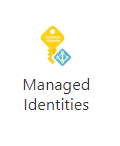
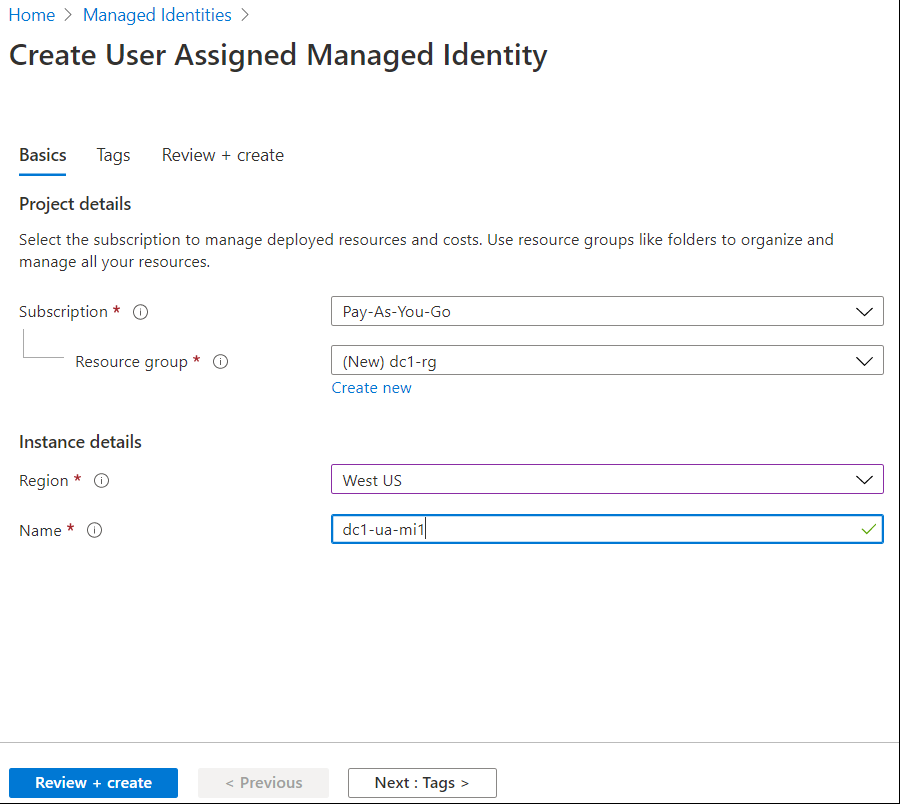
Access Azure Blob Storage from Azure App Service using User Assigned Managed Identity

Reference sample code: <https://github.com/milindvb/mbJavaSDKBlobStorageUserManagedIdentity>

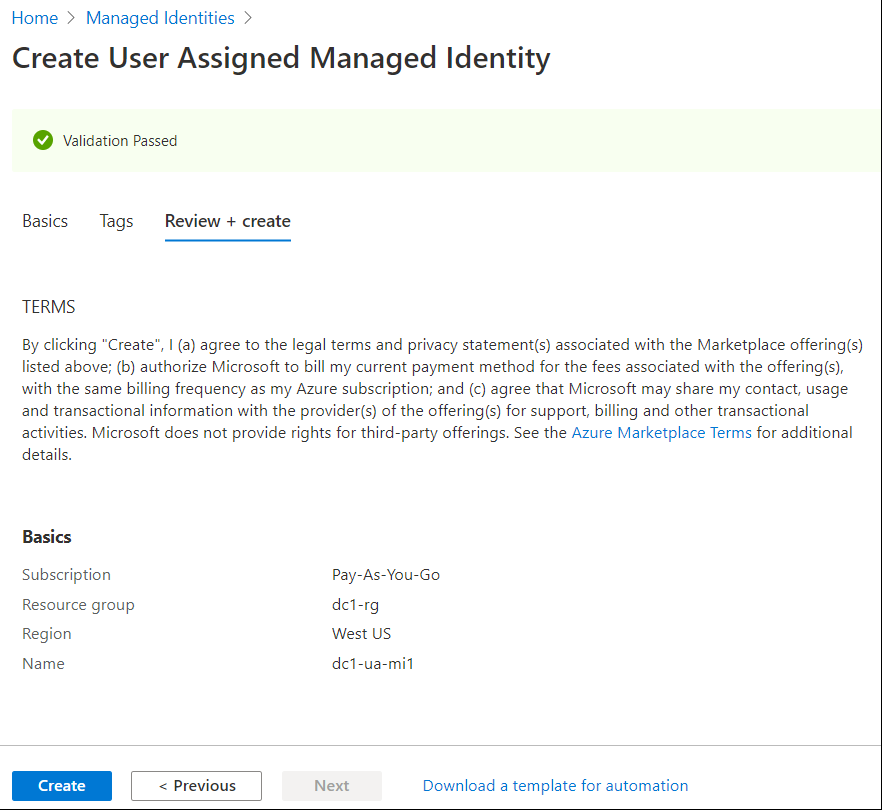
* Create a User Assigned Managed Identity





* Click on **Review + Create**

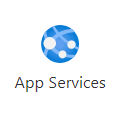
On the next screen

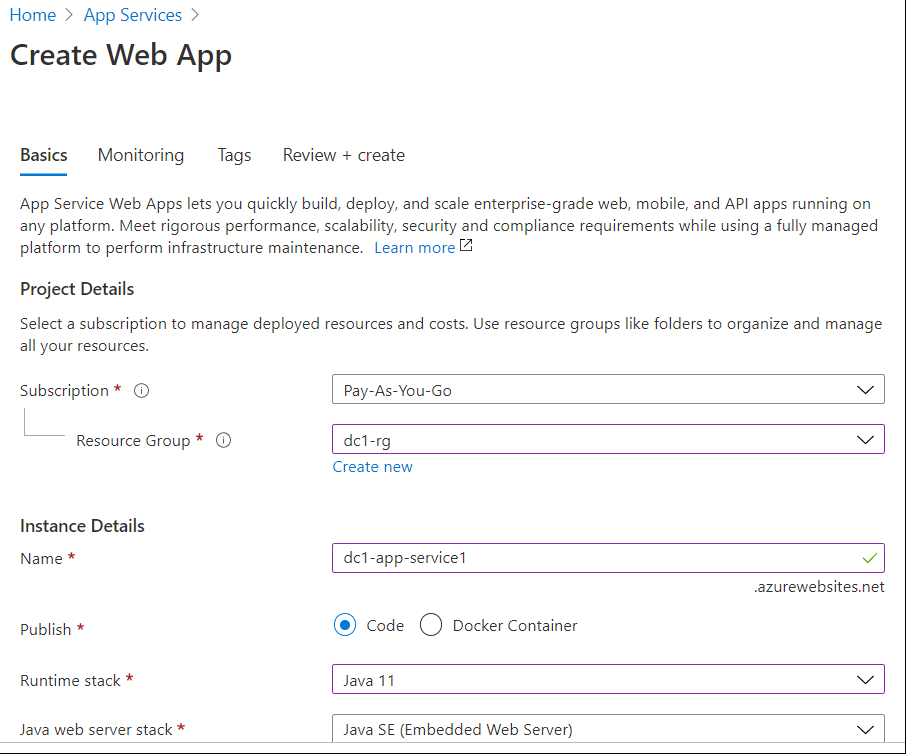
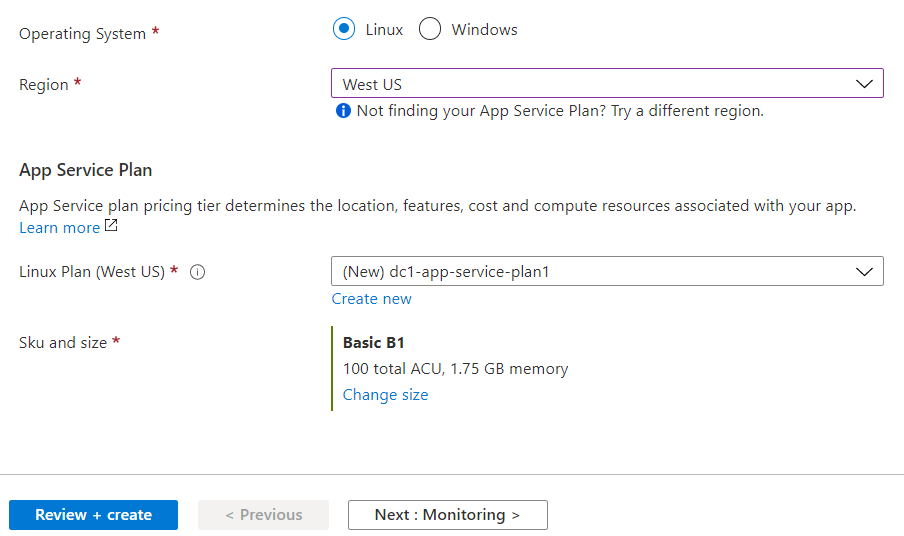


* Click on **Create**
* Go to Managed Identity detail screen, click on **Overview**
* Copy **Client Id** value and save it in an easily available location

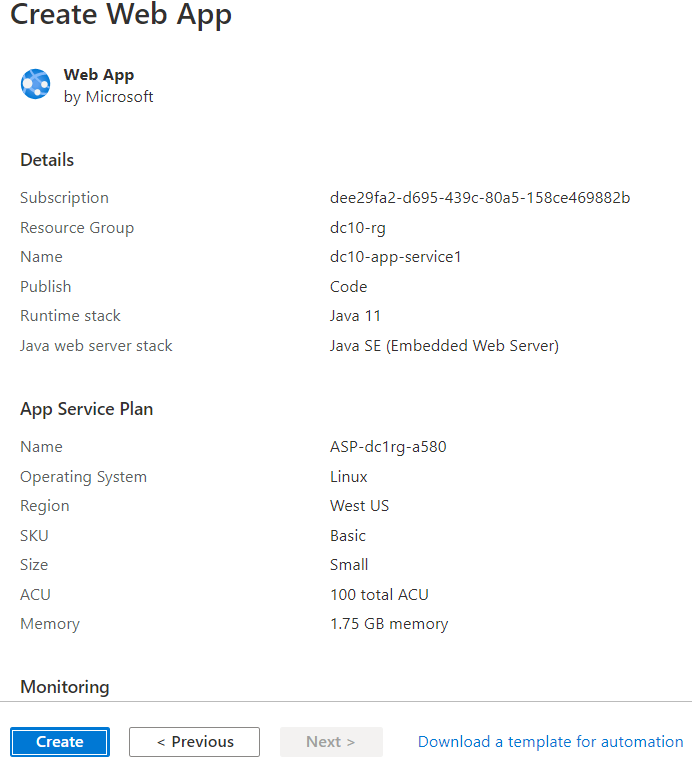


* Create an App Service

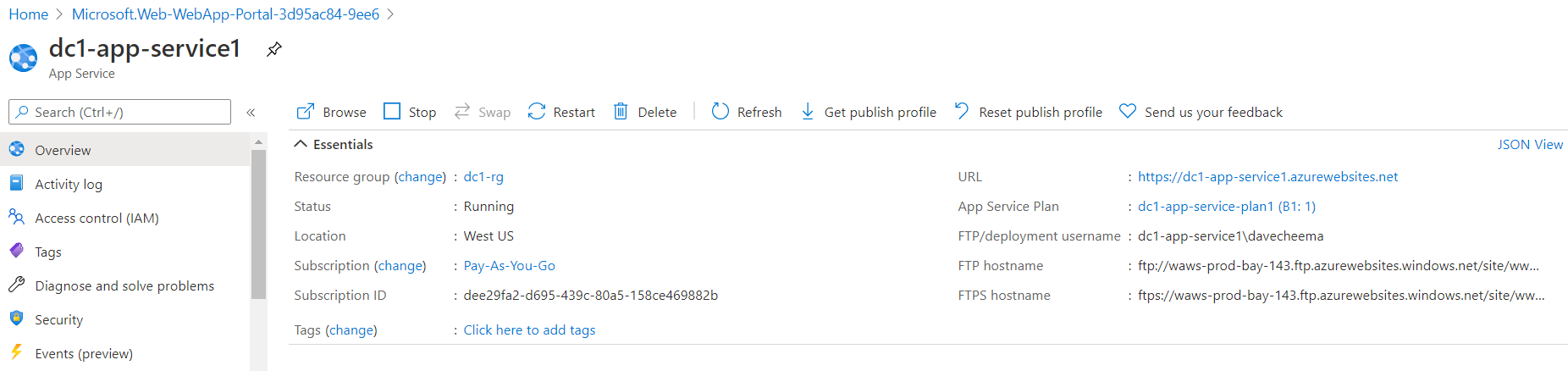


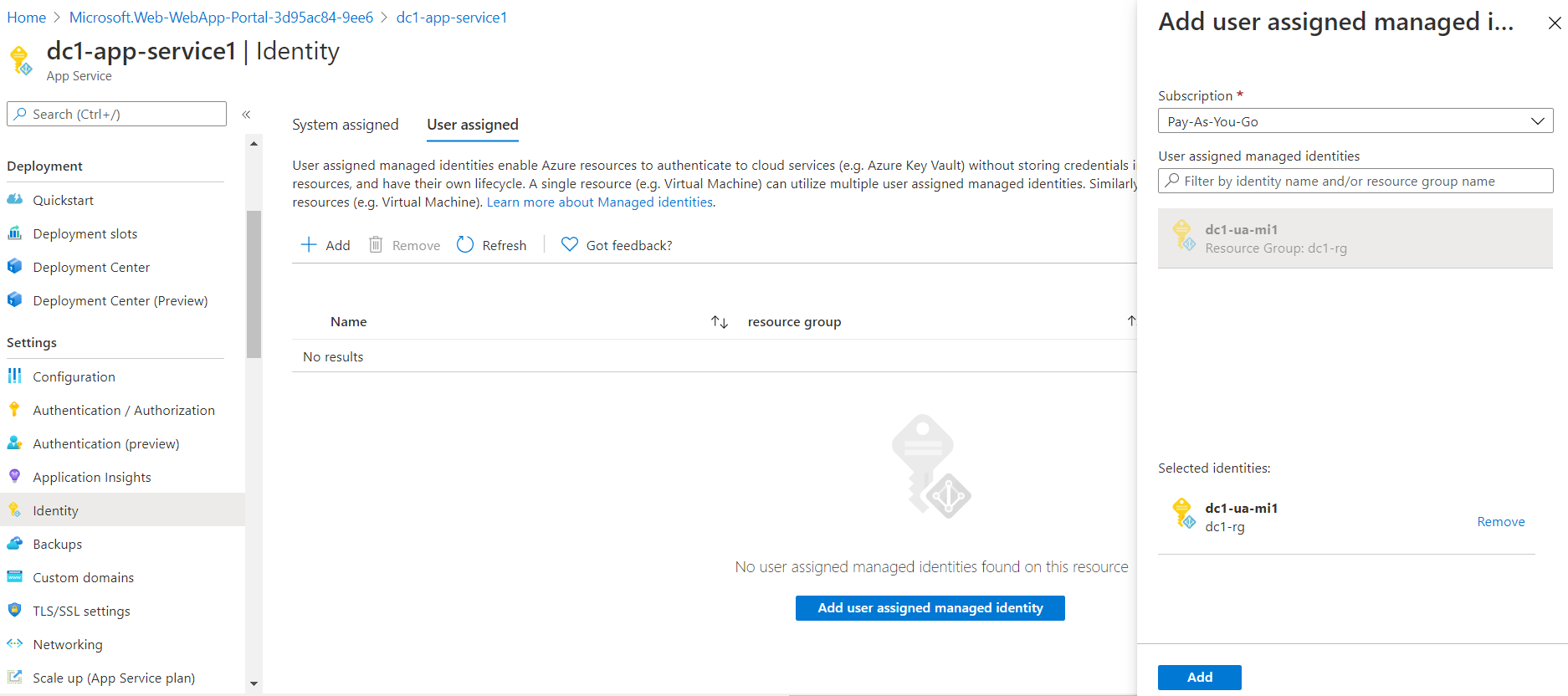
* Fill out all fields, click on **Review + create**



* Click on **Create**
* Note down app service details on the top right side of the page

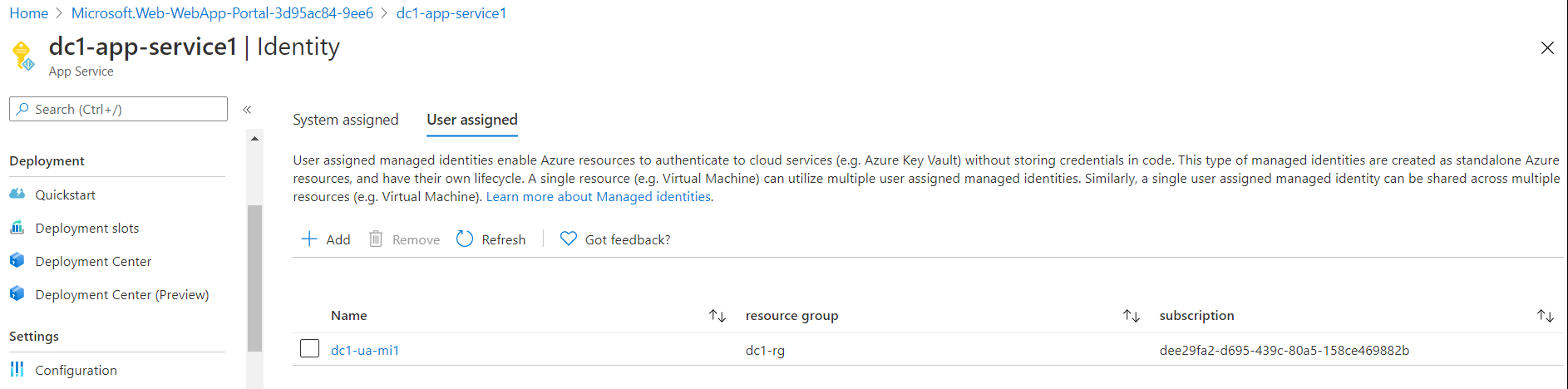


* Go to App Service details

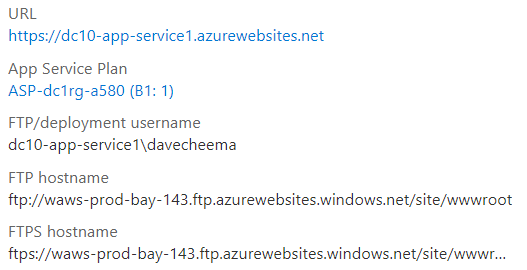


* Click on Identity on the left side
* Click on User assigned on the top right side
* Search for User Managed identity, e.g., dc1-ua-mi1
* When found, click on it and click on Add at the bottom left of the screen

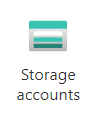
You will see:

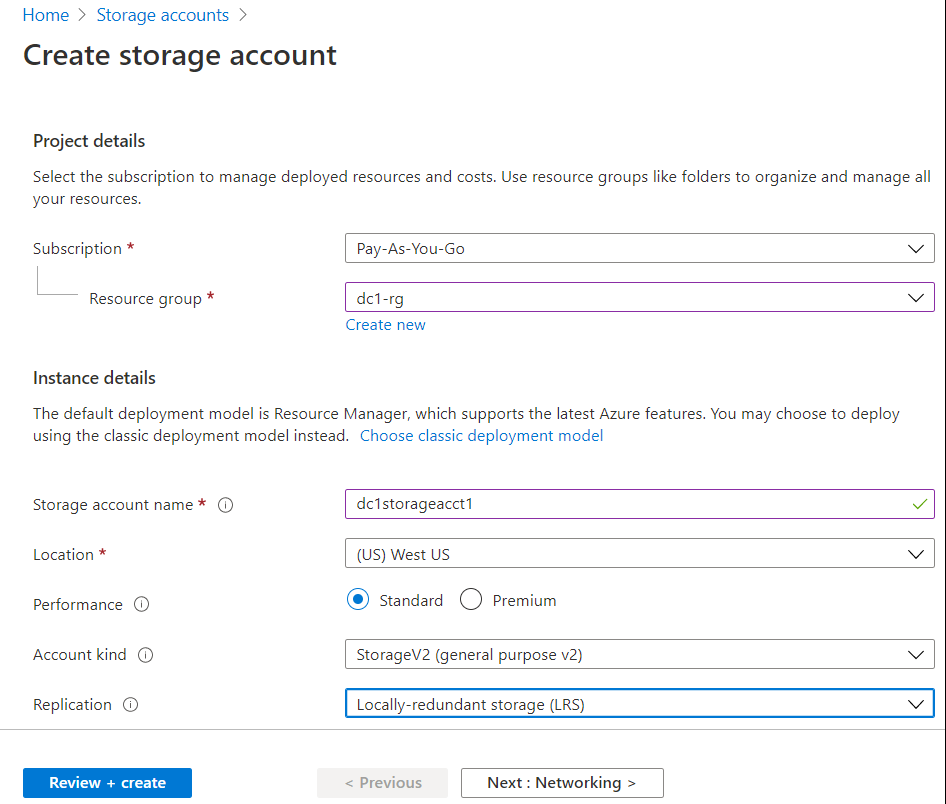


* Go to App Service detail screen and copy the following items and save them to a known location:

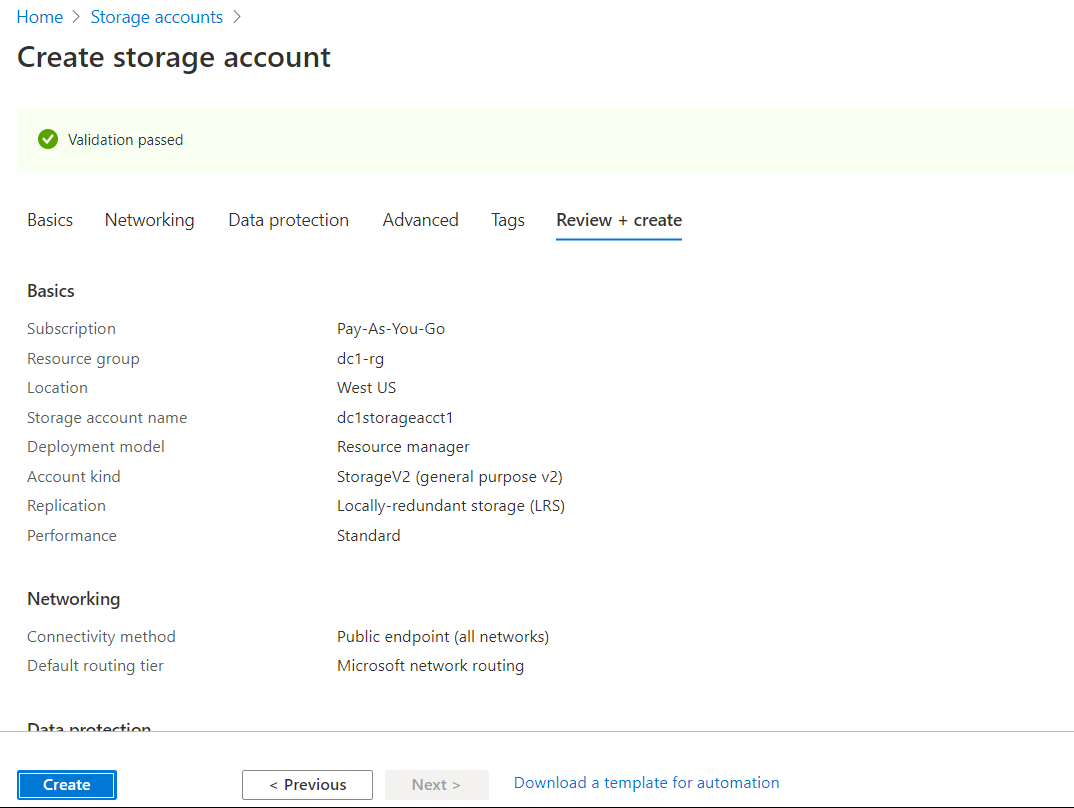


* Create a storage account

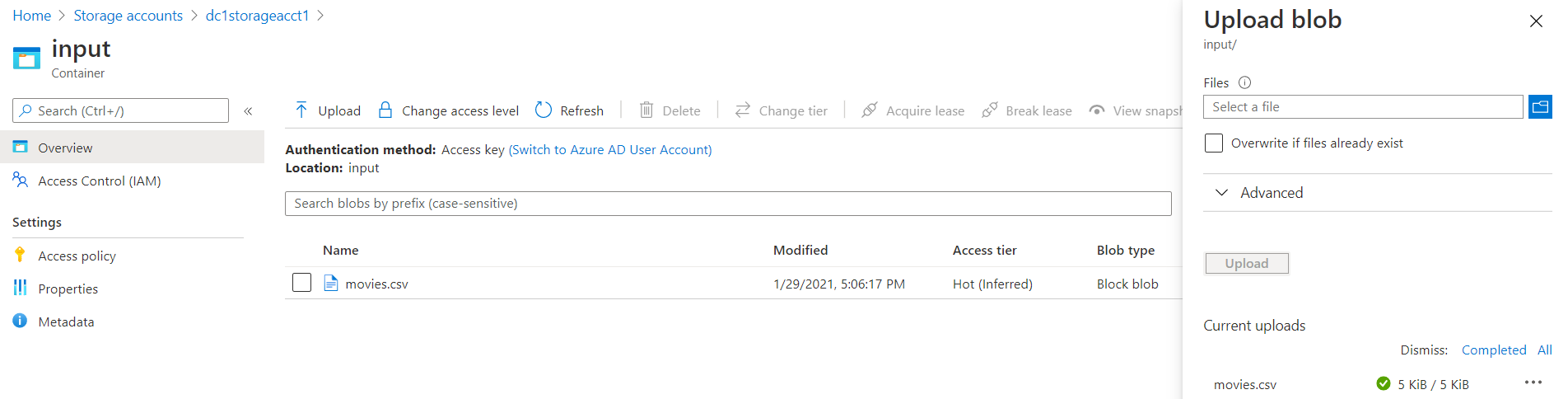




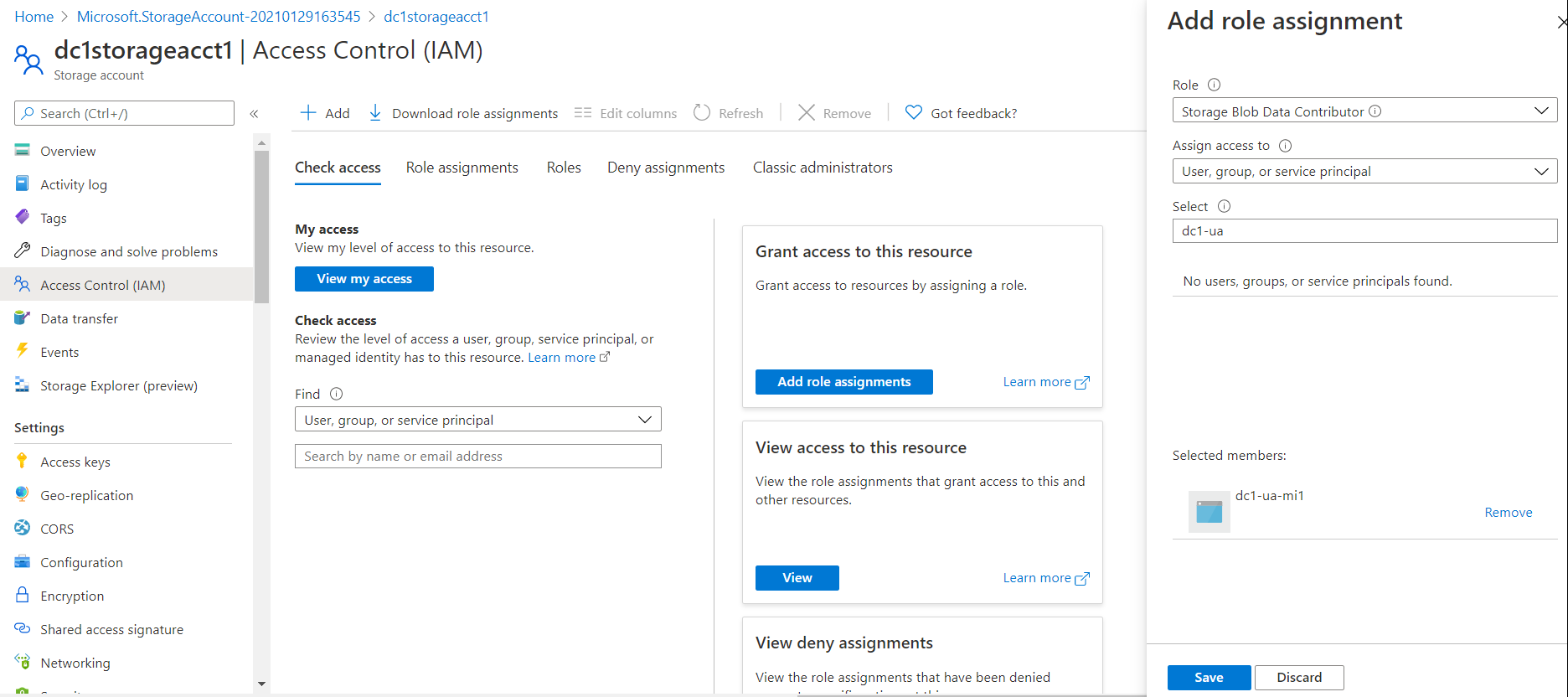
* Enter all required information, click on **Review + create**



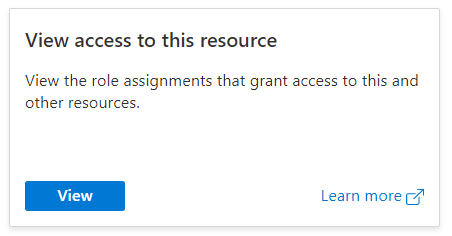
* Create on Create
* Create a container and upload a test file as shown below:



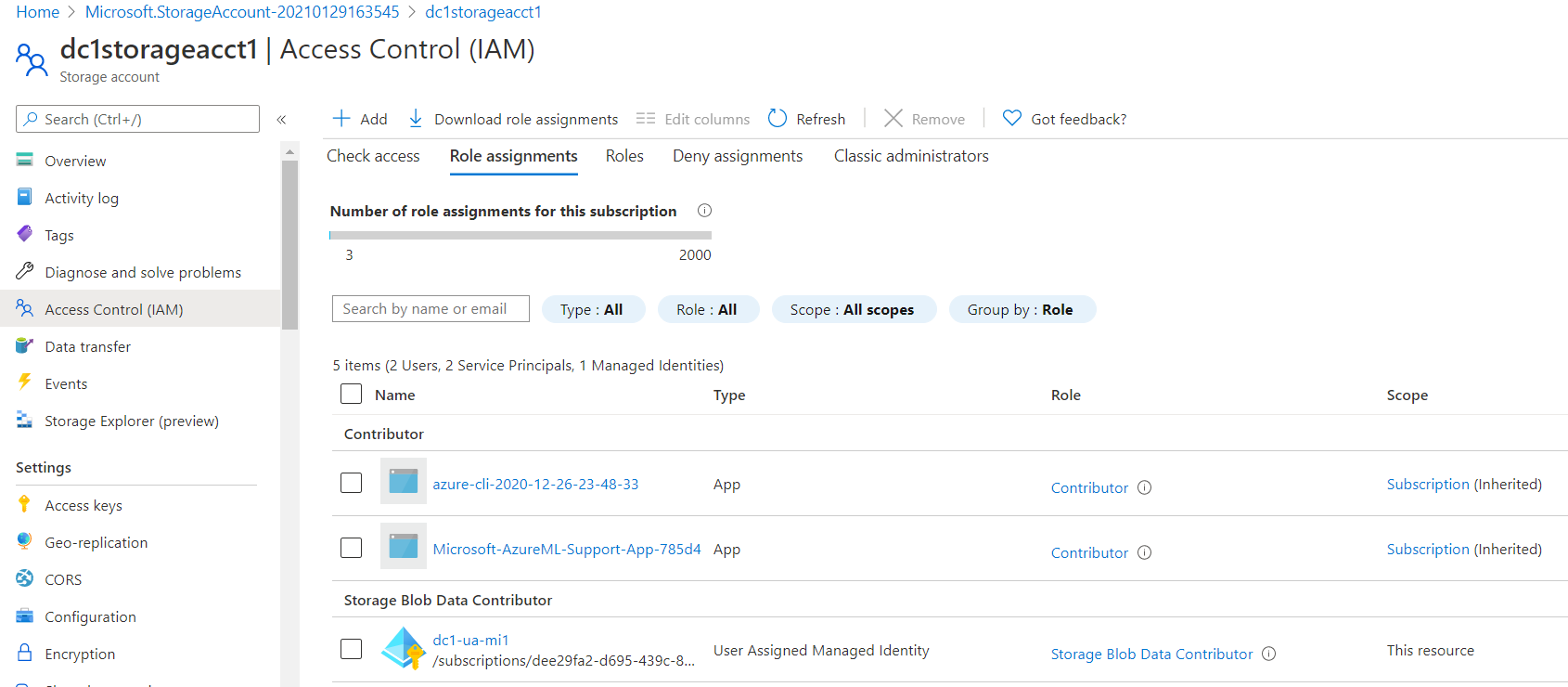
* Go to Storage Account detail



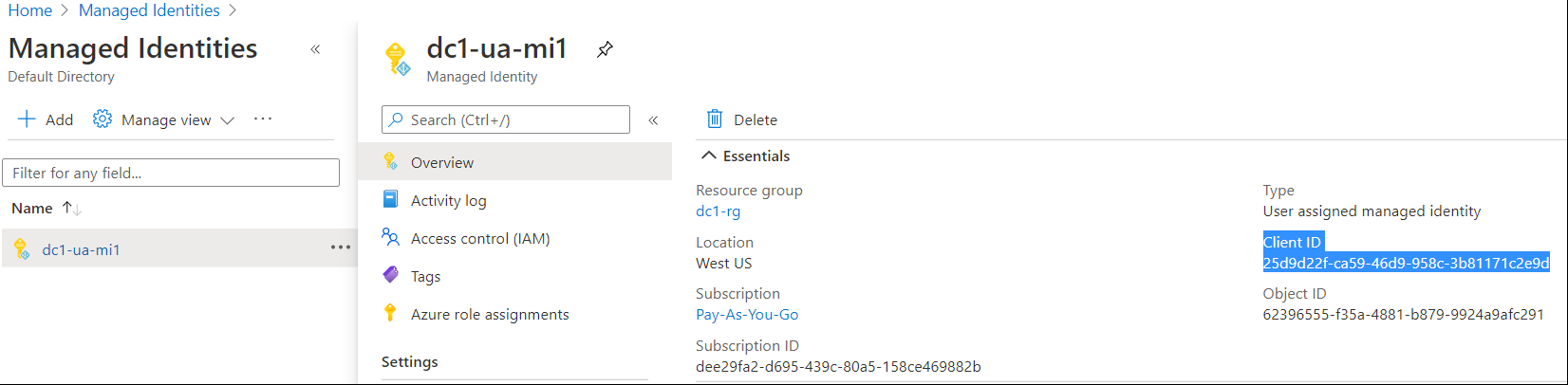
* Click on Access Control (IAM), Click on Add
* Select **Storage Blob Data Contributor**, select **dc1-ua-mi1** and click on **Save**
* Optionally, you can verify the permissions have added are saved



You will see something similar to:



* You should be able to see Storage Blob Datta Contributor, dc1-ua-mi1
* Go the User Managed Identity detail screen:

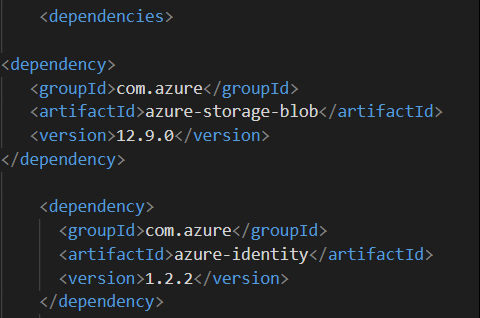


* Note down the **Client ID**
* Go to your **development environment**, e.g., Visual Studio Code, NetBeans, etc., create a web app.

**Note:** the code for this example used here is downloaded from <https://github.com/milindvb/mbJavaSDKBlobStorageUserManagedIdentity> and Visual Studio Code was used to develop and compile code

* Open the project in Visual Studio Code
* Open **pom.xml** file

1. Go to the **<dependencies>** section and add the following elements:



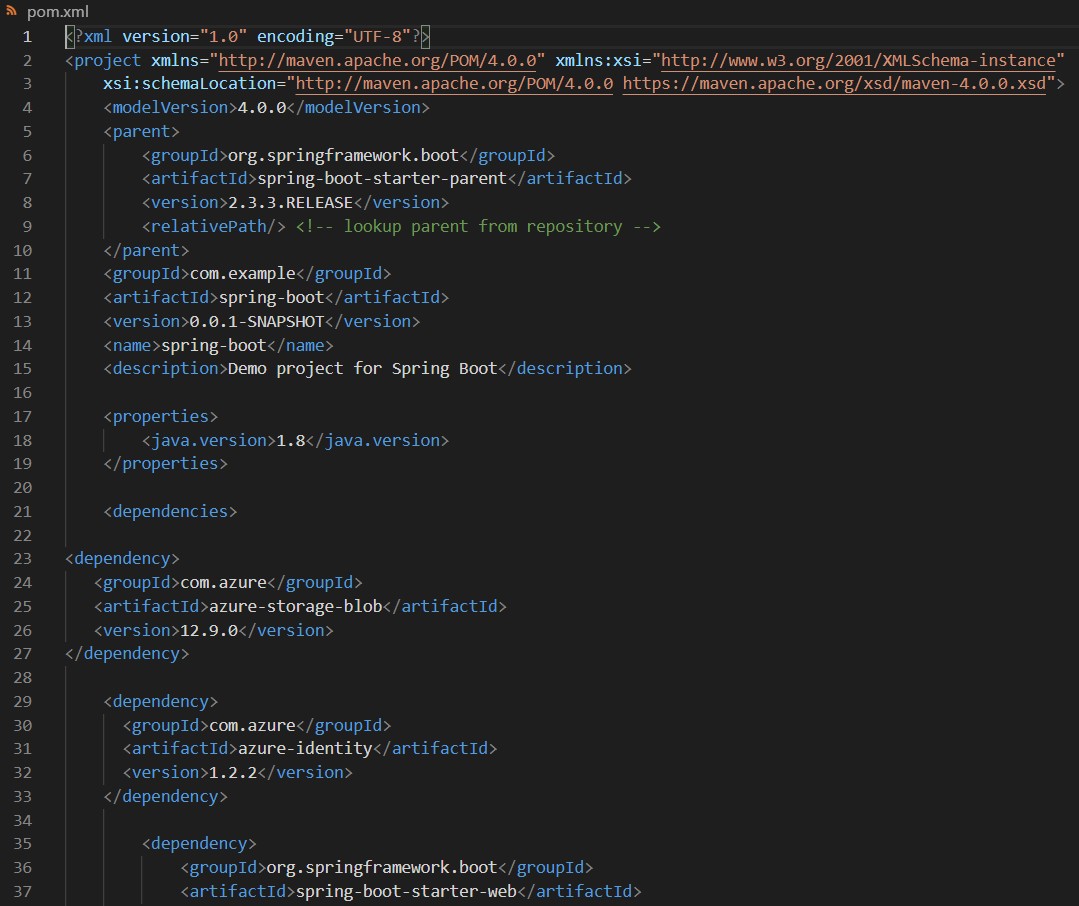
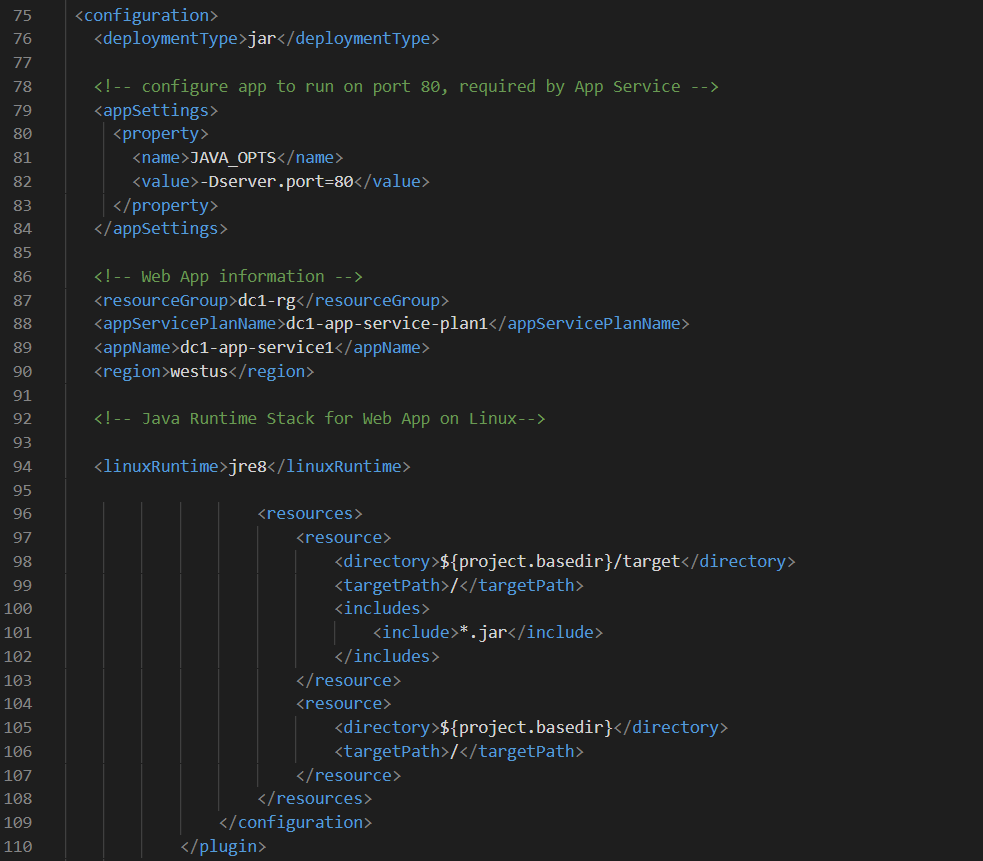
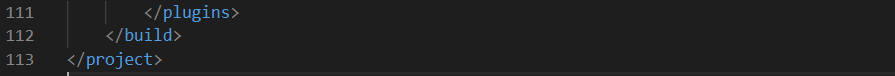
1. Add/update the following elements:

|  |  |  |
| --- | --- | --- |
| Machine generated alternative text: Web App information  <resourceGroup>changeThis-ResourceGrp</resourceGroup>  <appName>changeThis-Appname</appName>  centralus</region>  <reglon>  Java Runtime Stack for Web App on Linux-->  <1inuxRuntime>jre8</1inuxRuntime> | =====> | Machine generated alternative text: Web App information  <resourceGroup>dc1-  < appServicep1anName>dc1 -app- service-planl</ appServicep1anName>  <appName>dc1-app-service1</appName>  <region / region >  Java Runtime Stack for Web App on Linux-->  <1inuxRuntime>jre8</1inuxRuntime> |

Note: It must be just above the <linuxRuntime> element

* **Save** pom.xml

The completed pom.xml file should look similar the sample pom.xml shown below:

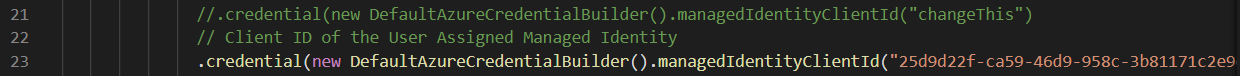
   

* Open src --> main\java\com\example\springboot --> HelloController.java file
* Make the following changes:

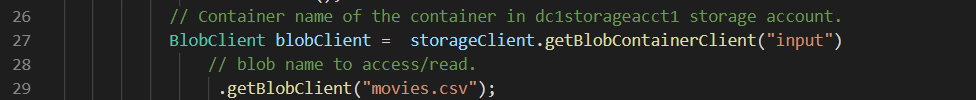
Line# 18: replace the storage account name with your own storage account name



Line# 23: replace the managedIdentityClientId with the Client ID of your user-assigned managed identity

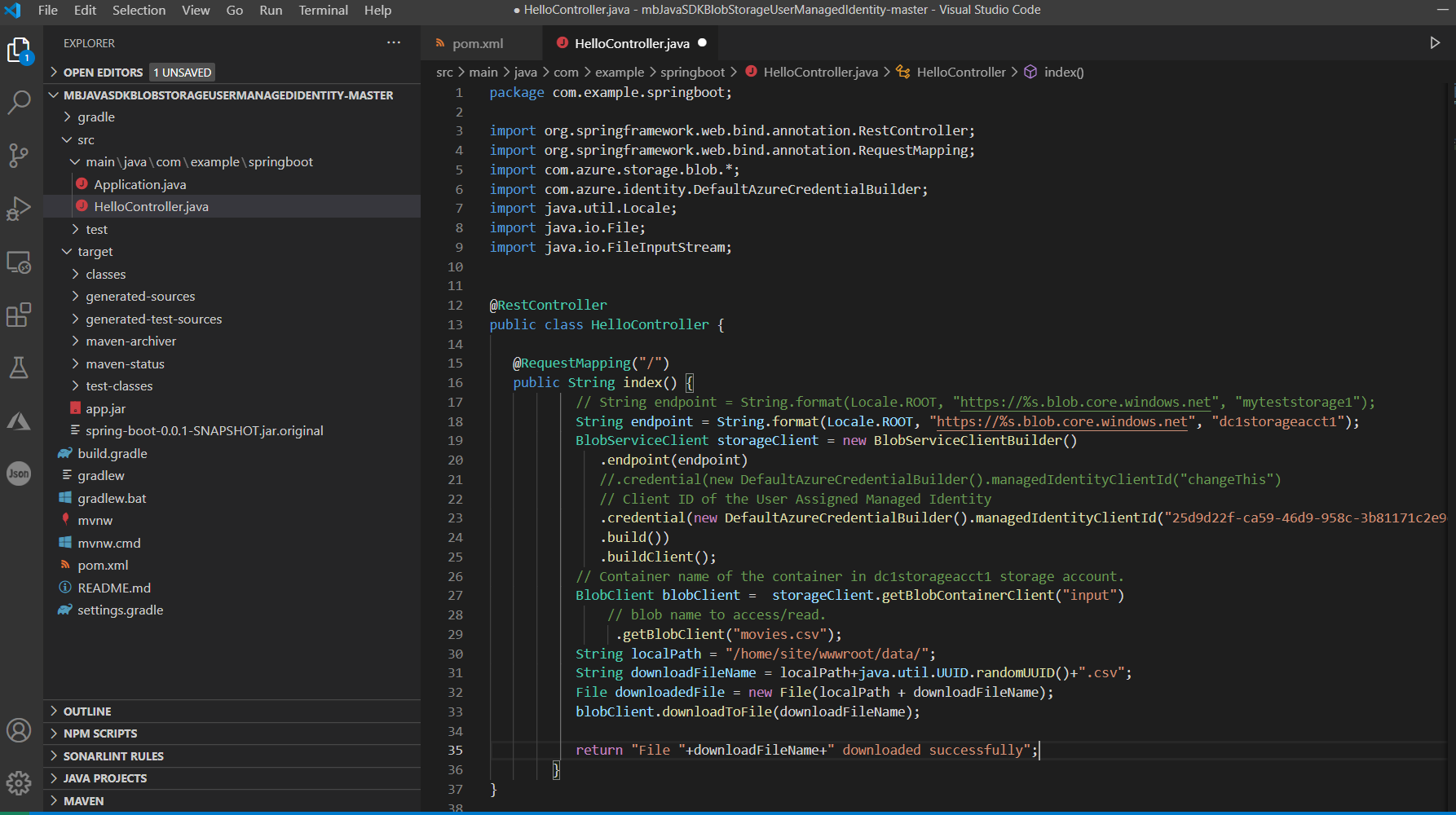


Lines 27 and 29: replace “input” with the name of your container. Replace “movies.csv” with the name of your blob



Note: your line numbers may vary.

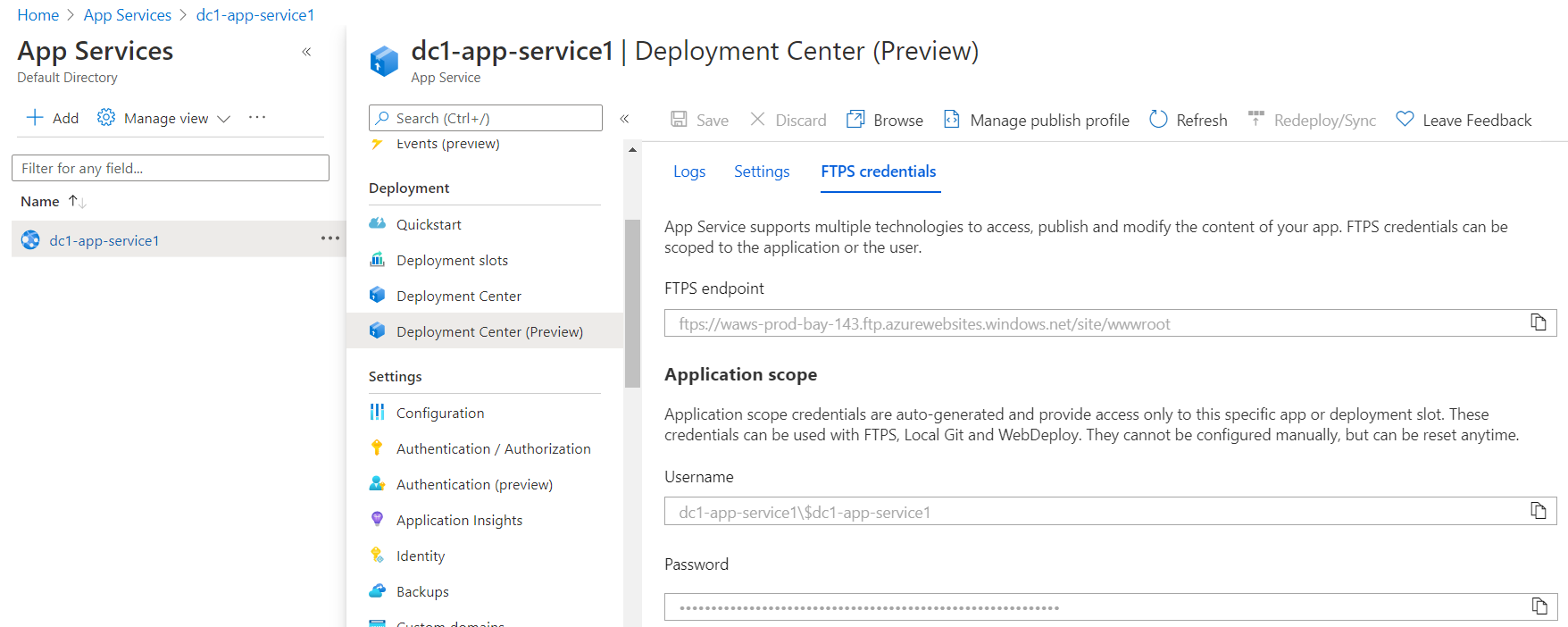
The completed file should look like the image of the source file shown below:



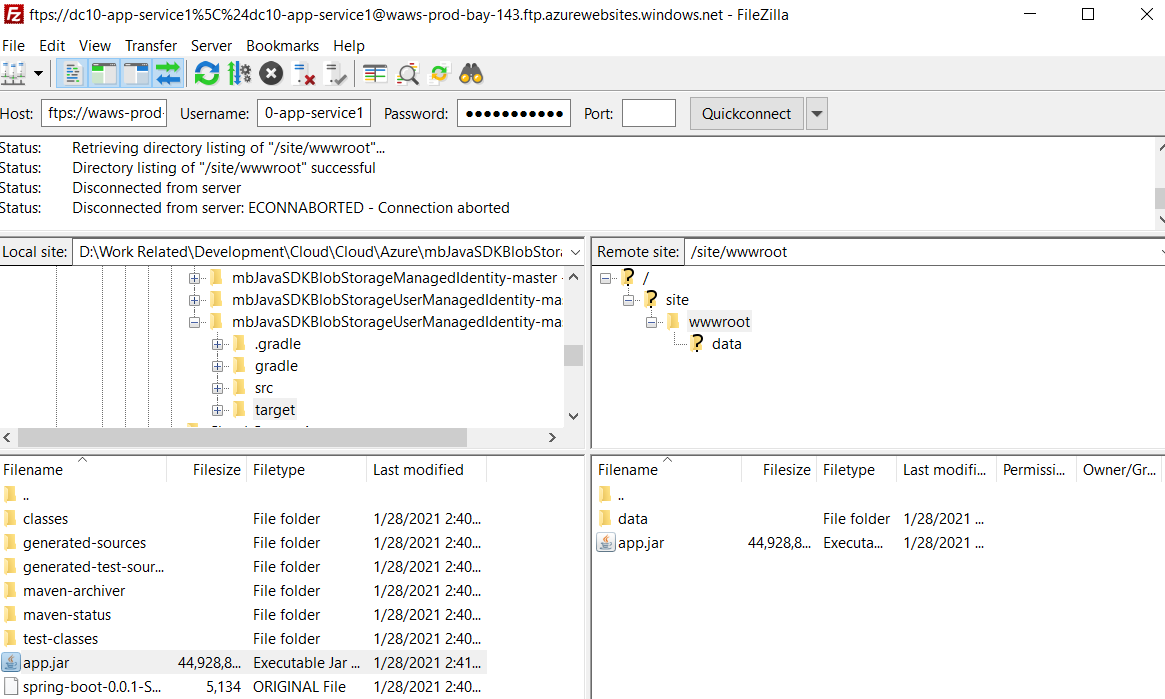
* You can clean the workspace, compile code, turn off the test project and create the JAR file using the following command in a Terminal window: **mvn clean package -DskipTests**
* Rename the .jar file

|  |  |  |
| --- | --- | --- |
| Machine generated alternative text: v main \ java \ com\example\springboot  O Applicationjava  O HelloController.java  > test  v target  >  >  >  >  >  >  classes  generated-sources  generated-test-sources  maven-archiver  maven-status  test-classes  spring-boot-O.O. 1 -SNAPSHOT.jar | Rename  .jar file to  App.jar | Machine generated alternative text: v main \ java \ com \example\springboot  O Applicationjava  O HelloController.java  > test  v target  >  >  >  >  >  >  classes  generated-sources  generated-test-sources  maven-archiver  maven-status  test-classes  app.Jar |

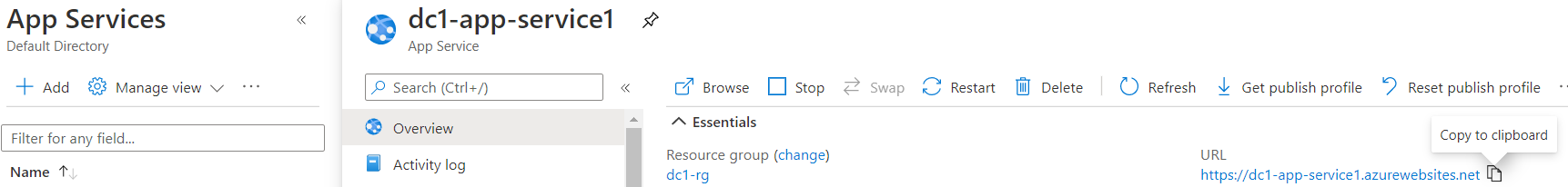
* Go to the target section on the left side and rename **spring-boot-0.0.1-SNAPSHOT.jar** 🡪 **app.jar**
* Go to your App Service instance in the Azure Portal, e.g., dc1-app-service1
* Click on Deployment Center (Preview) 🡪 click on **FTP credentials**



* Open up an SFTP (e.g., Filezilla) session
* Copy **FTPS endpoint**, **Username**, **Password** settings of the App Service and enter them into the FileZilla credentials. Hit Enter
* Upload the app.jar file into the Azure App Service workspace



* Go back to the Overview section of your App Service instance in the Azure portal, **Stop and Start** the service.
* Copy the URL of the App Service, e.g., https://dc1-app-service1.azurewebsites.net



* Wait a couple of minutes
* Open a new tab and paste the app service address in the Browser’s Address Bar

If all goes well, you should a message similar to what is shown below:

