SSIS to ADF Migration Deployment Document for Azure Func & Batch Acc

Deployment Steps

Code check Commit XXXXXX

Please take backup for existing objects for roll back.

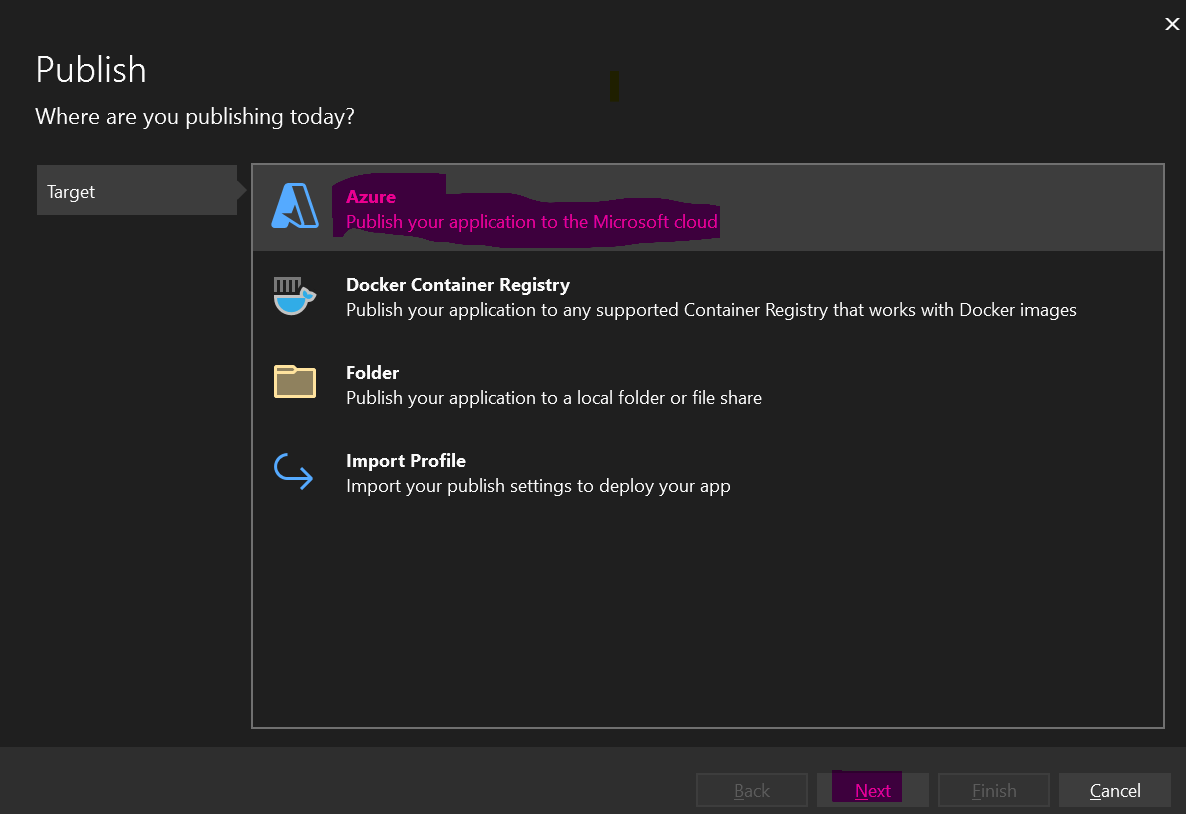
1. Create the below procedures – “SP\_UI\_EXPORT\_FIELDDEFINITIONS” and “sp\_export\_excel\_fieldoverload”



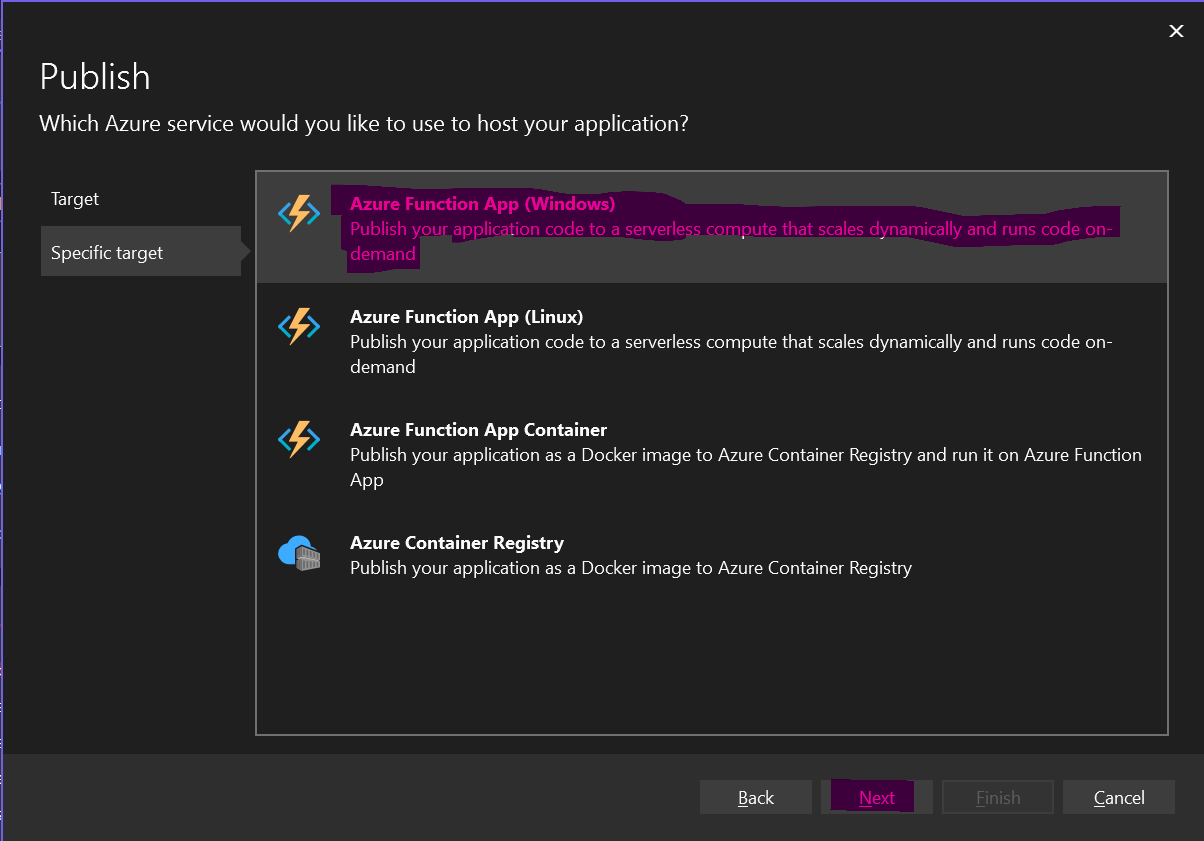
1. Azure Function Deployment (ZIP deployment) from **Visual Studio (TO BE IGNORED UNTIL THE FUNCTION IS CREATED IN AZURE PORTAL)**
   1. A new branch of the Azure function has been created as the old one was outdated and some of the code was not working because of that.

Any further changes that need to be done in the Azure Function needs to be done on “Neba\_Export\_V2”. If required, the old code can be removed on a later date.

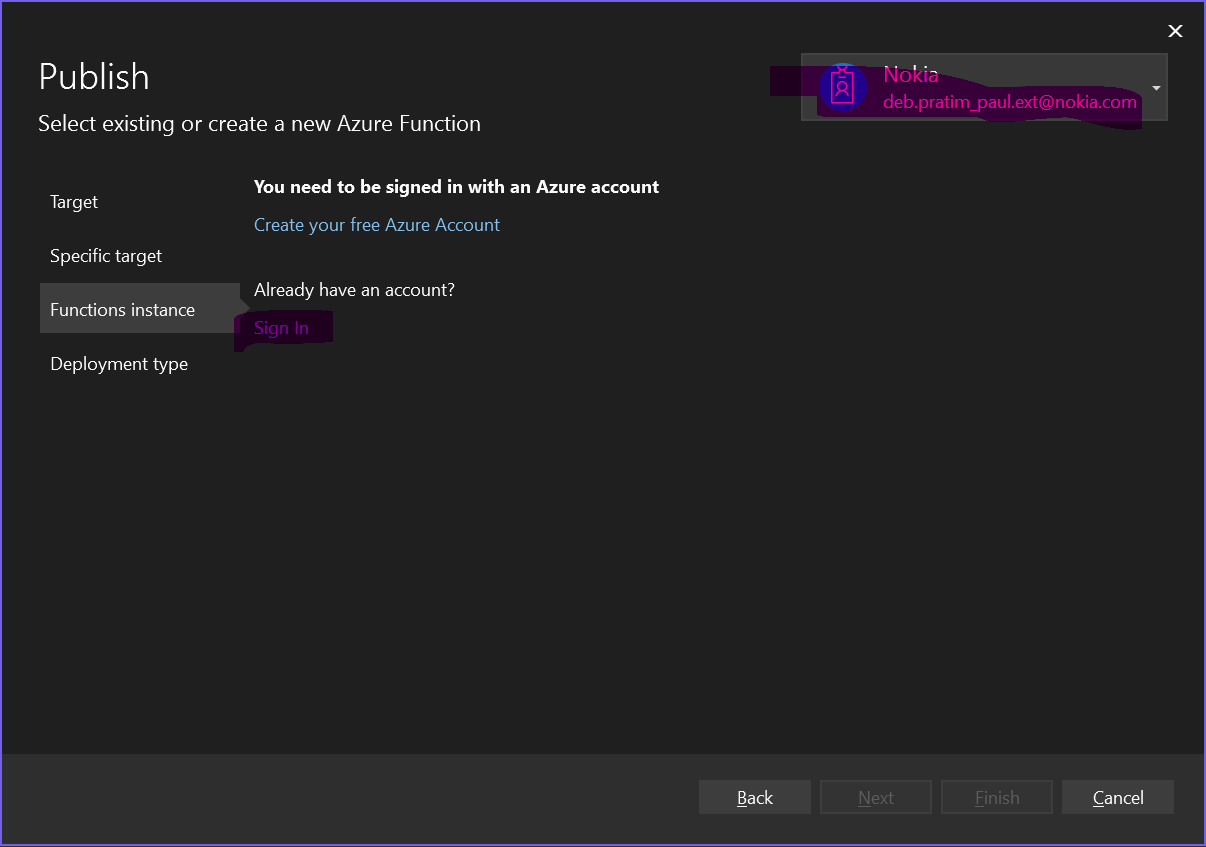
* 1. Once the latest code has been pulled, the user needs to right click on the project “Neba\_Export” and then select the “Publish” option.
  2. **First Time Deployment**
     1. The user needs to select the “**Azure**” option.



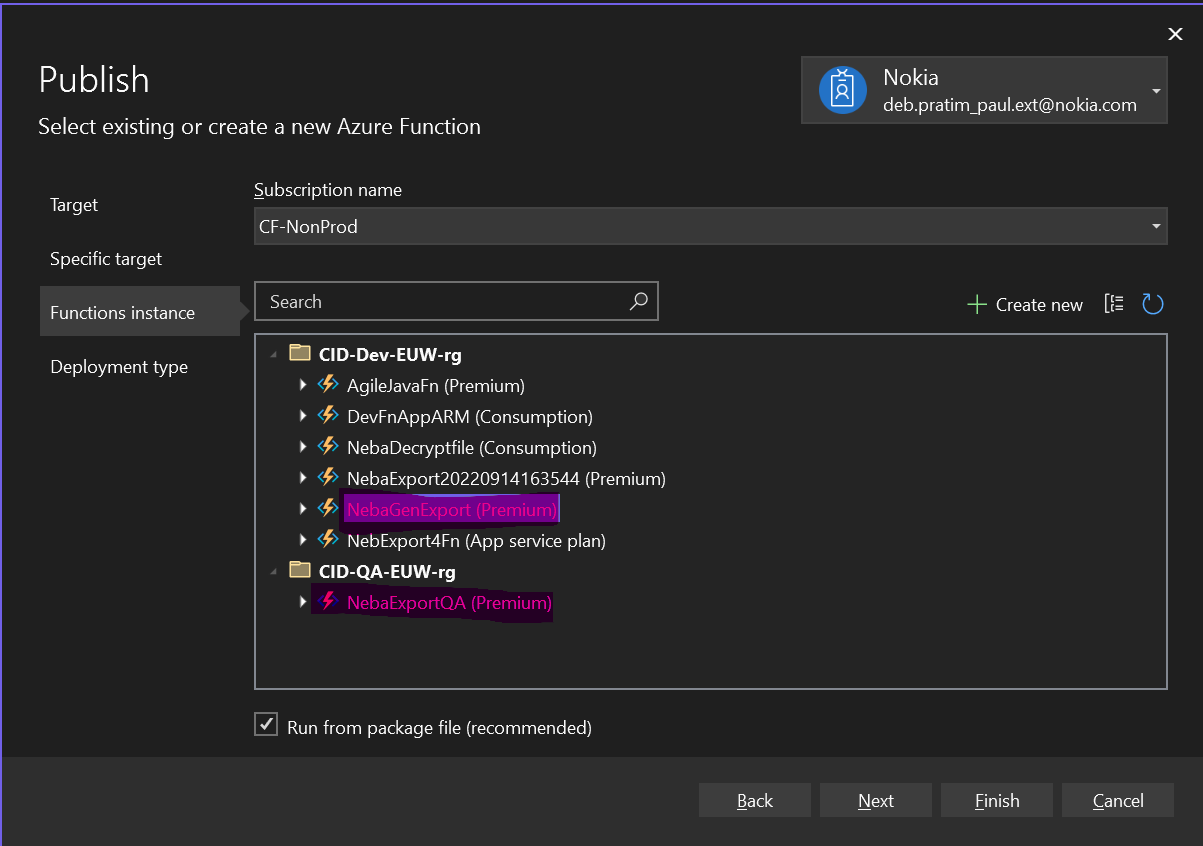
* + 1. On clicking the “Next” button the user needs to select the “Specific Target” from the window i.e., “Azure Function App (Windows)” and click on “Next”.



* + 1. In the next screen the user needs to authenticate by “AD Credentials” of Nokia.



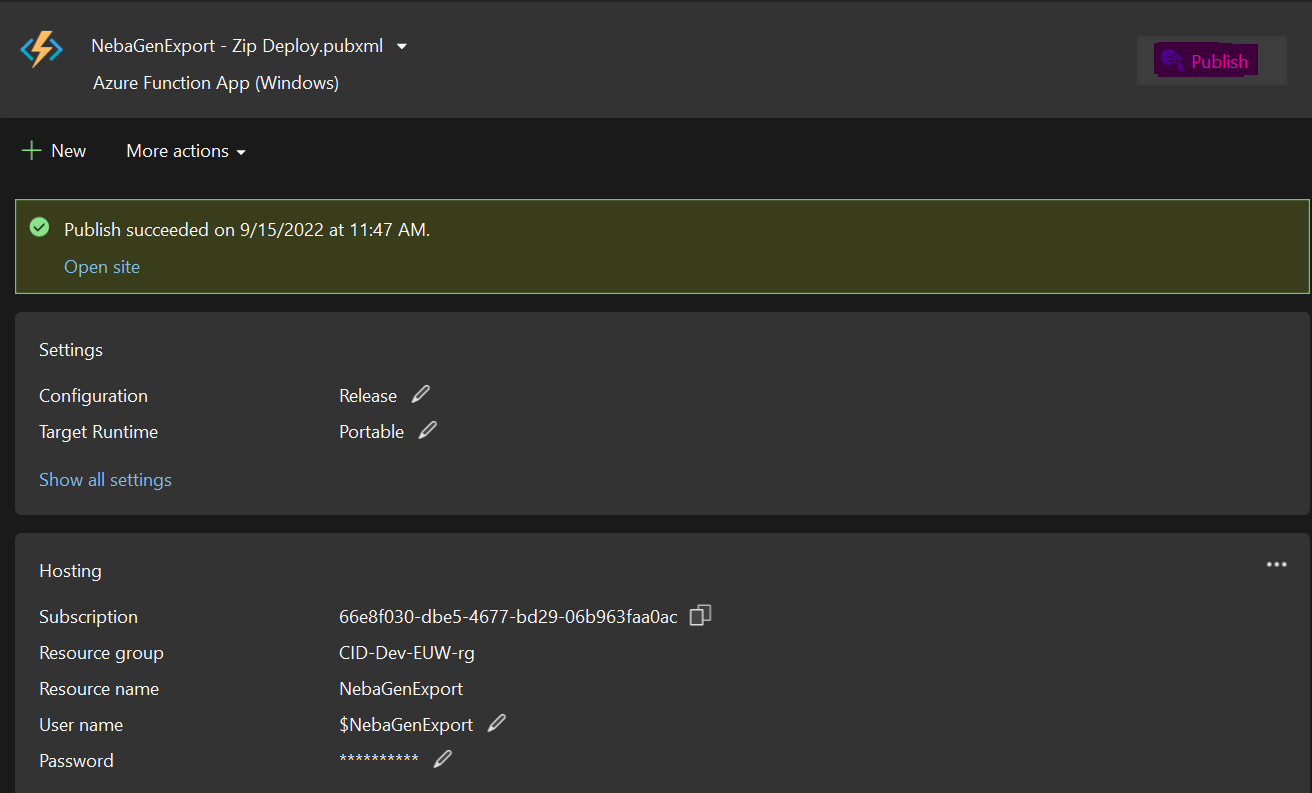
* + 1. Once the validation is done, the user will be able to view the list of Function’s that has been configured already in the “Azure Portal”.



* + 1. On selecting the valid function “NebaGenExport” \* (for Dev), click on “Finish” and the Visual Studio will be able to connect to the Azure Portal automatically and will also select the function as well.

**N.B.: \*Based on the environment the function name will change e.g., NebaGenExportV1 (Dev) and NebaGenExportQAV1 (Support/QA)**

* 1. For Users who has already configured the Visual Studio Publish can already just click on “Publish” option and the deployment will start.



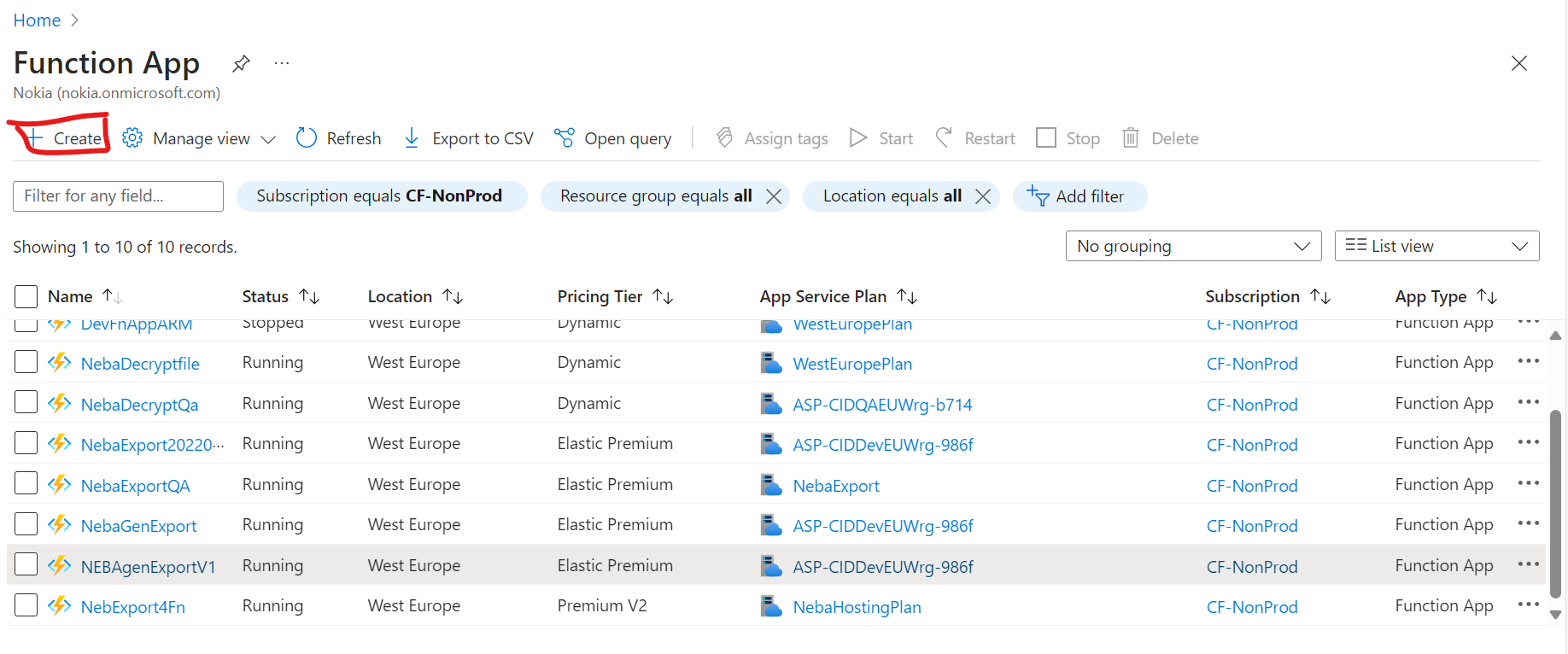
Please Note that once the deployment has started, the user needs to view the message “Function is ready” in the output window of the Visual Studio.

**Roll Back**

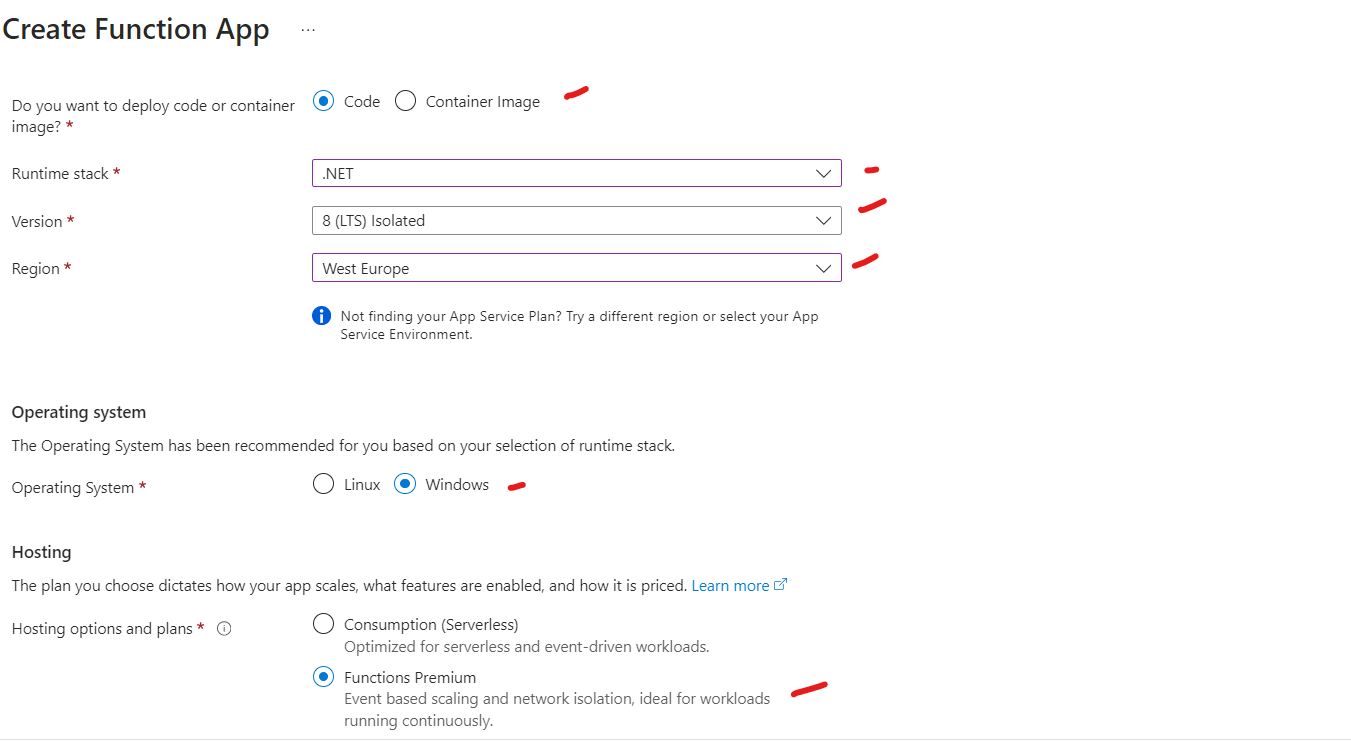
1. If the deployed code does not work then, we need to deploy the old version of the code from the DEVOPS.

**Creating the New Azure Function**

1. Click on “FunctionApp” in the <https://portal.azure.com/>
2. Click on the “Create” Button as show below –



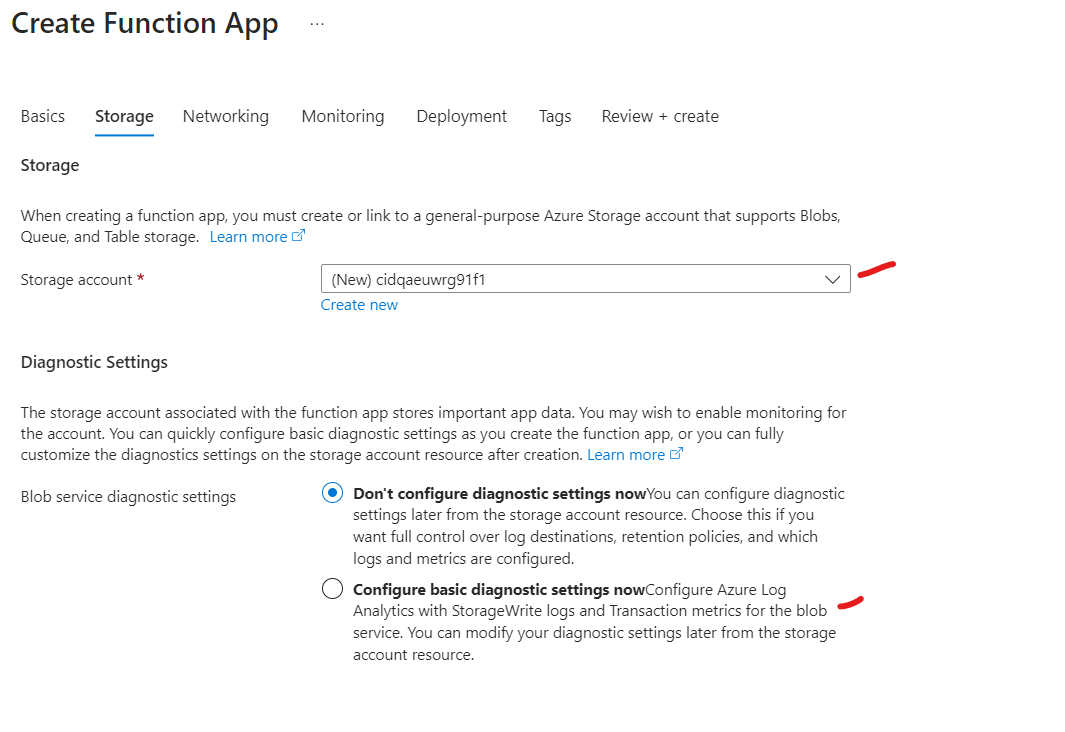
1. Select Subscription as “CF-NonProd” and Resource Group as “CID-QA-EUW-rg”. Provide the Function Name as “**NebaGenExportQAV1**” and fill the details as mentioned in screenshot ­­­below –



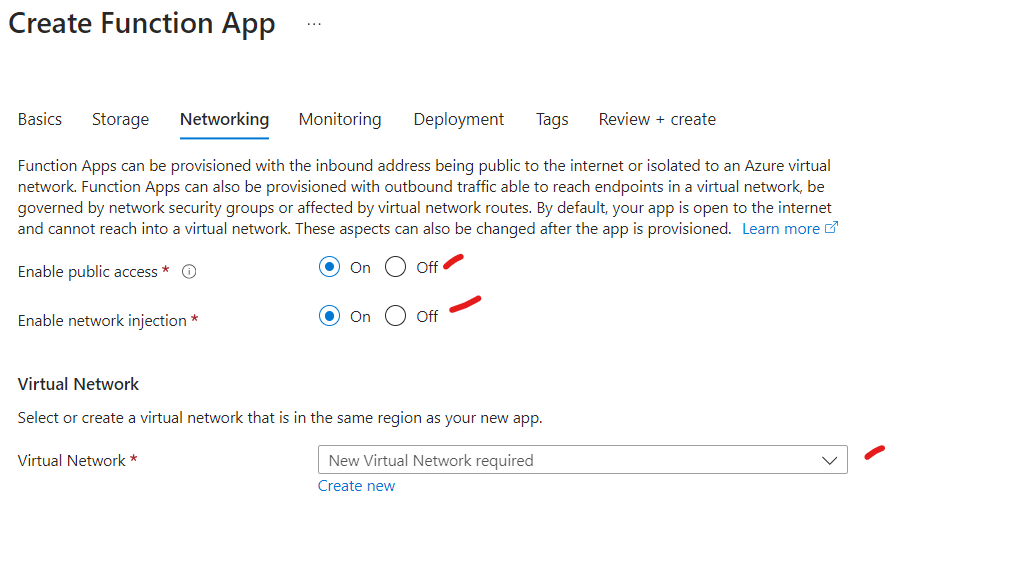
A screenshot of a computer

Description automatically generated

1. In the “Storage” section select the below options mentioned in the Screenshot below –



1. In the “Networking” Section select the option mentioned in the screenshot below –



Select the respective vNet from the options in dropdown.

1. In the “Monitoring” Section select the options mentioned in the screenshot below –

A screenshot of a computer

Description automatically generated

1. Skip the “Deployment” section and “Tags” section.
2. Once the Validation has passed on the “Review+Create” section “Create” button will be enabled, on clicking the option will create the function.

**N.B.: Select the options which are marked as “Red” in the screenshots.**

**Creating the New Batch Account**

1. Click on “Batch Accounts” in the <https://portal.azure.com/>
2. Click on “Create” button as mentioned below –

A screenshot of a chat

Description automatically generated

1. Select Subscription as “CF-NonProd” and Resource Group as “CID-QA-EUW-rg”. Provide the Function Name as “**cidqabatchstgaccount**” and fill the details as mentioned in screenshot ­­­below –

A screenshot of a computer

Description automatically generated

1. In the “Advanced” section select the options mentioned in the screenshot below –

A white background with black text

Description automatically generated

1. On Clicking on the “Create a private endpoint” in the “Networking” section a valid “Name” needs to be provided and proper VNet and Subnet needs to be selected from the dropdown including the “Private DNS Zone” as mentioned in the screenshot below –

A screenshot of a computer

Description automatically generated

1. Skip the “Tags” Section.
2. Once the Validation has passed on the “Review+Create” section “Create” button will be enabled, on clicking the option will create the function.

**N.B.: Select the options which are marked as “Red” in the screenshots.**

1. Once the Batch Account has been created select “Storage Account” from the left menu as mentioned in the screenshot below and click on “Select a Storage Account” –

A screenshot of a computer

Description automatically generated

1. Select “Pools” from the left-hand side of the screen and click on “Add” as mentioned in the screenshot below –

A white background with black text

Description automatically generated

1. Provide the below details as mentioned in the below screenshot –

A screenshot of a chat

Description automatically generated

1. In the “Start Task” Section select the options as mentioned in the screenshot below –

A screenshot of a computer

Description automatically generated

In the Command Line Textbox mention the below code

* powershell /c \"BootUpScript\_BatchStgAcc.ps1\"

Click on the Resource Files as mentioned in the screenshot below –

A screenshot of a computer

Description automatically generated

**N.B.: DO CHECK THE INCLUDE SAS CHECKBOX AND PROVIDE A PERIOD IN DAYS**

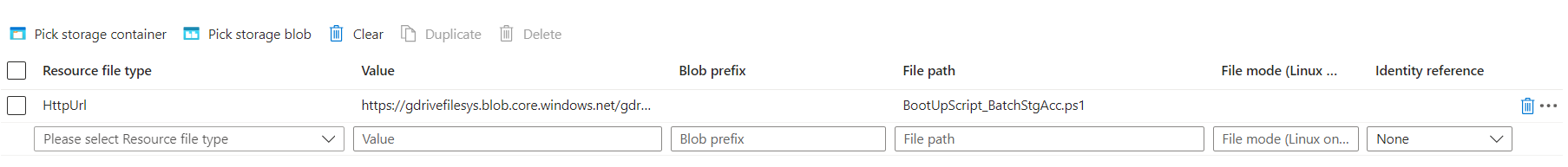
Select the required storage account from the list as mentioned in the screenshot below and select the PowerShell script folder location –

A screenshot of a computer

Description automatically generated

N.B.: The PowerShell script name should be same as the one mentioned above in the command line.

The path location should be looking like this once the selection has been done as mentioned below –



The VNet Section in the pool here can be ignored in the “Pools” section

A screenshot of a computer

Description automatically generated

Once done click on “Ok” to create the pool.

1. Once the pool has been created click on the pool name as shown in the screen and on the left-hand side there will be a “Nodes” option and validate the “Pagination Effort Limit” as mentioned below in the screenshot below –

A white background with black and red text

Description automatically generated

1. Validate the Start Task inside the Pools Section as mentioned in the screenshot below –

A screenshot of a computer

Description automatically generated

1. The following PowerShell script needs to be used and stored in the storage location and the same path needs to be provided in the Resource location mentioned above.



The current Path in development environment is -- **users/export/GenExport/SASTokenScript/**

1. Once the pools have been created it should be looking like this as mentioned below in the screenshot –

A white background with black text

Description automatically generated

Once all the deployment is done need to configure the Gen-Export ADF pipeline where the variable “SupportEmail” needs to be configured as per the environment requirements. Provided below is the screenshot for reference.

A screenshot of a computer

Description automatically generated