**Connect Azure Service Management API’s**

To connect Azure service management API’s programmatically from our app then there will be two options available.

1. Azure Management Certificate
2. Azure Active Directory

**Azure Management Certificate**

To connect the API’s using management certificate we have to generate and download new certificate from azure then we have to convert that into X509Certificate2 and attach this certificate under WebRequestHandler. ClientCertificates in your web request call then you can get the results. Kindly follow the below steps to achieve this.

1. Download the Azure Setting file by clicking the below link.

<https://manage.windowsazure.com/publishsettings/index?client=xplat>

once you click on the link you have to login with your Azure Live credential then you got your setting file downloaded.

1. If you open that file, it just an XML file you will have your subscription with ID and manage certificate Just parse that certificate using C# XML class and convert that into an X509Certificate2



1. Once certificate is ready you have to insert into HTTP Handler then you can call an azure API’s using that handler.



1. That’s All, and you can find complete project at

<https://github.com/M-Sivanraj/AzureAPIAccess_MC>

**Azure Active Directory**

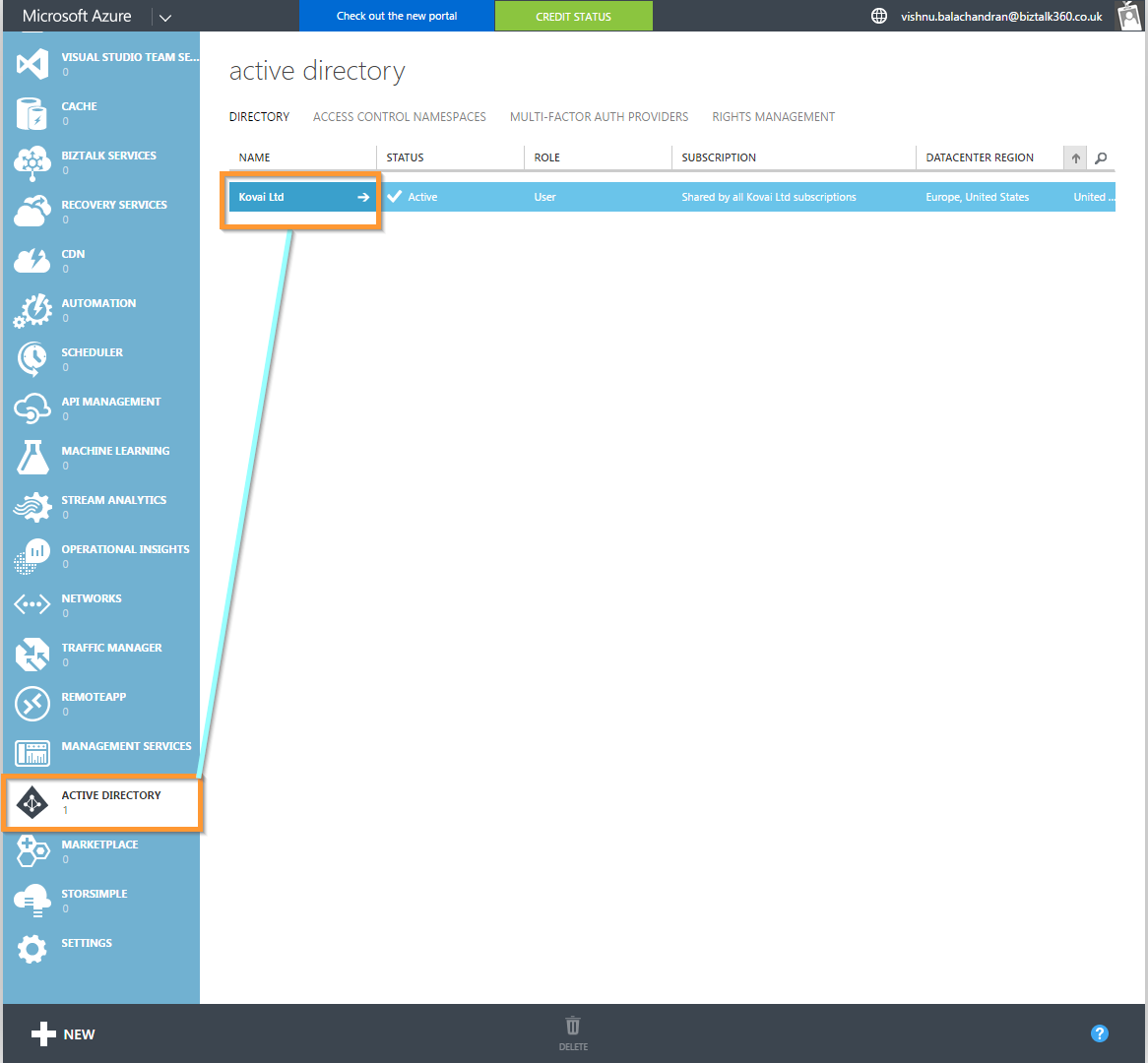
To connect using azure active directory then you have to create a new active directory in Azure and get an ClientID, Appkey and TenantID to call the Azure API’s.



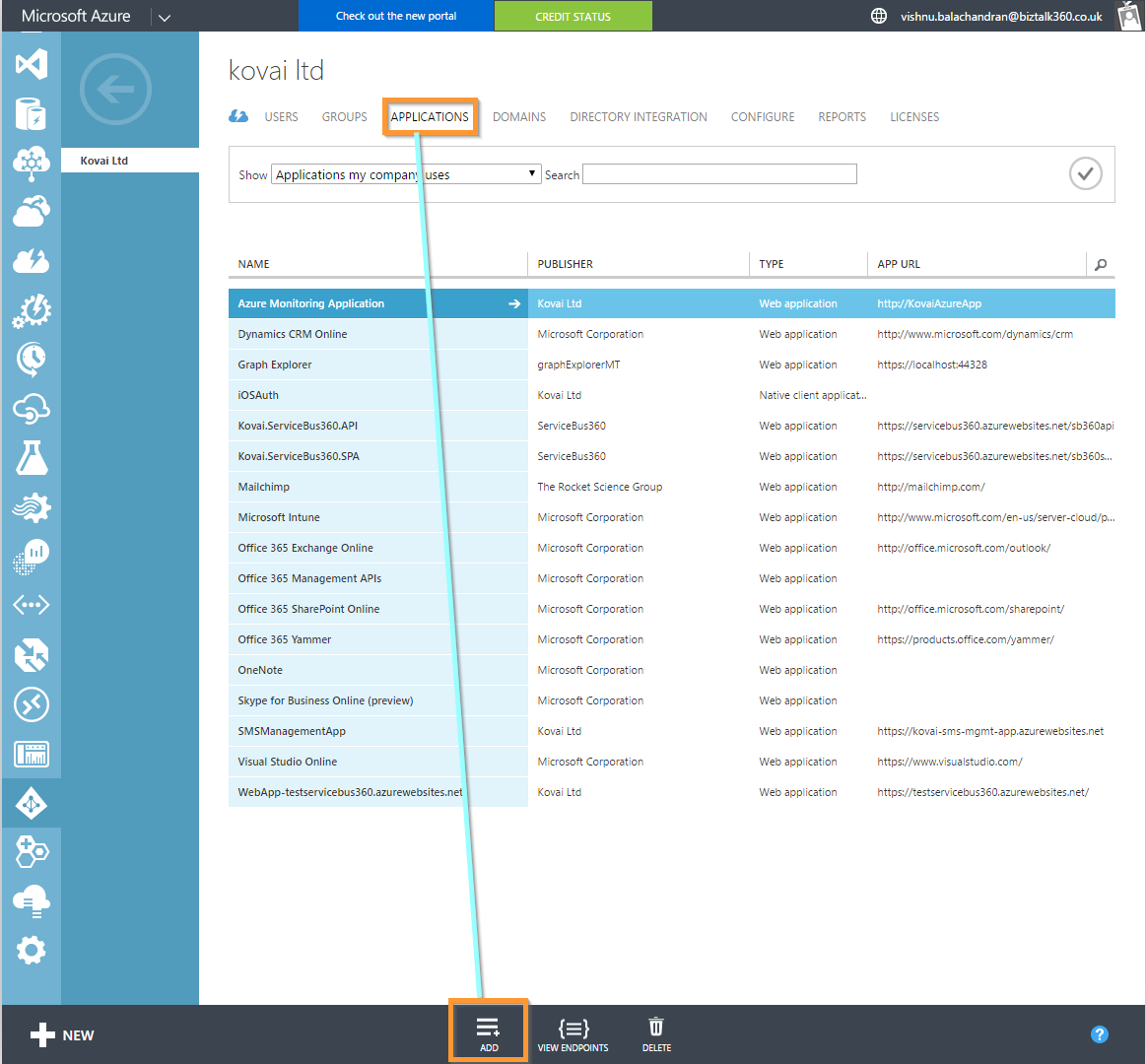
So the above values needed to connect Azure service management API’s using Azure AD, now we will see how and where we will get the above all values.

Login to you Azure ( [manage.windowsazure.com](http://manage.windowsazure.com) ) account.

Click on the **Active Directory** and then click on the **name of your default active directory.**

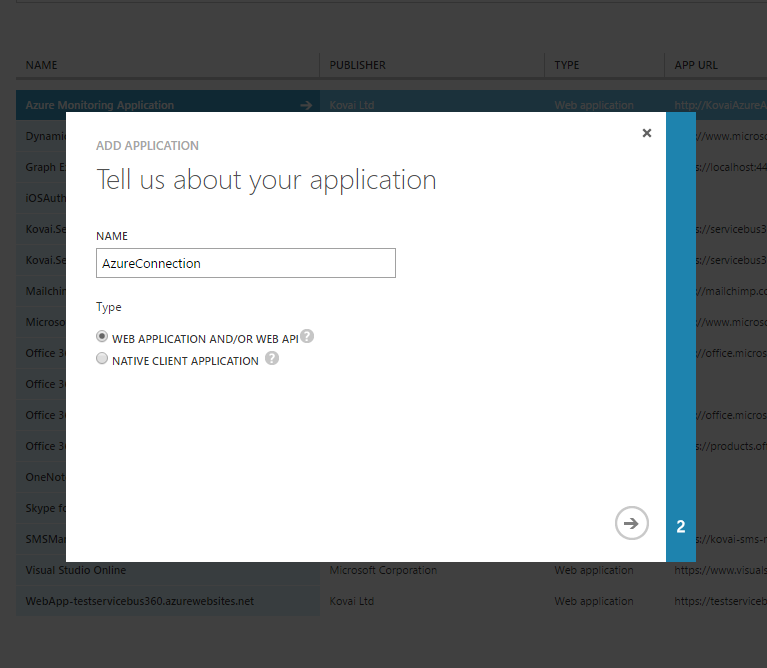


Click on **applications** and click on **Add** at bottom of you Portal.

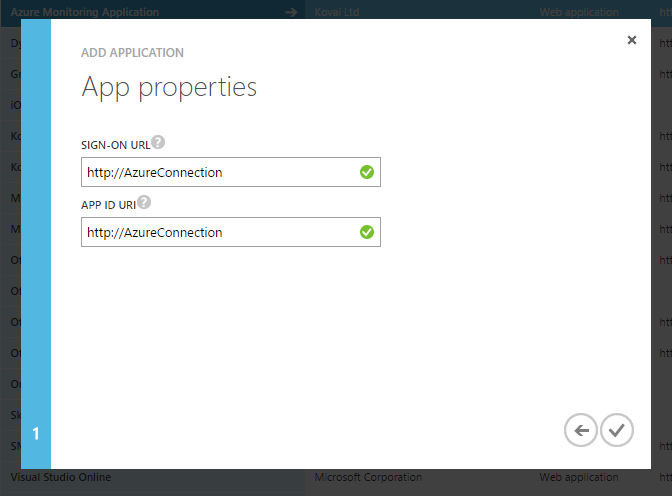


If you click on the **Add** you will find the popup with some input field.

Give your **Application Name** and click **Next**.

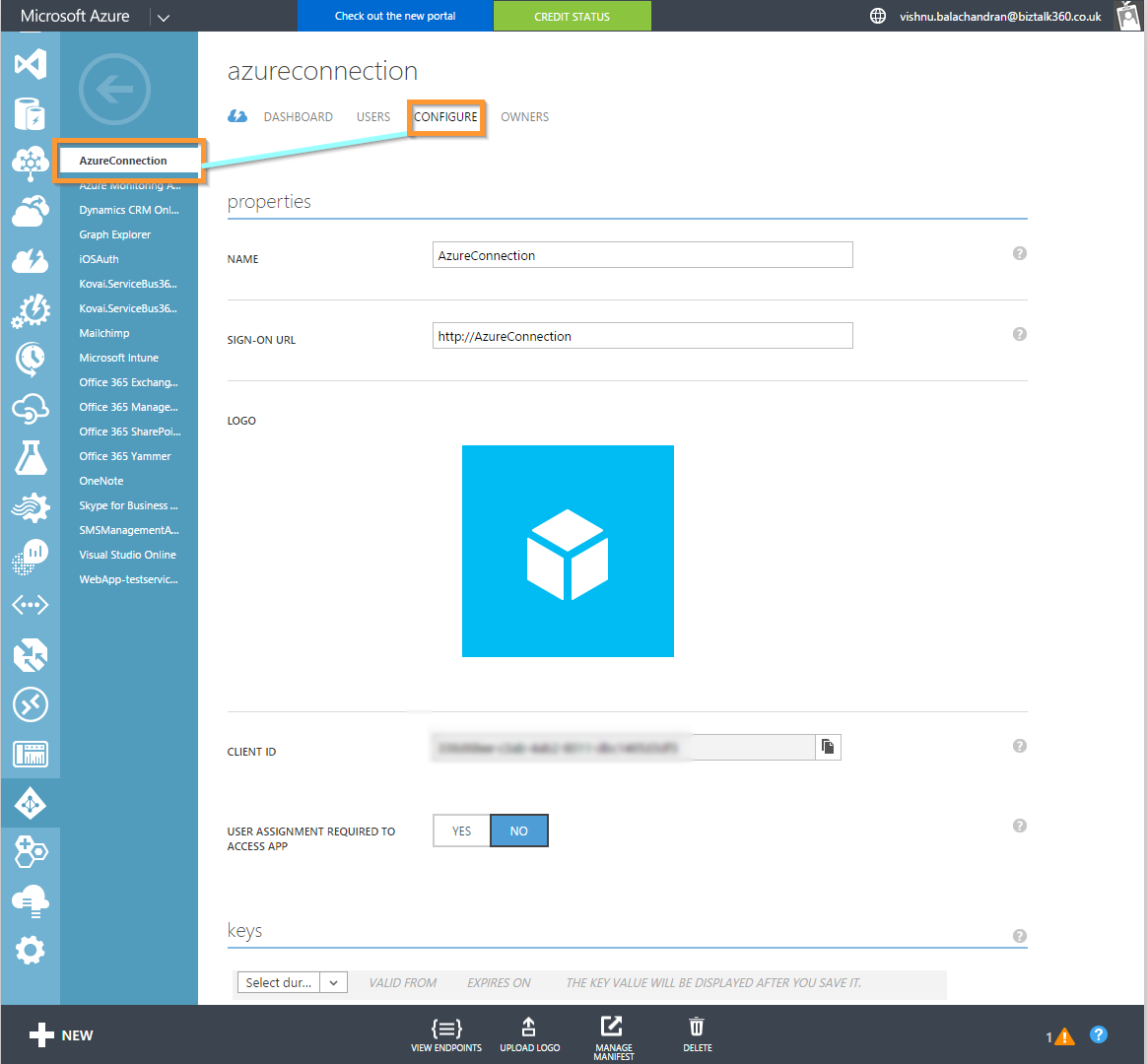


And give the **application URI** with right format like below. And click tick Icon.

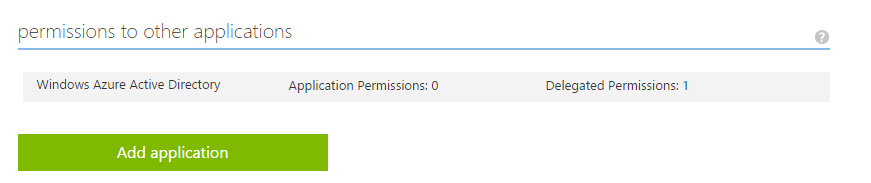


Once your app is added you will see the application dashboard page and you will find various links at top of the page.

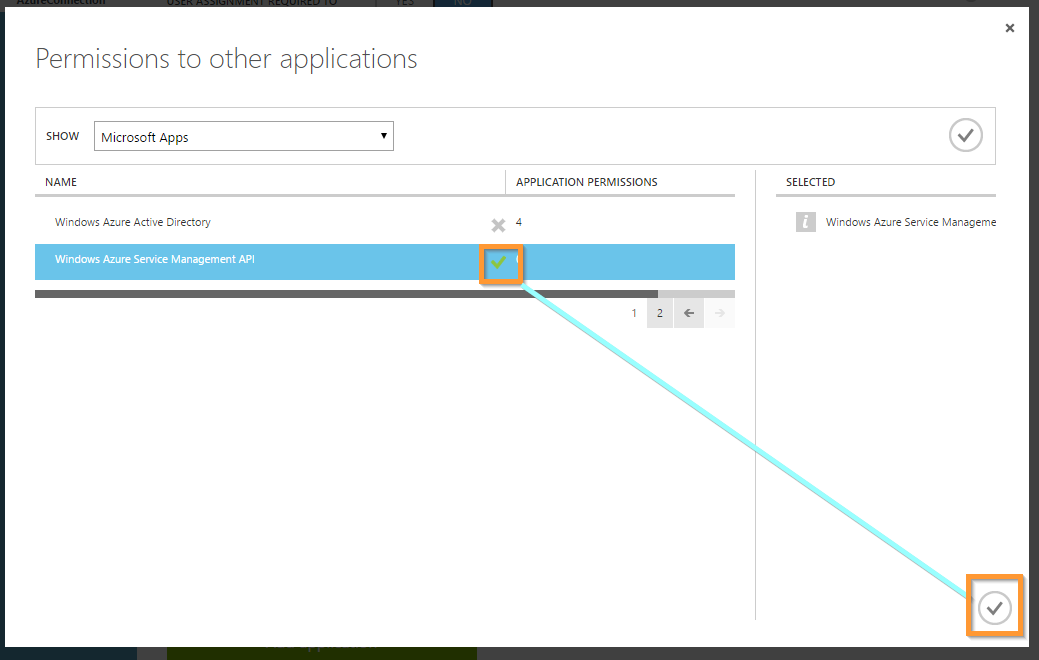
Click on the **CONFIGURE** Button which will appear on top of the page.



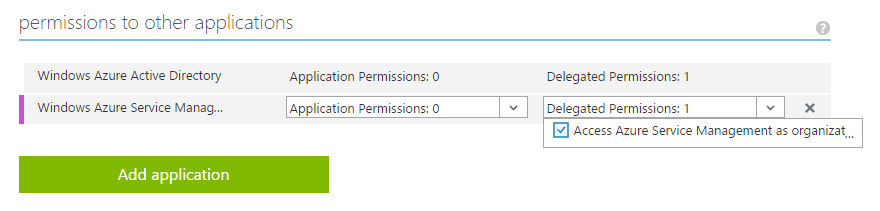
At the bottom of the "Configure" page you can add "permission to other applications". Go ahead and add another application.



Choose the "Windows Azure Service Management" application.



Make sure to select the permission you want to grant as well.



And that's about it! Save the changes and we are ready to start making our code example run!

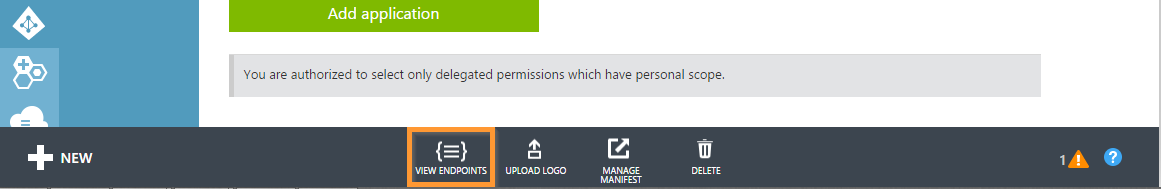
Everything is set up and ready for us now. If you see my app setting screen above, we have to fill following properties there.

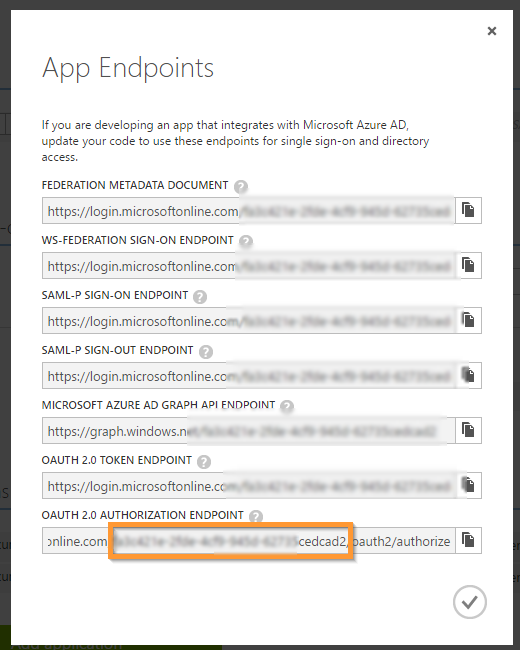
* TenantID
* ClientID
* AppKey
* SubscriptionID

**TenantID**

Tenant is just the default domain for your Active Directory. You can find your tenantID in your endpoint popup.

So click on the **VIEW ENDPOINTS** at the bottom of the page.

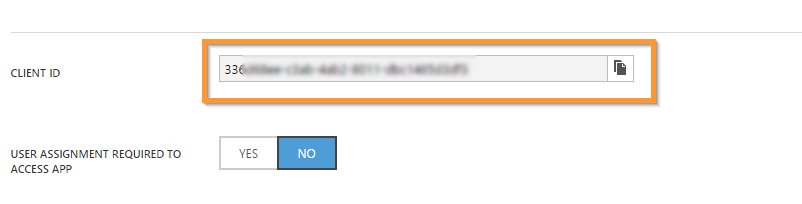




You can find the tenant id there copy that and replace that with our app setting KEY.

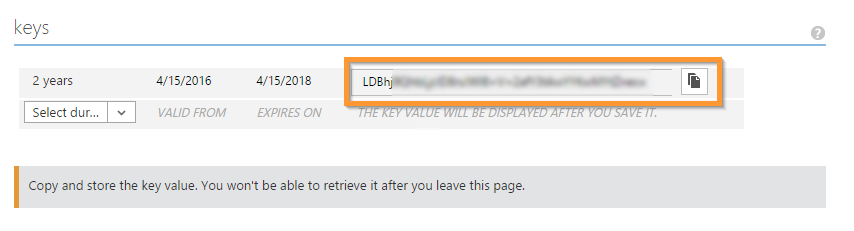
**ClientID**

This is the ID of the application you just created, you can find it on the Configuration-page for the app, where we added permissions.



**AppKey**

On the very same configuration page you can add an application key by choosing an option in the dropdown, when you save the configuration it will be displayed and you can copy it. You won't be able to see it again later, so copy it right away.

****

Fill all the values we took properly into app Key values. Once you updated you’re all the values then run below code to connect and get all logic apps available in your azure account.



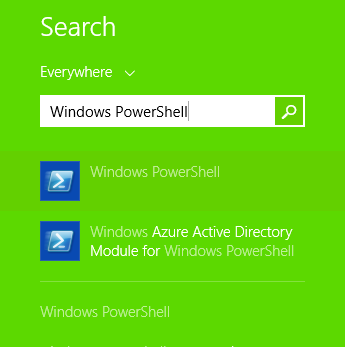
**Granting our application permissions**

If you fill in the values and try to run the application, you will get an error that looks something like this.

**AuthorizationFailed**: The client '<some-id>' with object id '<some-id>' does not have authorization to perform action ‘<Your action against Azure>’ over scope '<some-scope>'

That's because our application hasn't been granted permission to actually access our resource group. In order to grant permission, we turn to PowerShell.

Open Windows Powershell as adminstrator4



Login to Your Azure using following Command.

Login-AzureRmAccount

Once you executed above command it will open an Azure Login window so there you can give your azure username and password then click "signin" button or press enter.

First, select the subscription using below command that you want to access and change mode.

Select-AzureSubscription -SubscriptionId <subscription-id>

Then, assign the Reader-role to your application using below commmand (identified by the "some-id"-value from the error message we got) on the resource group.

New-AzureRmRoleAssignment -ObjectId '<some-id>' -ResourceGroupName Group -RoleDefinitionName Reader

If that goes well, you should be all set to run the application and list all the Logic Apps within that resource group.

For your reference find the code here. <https://github.com/M-Sivanraj/AzureAPIAccess_AD>.