

Session 1: Microservices with Azure Function and Cosmos DB

Lab 1: Creating an Azure Function locally

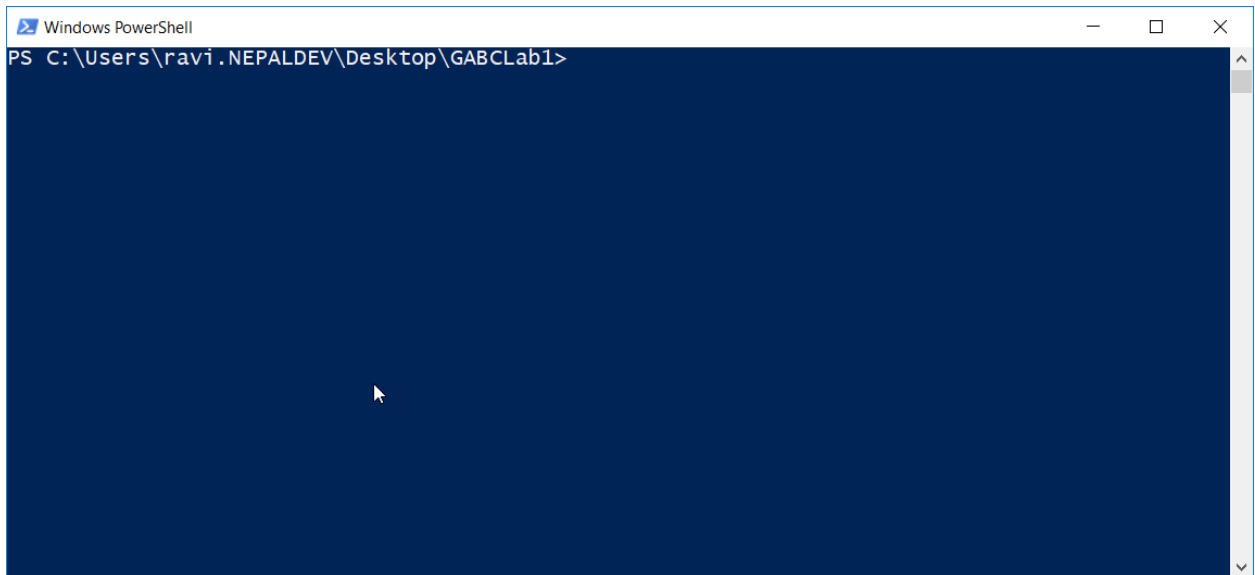
In this lab, we will learn how to create an Azure Function locally then later on deploy it to Microsoft Azure.

Pre-requisites:

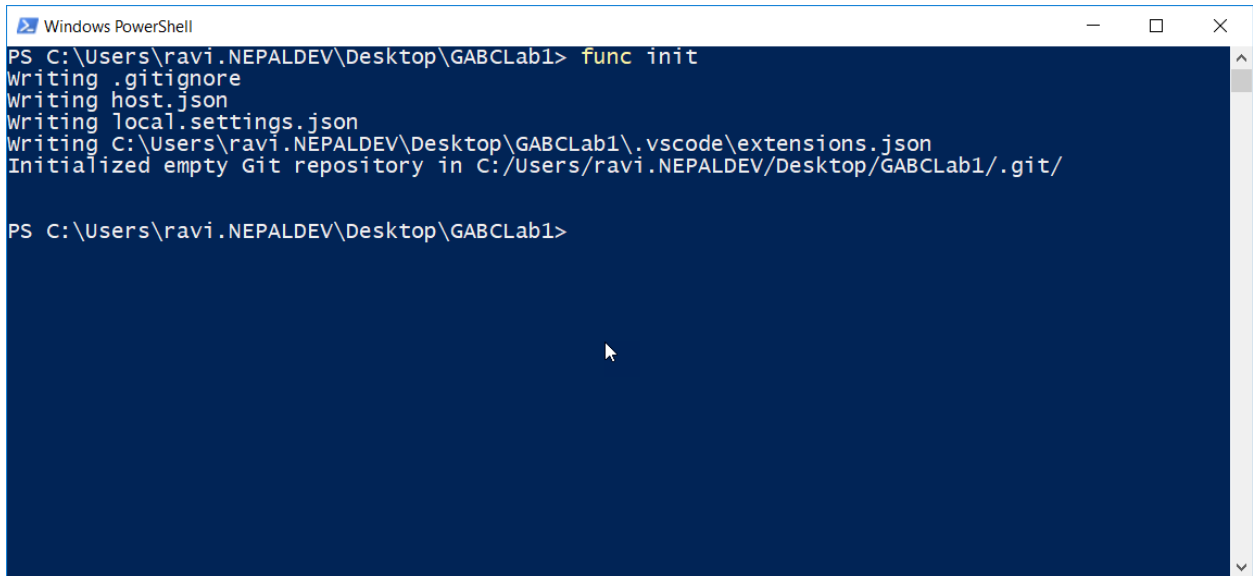
1. Azure Function CLI
2. Visual Studio Code
3. Active Azure Subscription
4. Azure Function Extension in Visual Studio Code

Step 1: Create a directory on your desired location where you want to save your function application

Step 2: Open terminal / command prompt in your desired location



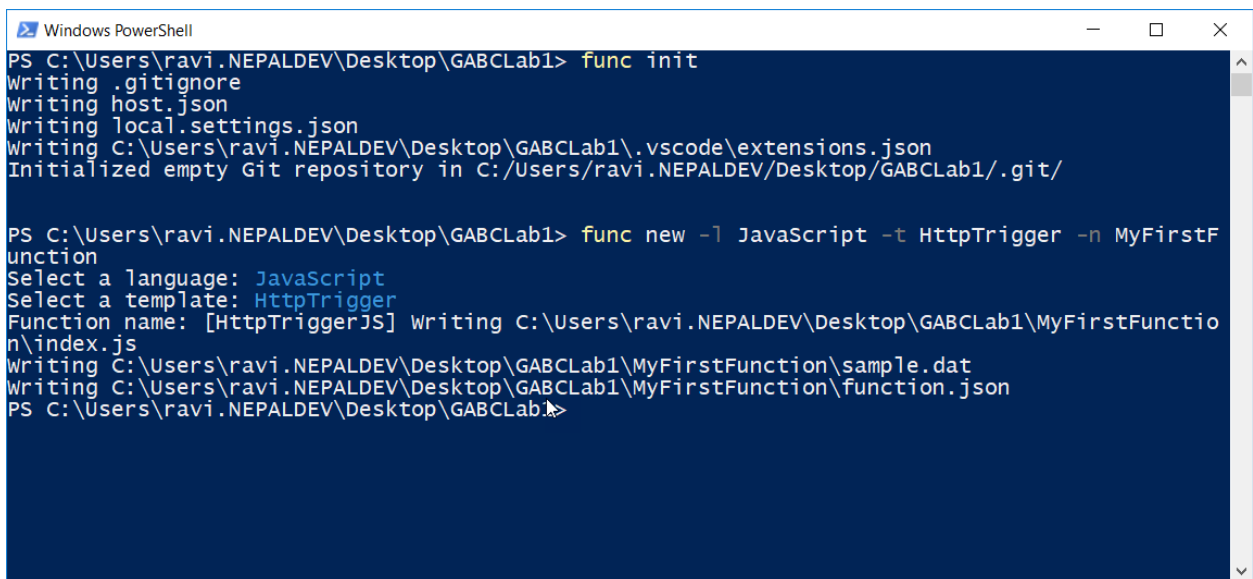
Step 3: Type command **func init** in the directory. This will create an function app in your directory



```
Windows PowerShell
PS C:\Users\ravi.NEPALDEV\Desktop\GABCLab1> func init
Writing .gitignore
Writing host.json
Writing local.settings.json
Writing C:\Users\ravi.NEPALDEV\Desktop\GABCLab1\.vscode\extensions.json
Initialized empty Git repository in C:/Users/ravi.NEPALDEV/Desktop/GABCLab1/.git/

PS C:\Users\ravi.NEPALDEV\Desktop\GABCLab1>
```

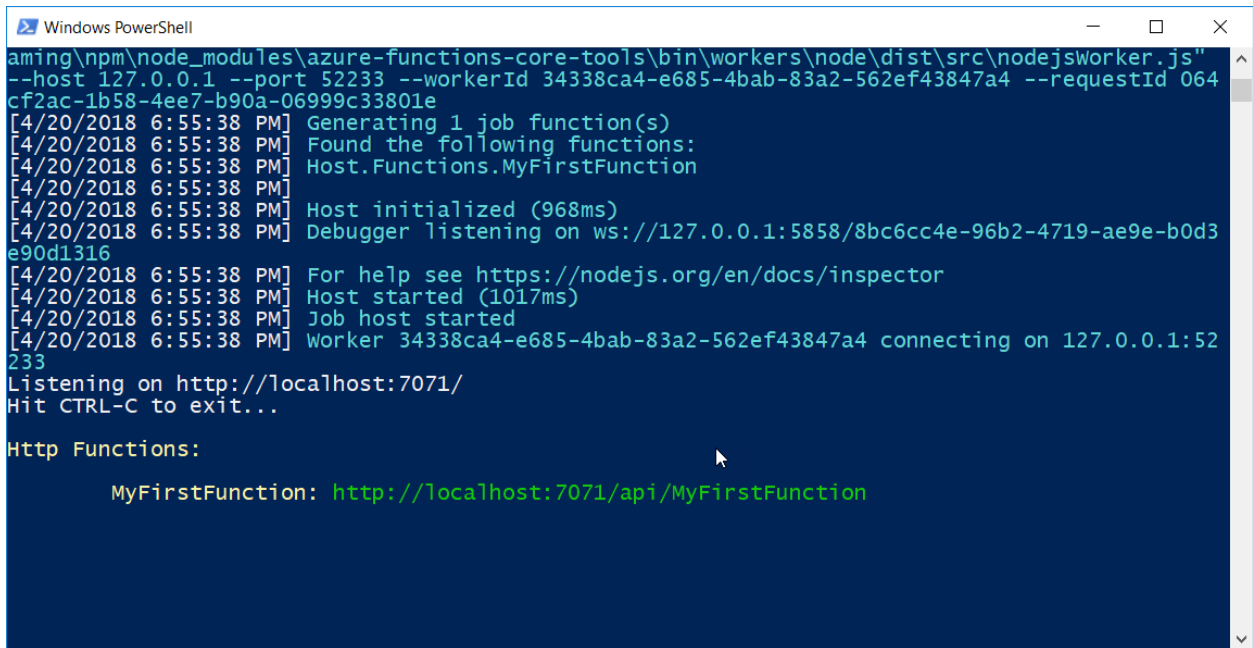
Step 4: Type command **func new -l JavaScript -t HttpTrigger -n MyFirstFunction**



```
Windows PowerShell
PS C:\Users\ravi.NEPALDEV\Desktop\GABCLab1> func init
Writing .gitignore
Writing host.json
Writing local.settings.json
Writing C:\Users\ravi.NEPALDEV\Desktop\GABCLab1\.vscode\extensions.json
Initialized empty Git repository in C:/Users/ravi.NEPALDEV/Desktop/GABCLab1/.git/

PS C:\Users\ravi.NEPALDEV\Desktop\GABCLab1> func new -l JavaScript -t HttpTrigger -n MyFirstFunction
Select a language: JavaScript
Select a template: HttpTrigger
Function name: [HttpTriggerJS] Writing C:\Users\ravi.NEPALDEV\Desktop\GABCLab1\MyFirstFunction\index.js
Writing C:\Users\ravi.NEPALDEV\Desktop\GABCLab1\MyFirstFunction\sample.dat
Writing C:\Users\ravi.NEPALDEV\Desktop\GABCLab1\MyFirstFunction\function.json
PS C:\Users\ravi.NEPALDEV\Desktop\GABCLab1>
```

Step 5: Type command **func host start**

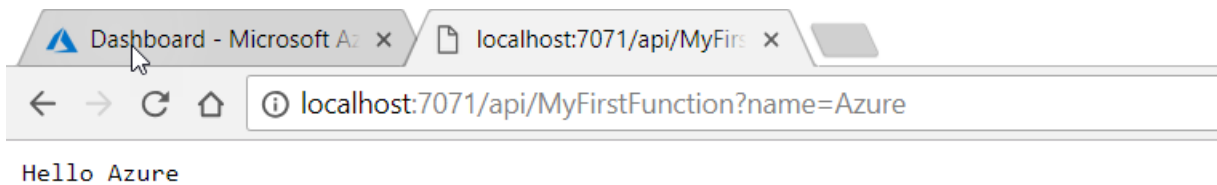


```
aming\npm\node_modules\azure-functions-core-tools\bin\workers\node\dist\src\nodejsworker.js"
--host 127.0.0.1 --port 52233 --workerId 34338ca4-e685-4bab-83a2-562ef43847a4 --requestId 064
cf2ac-1b58-4ee7-b90a-06999c33801e
[4/20/2018 6:55:38 PM] Generating 1 job function(s)
[4/20/2018 6:55:38 PM] Found the following functions:
[4/20/2018 6:55:38 PM] Host.Functions.MyFirstFunction
[4/20/2018 6:55:38 PM] Host initialized (968ms)
[4/20/2018 6:55:38 PM] Debugger listening on ws://127.0.0.1:5858/8bc6cc4e-96b2-4719-ae9e-b0d3
e90d1316
[4/20/2018 6:55:38 PM] For help see https://nodejs.org/en/docs/inspector
[4/20/2018 6:55:38 PM] Host started (1017ms)
[4/20/2018 6:55:38 PM] Job host started
[4/20/2018 6:55:38 PM] Worker 34338ca4-e685-4bab-83a2-562ef43847a4 connecting on 127.0.0.1:52
233
Listening on http://localhost:7071/
Hit CTRL-C to exit...

Http Functions:

    MyFirstFunction: http://localhost:7071/api/MyFirstFunction
```

Step 6: Open <http://localhost:7071/api/MyFirstFunction?name=Azure> in browser and test it



Congratulations, you created your first function application!

Now, open the code in your favorite text editor and let's see how it is structured.