we will need to find out the name of our Azure AD tenant. This is required during user creation in a non custom domain named tenant i.e. a domain name with a .onmicrosoft.com suffix.

### Task 1 – Adding users and groups

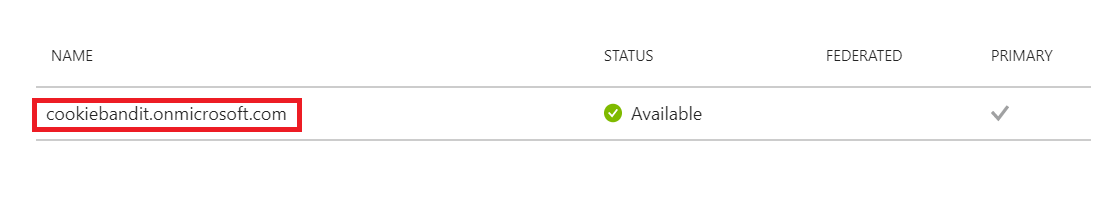
1. Sign in to the Azure Portal at <https://portal.azure.com>.
2. Select Azure Active Directory from the left pane.



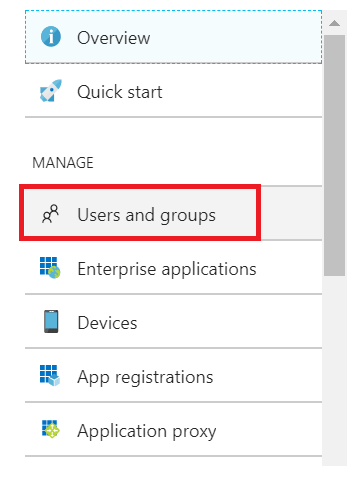
1. Select Custom Domain Names.



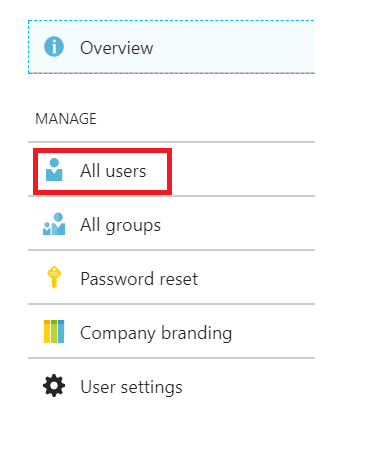
1. Record the name of your Azure AD tenant, in this case cookiebandit.onmicrosoft.com



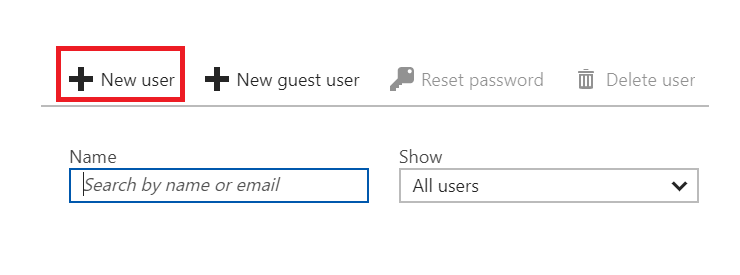
1. Click Users and groups



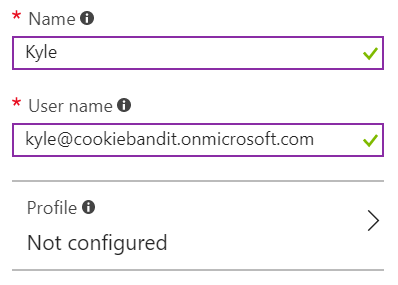
1. Select All users.



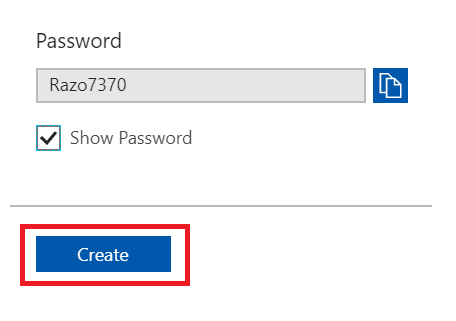
1. In the right-hand pane, select New user.



1. Type in a name in the Name text box and type in a username in the Username text box. The format of the username should be an email address, starting with the name used in the Name text box and ending with the Azure AD tenant name you recorded in step 4 e.g. [kyle@cookiebandit.onmicrosoft.com](mailto:kyle@cookiebandit.onmicrosoft.com)



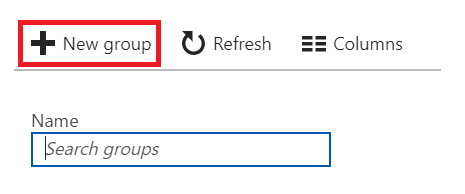
1. Scroll down to the Password section, tick the Show Password tick box, record the password and then click Create.



1. You need to create 4 more users, remember to write the generated passwords down. If you ever log in as these users, you will need to change their password.

|  |  |
| --- | --- |
| Name | Role |
| angie | User |
| fred | User |
| george | User |
| alex | User |

1. You now need to create a few groups and then put your users into specific groups. You create groups the same way you created users except that you will select the All groups menu item and then select New group.



Add the following groups with users. The **Membership** **type** will be **Assigned**:

|  |  |  |
| --- | --- | --- |
| Group | Description | Users/Groups |
| Admins | Resource Group administrators – full control | kyle |
| Web Dev | Web Developers – access web as a contributor but cannot access SQL | george, alex, Admins |
| Web Reader | Web readers can only read web app information | fred |
| SQL Dev | SQL Dev can access databases, make changes, add tables etc | angie, alex, Admins |

## Exercise 2: Deploy a Web application using ARM

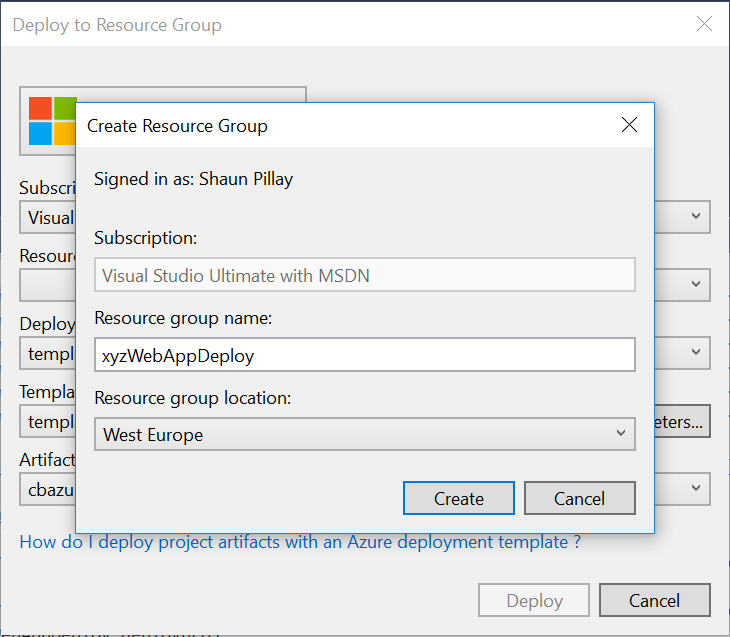
### Task 1 – Create the Azure Resource Group project

For this exercise, you will use a Visual Studio project that has an Azure Resource Group deployment project and a web application that gets deployed with the template. (This is the source code for the ARM lab performed earlier).

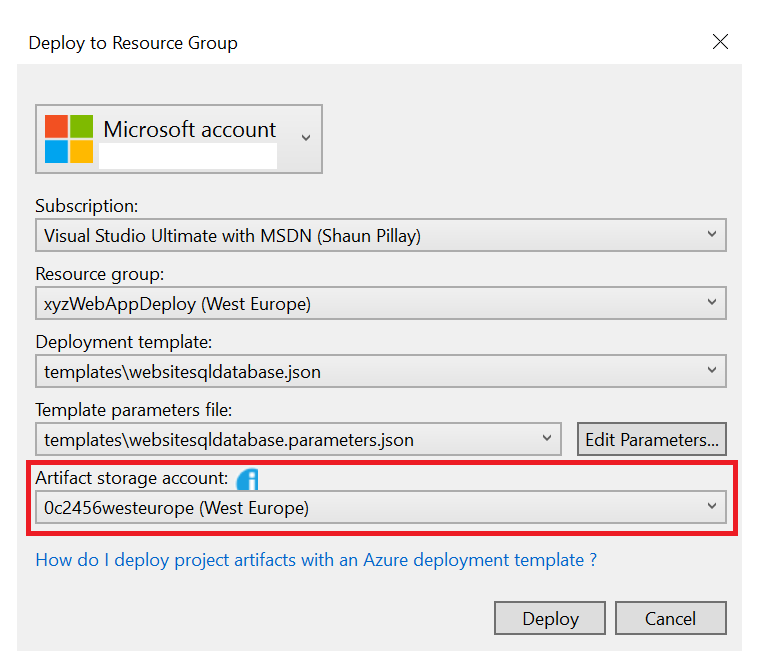
1. Open Visual Studio as an Administrator.
2. Select **File | Open | Project/Solution** and then browse to **C:\AzureIaaSWS\M6 - Azure Identity\Labs\IntroToRBAC\Source\WebAppDeploy\WebAppDeploy.sln** and open the project.
3. Right click on the **WebAppDeploy** solution in the Solution Explorer, and select **Build**.

### Task 2 – Deploy to Azure

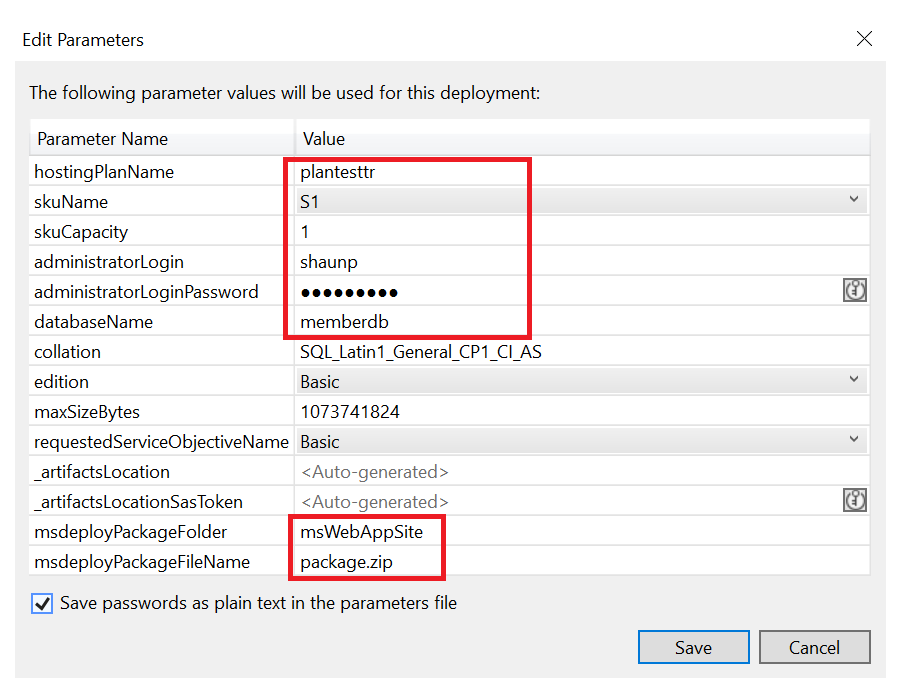
1. Right click on the **WebAppDeploy** project in the Solution Explorer and select **Deploy | New Deployment.**
2. In the *Deploy to Resource Group* dialog box, if you have not already signed in to Azure, click the **Sign In** button to first sign in.
3. Once you are signed in to Azure, select the dropdown in the Resource Group box and create a new, unique resource group name. A dialog box will appear that allows you to select the resource group location also. Then, select the **Create** button.



1. Down at the bottom of the *Deploy to Resource Group* dialog box, you need to select the storage account that the resource template and app will be deployed to prior to Azure setting things up. You need to already have a pre-existing ARM storage account. Select your storage account from the dropdown.



1. Now select the **Edit Parameters** button. This will fill in parameters to be passed to the deployment.



The parameters that **MUST** be set are circled above.

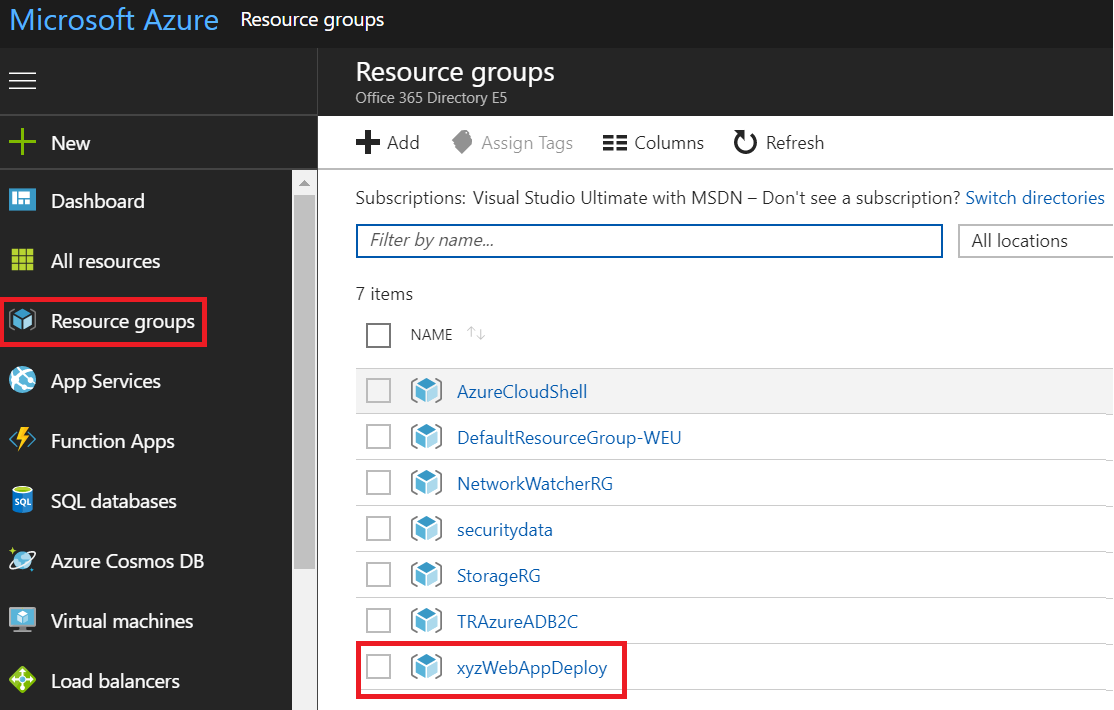
* hostingPlanName – the hosting plan name for the website – if one already exists you can use it
* skuName – Azure Web App tier
* skuCapacity – size of the web machines
* administratorLogin/Password – login and password to the db server
* databaseName – needs to be memberdb for this sample
* msdeployPackageFolder – the **project name** of the web application being deployed. Pay no attention to the fact that it says ‘folder name’.
* msdeployPackageFileName – the name of the package file to be deployed. In our case, the name will be **package.zip**

Make sure you check the ‘**Save Passwords as plain text…’** checkbox and select **Save**.

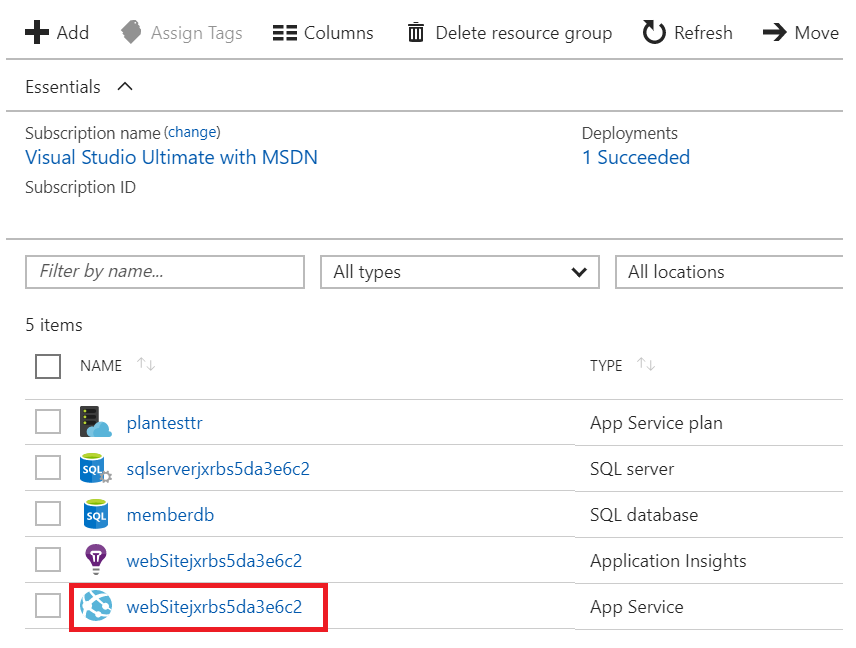
1. Click the **Deploy** button. If the Edit Parameters box pops up again, just make sure you select the Save Passwords checkbox again and select **Save**.

### Task 3 – Confirm Deployment of Web App in Azure

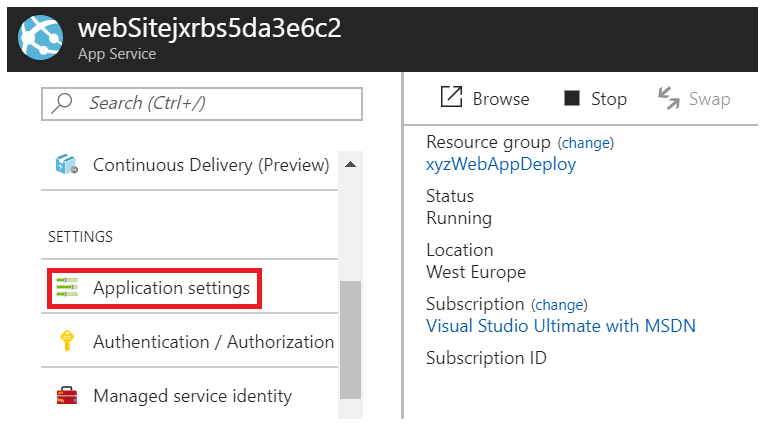
1. Log in to the Azure portal at <https://portal.azure.com>. You will not be able to see resource groups from the current portal.
2. On the left hand side, click the **Resource groups** button. Click on the name of the resource group you created for the deployment.



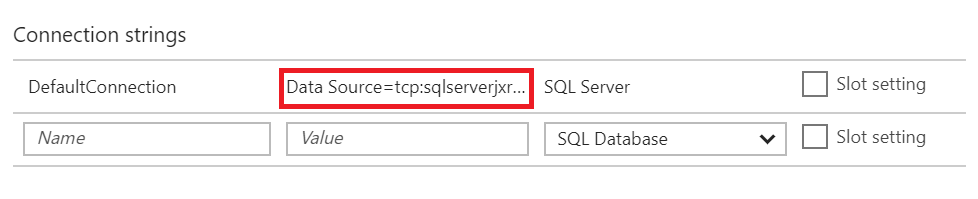
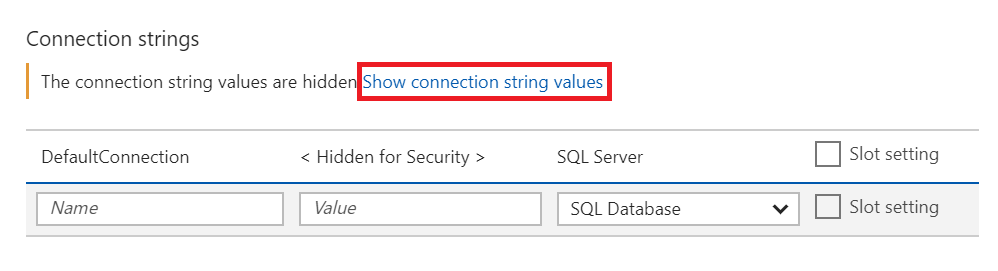
1. Your resource group blade will open. It will display all resources that belong to this resource group. Click on the **web app**. The web app blade will appear.



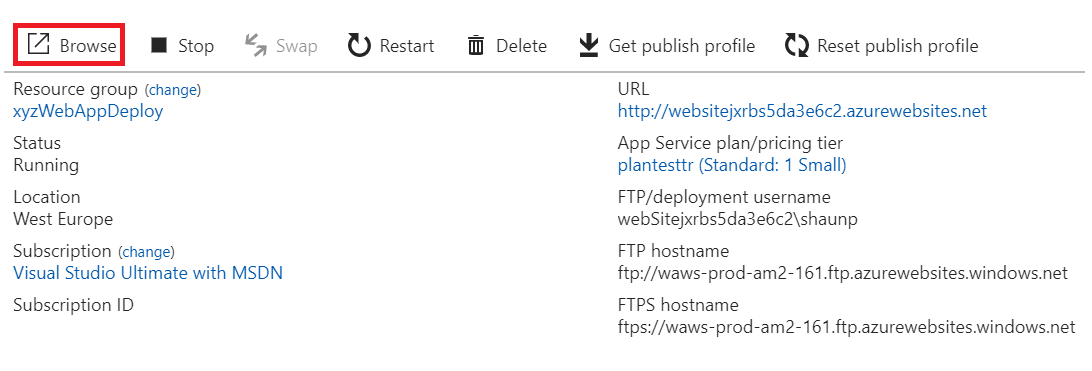
1. In the web app blade, scroll down and click on the **Application settings** link.



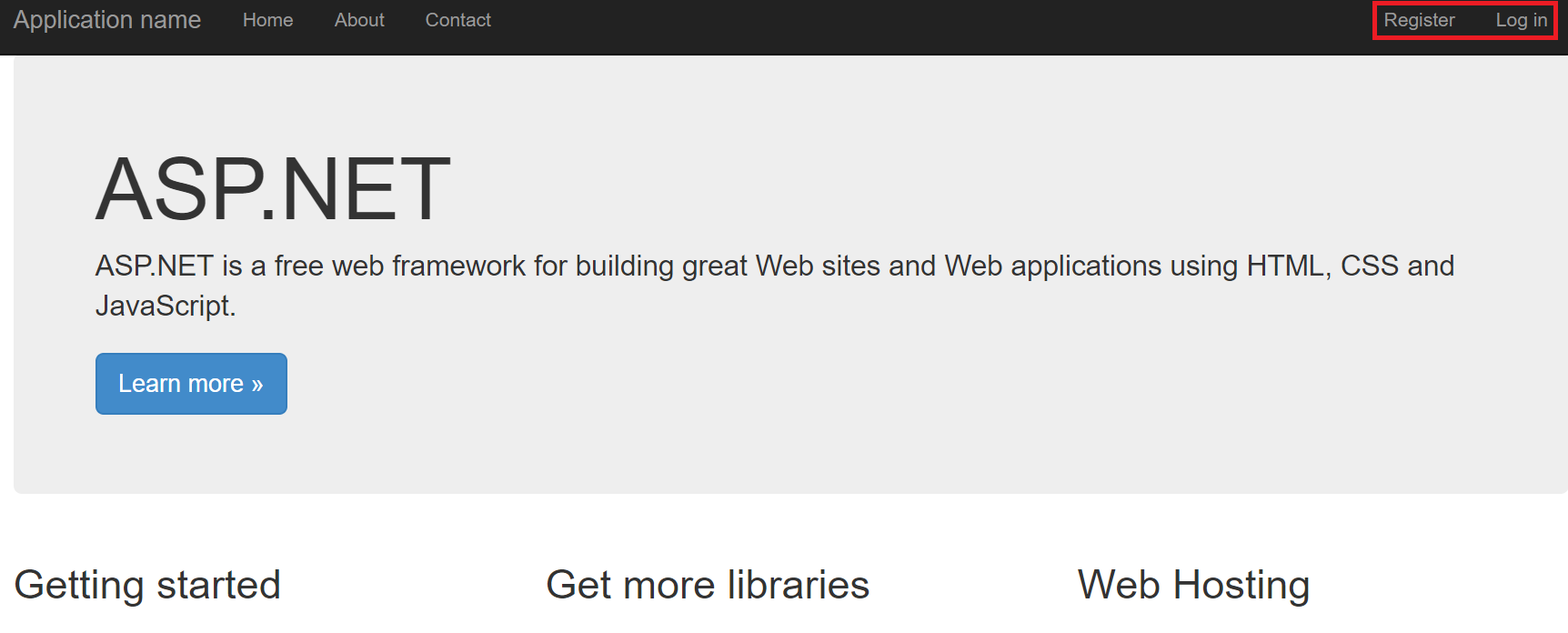
1. In the Application settings blade, scroll down to the Connection Strings setting where you’ll see a link ‘*Show Connection String values’*. Click this and you will be able to confirm that your connection string is set correctly.



1. Close the Web app settings and the Settings blade.
2. To test the application, click on the **Browse** button on the web app blade.



1. Assuming everything deployed correctly, you should see a website similar to the screenshot below. You should also be able to register new users and log in.



## Exercise 3: Setup Users using RBAC

In Exercise 1, you created a few users in your AAD and then placed those users into Groups. What you will do now, is map those Users/Groups to Roles within our resource group.

The settings should match these requirements:

|  |  |  |
| --- | --- | --- |
| RBAC Role | Level | AAD Group |
| Owner | Resource Group | Admins |
| Website Contributor | Web App | Web Dev |
| Reader | Web App | Web Reader |
| SQL DB Contributor | SQL Server | SQL Dev |

As a reference, the selected RBAC roles above have the following capabilities:

**Owner** – Owner can manage everything, including access.

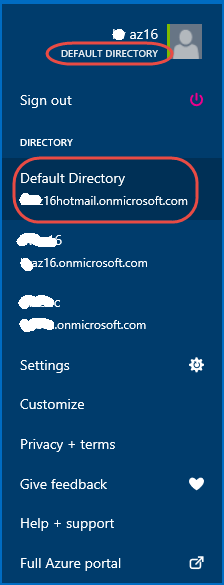
**Website Contributor** – Website Contributor lets you manage websites (but not web plans) but does not allow you to change access to them.

**Reader** – Readers can read information on the resource, but can’t change anything.

**SQL DB Contributor** – Lets you manage SQL databases, but not access to them. Also, you can’t manage their security-related policies or their parent SQL servers.

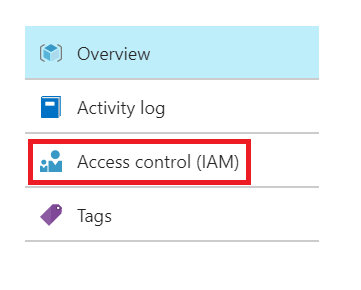
<http://azure.microsoft.com/en-us/documentation/articles/role-based-access-control-configure/>

1. Log in to the Azure portal at <https://portal.azure.com>.
2. When you first log in, find your name/initial on the upper right hand corner icon. When you select the icon, a dropdown appears. You need to confirm that the Directory that is chosen is the default directory. It should look something like <username><emaildomain>.onmicrosoft.com.

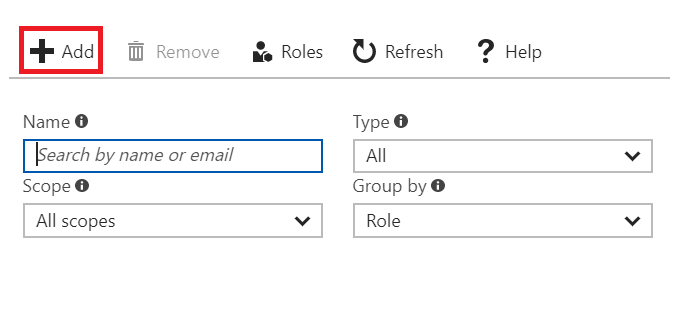


1. Select the **Browse** button on the left side of the portal window and browse for **Resource Groups**.
2. Find the resource group you created in Exercise 2 select it from the list. This should open up your resource group blade.
3. The first thing you need to do is set permissions at the **resource group** level. Recall from the table above that *kyle* is an administrator and should be set to Owner for all items in this resource group. For best practice though, it is best to add AAD groups, and not individuals, for specific permissions.

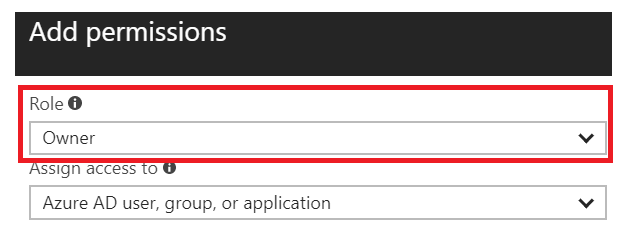
In the resource group blade, click on the Access control (IAM) tile to get to the RBAC settings.



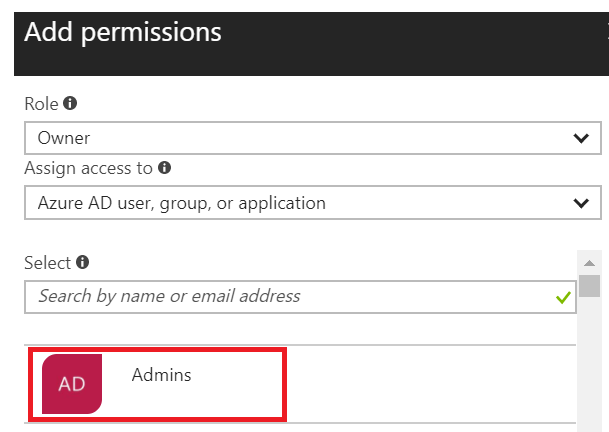
1. Click the **Add** button.



1. In the Add Permissions blade, choose **Role** and then choose **Owner**.

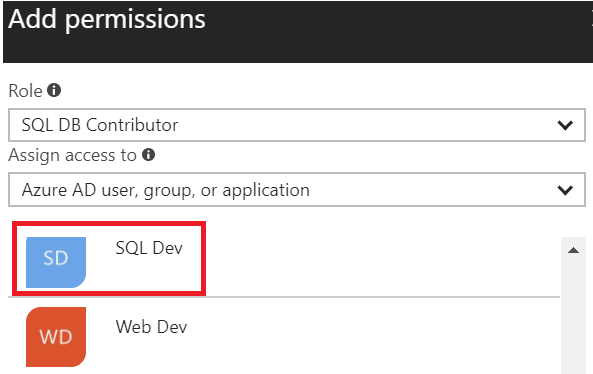


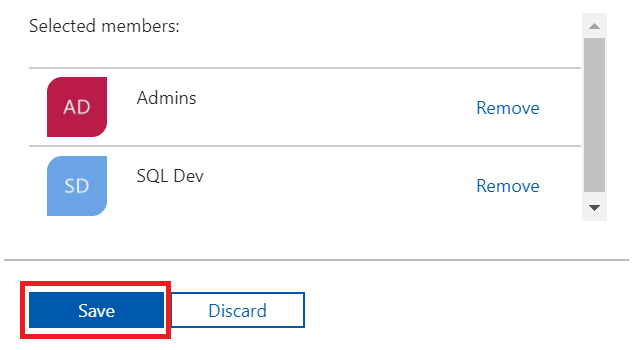
1. Select the **Admins** group.



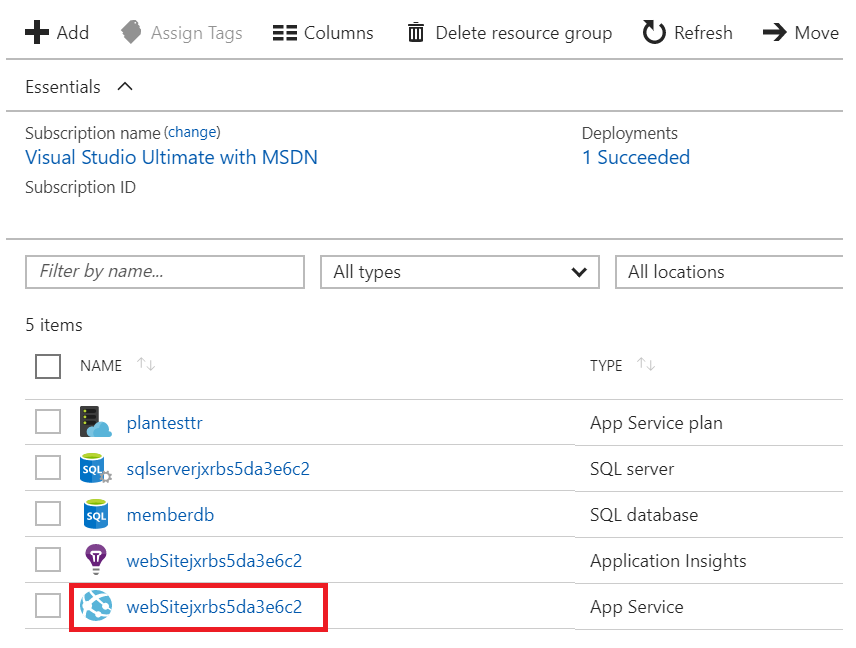
1. Recall that you want angie and alex to have database developer/modification permissions. In order to provide this, you **can’t** just click on the database name in the resource group, go to the database blade and then set users. Instead, you will need to set this permission at the resource group level.

With the Add Permissions blade still open, choose **Role** and then choose **SQL DB Contributor**, then select the **SQL Dev group** and click Save.

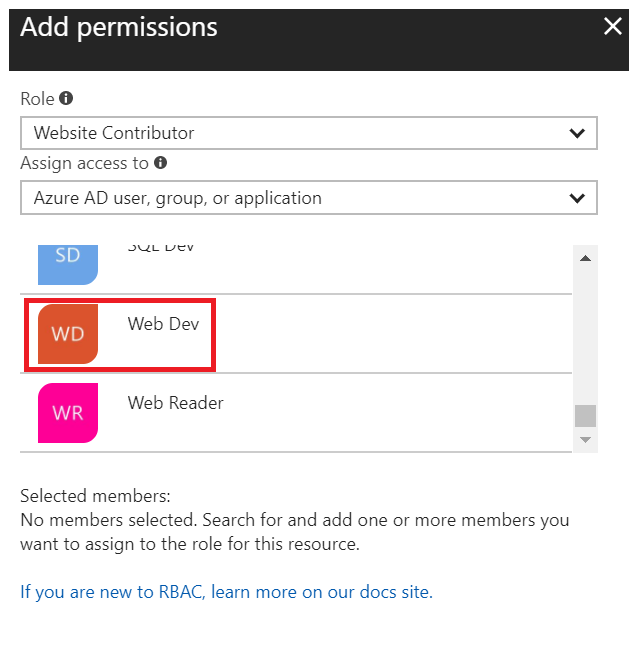




1. Close the Access Control (IAM) blade.
2. In order to add the permissions for our **Web Dev** and **Web Reader** AAD groups, you can click directly on the web app icon. Click the web app icon. The web app blade will open.



1. Click the Access Control (IAM) icon like you did earlier in order to get to the user permission settings.
2. On the Access Control (IAM) blade, click **Add**.
3. Click on Role and then choose **Website Contributor**, then choose the Web Dev group and click Save.



1. While still in the Add Permissions blade, select Role and choose the **Reader** role, then select the Web Reader group and click Save.

## Exercise 4: Test Users using RBAC permissions

Now that you have the role permissions set, you will test the capabilities of each of the users in your AAD groups.

### Task 1 – Confirm AAD Admin group permissions

1. Right click on your Internet Explorer icon and select **Start InPrivate Browsing**.
2. Log in to the Azure portal using the credentials [kyle@yourdomain.onmicrosoft.com](mailto:kyle@yourdomain.onmicrosoft.com). If you have not logged in using this credential before, you will be prompted to change the password.
3. Select the **Browse** menu button and then select **Resource Groups** and find the resource group you create in Exercise 2. Click on the resource group. You should notice that the only resource group(s) that will be shown in the list, are the ones that *kyle* has permission to access.
4. Within the resource group blade, you can perform tests like add other users permission, modify the web app etc and you will notice that *kyle* has full permissions.
5. As another test, click on the Home button on the left hand menu.
6. Select the **New** button in the lower left corner of the portal and try to create something new. You will be allowed to create a new resource, but you can only put the resource in the resource group *kyle* has permissions for.
7. Log out of the Azure portal by clicking on the Home button on the left side of the portal and then *kyles’s* icon on the upper right hand side of the portal window. Close the browser.

### Task 2 – Confirm AAD SQL Dev group permissions

1. Right click on your Internet Explorer icon and select Start InPrivate Browsing.
2. Log in to the Azure portal <https://portal.azure.com> using the credentials **angie@yourdomain.onmicrosoft.com**. If you have not logged in using this credential before, you will be prompted to change the password.
3. Select the **Browse** menu button and then select **Resource Groups** and find the resource group you create in Exercise 2. Click on the resource group. Notice that *angie* can only see the database server and memberdb database.
4. Click on the **memberdb** database, and then click on the **Settings toolbar icon** in the database blade. You can test here to see if *angie* has permissions to make database changes.

However, since *angie* is only a SQL DB Contributor, she cannot make changes such as turning on Auditing. Click the Auditing link to see that she has no access here.

1. Log out of the Azure portal by clicking on the Home button on the left side of the portal and then *angie’s* icon on the upper right hand side of the portal window. Close the browser.

### Task 3 – Confirm AAD Web Dev permissions

1. Right click on your Internet Explorer icon and select Start InPrivate Browsing.
2. Log in to the Azure portal <https://portal.azure.com> using the credentials **george@yourdomain.onmicrosoft.com**. If you have not logged in using this credential before, you will be prompted to change the password.
3. Recall that *george* is in the Web Dev AAD group, that has been given Website Contributor permissions, directly to the web app, meaning, *george* has no access to other resource group items.

Because the Web Dev group only has access to the web app, if *george* tries to browse for resource groups, he will not see any listed. Instead, select **Browse** and select **Web Apps**.

1. Select your web app from the list and this will open the web app blade.
2. Click on the **All settings** link.
3. Click down through the list of choices of settings. You will notice that *george* can change application settings, but cannot change things like Scale, Custom Domains, deployment slots etc. In order to change settings like these, *george* would need to be a Contributor.
4. Log out of the Azure portal by clicking on the Home button on the left side of the portal and then *george’s* icon on the upper right hand side of the portal window. Close the browser.

### Task 4 – Confirm AAD Web Reader permissions

1. Right click on your Internet Explorer icon and select Start InPrivate Browsing.
2. Log in to the Azure portal <https://portal.azure.com> using the credentials **fred@yourdomain.onmicrosoft.com**. If you have not logged in using this credential before, you will be prompted to change the password.
3. Recall that *fred* is in the Web Reader AAD group, that has been given Reader permissions, directly to the web app, meaning, *fred* has no access to other resource group items and he can only read settings for the web app.

Because the Web Reader group only has access to the web app, if you try to browse for resource groups, you will not see any listed. Instead, select **Browse** and select **Web Apps**.

1. Select your web app and then select the All settings like.
2. Notice that if *fred* goes to Application Settings, most settings he can read, some settings, such as connection strings, *fred* would have no access to.
3. Log out of the Azure portal by clicking on the Home button on the left side of the portal and then *fred’s* icon on the upper right hand side of the portal window. Close the browser.