

## Workshop - Proctor's Guide - athenahealth Austin March 2019

### Pre-event: Lab Environment Setup

- This hack will require a centralized environment to host a FHIR server that is prepopulated with patient data.
- We will be using the new Azure API for FHIR managed service and then populating it with dummy data.
- Install the Azure CLI if you haven't already.
  - For windows OS, use bash shell in Windows Subsystem for Linux (see WLS install under tool-set below)
- Install the latest version of nodejs (at least 10.x) on your machine, if using Windows, use the bash shell in the Windows Subsystem for Linux

Run in bash shell:

```
$ sudo apt-get update
```

Output:

```
rilian@RLIANGSB:~$ sudo apt-get update
Get:1 http://security.ubuntu.com/ubuntu bionic-security InRelease [88.7 kB]
Hit:2 http://archive.ubuntu.com/ubuntu bionic InRelease
Get:3 http://archive.ubuntu.com/ubuntu bionic-updates InRelease [88.7 kB]
...
42% [13 Packages 4324 kB/8570 kB 50%]
...
Get:28 <http://archive.ubuntu.com/ubuntu bionic-backports/universe> Translation-en [1604 B]
Fetched 16.3 MB in 35s (469 kB/s)
Reading package lists... Done
rilian@RLIANGSB:~$
```

```
$ sudo apt-get install jq
```

Output:

```
rilian@RLIANGSB:~$ sudo apt install jq
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following package was automatically installed and is no longer required:
  libfreetype6
Use 'sudo apt autoremove' to remove it.
The following additional packages will be installed:
  libjq1 libonig4
The following NEW packages will be installed:
  jq libjq1 libonig4
0 upgraded, 3 newly installed, 0 to remove and 173 not upgraded.
Need to get 276 kB of archives.
After this operation, 930 kB of additional disk space will be used.
Do you want to continue? [Y/n] Y
0% [Connecting to archive.ubuntu.com]
...
Get:1 <http://archive.ubuntu.com/ubuntu bionic/universe> amd64 libonig4 amd64 6.7.0-1 [119 kB]
Get:2 <http://archive.ubuntu.com/ubuntu bionic/universe> amd64 libjq1 amd64 1.5+dfsg-2 [111 kB]
Get:3 <http://archive.ubuntu.com/ubuntu bionic/universe> amd64 jq amd64 1.5+dfsg-2 [45.6 kB]
Fetched 276 kB in 11s (24.3 kB/s)
Selecting previously unselected package libonig4:amd64.
(Reading database ... 28490 files and directories currently installed.)
Preparing to unpack .../libonig4_6.7.0-1_amd64.deb ...
Unpacking libonig4:amd64 (6.7.0-1) ...
Selecting previously unselected package libjq1:amd64.
```

```

Preparing to unpack .../libjq1_1.5+dfsg-2_amd64.deb ...
Unpacking libjq1:amd64 (1.5+dfsg-2) ...
Selecting previously unselected package jq.
Preparing to unpack .../jq_1.5+dfsg-2_amd64.deb ...
Unpacking jq (1.5+dfsg-2) ...
Setting up libonig4:amd64 (6.7.0-1) ...
Setting up libjq1:amd64 (1.5+dfsg-2) ...
Processing triggers for libc-bin (2.27-3ubuntu1) ...
Processing triggers for man-db (2.8.3-2) ...
Setting up jq (1.5+dfsg-2) ...
rilian@RLIANGSB:~$

```

Optional:

```
sudo add-apt-repository universe
```

```
sudo apt-get update
```

### Configure FHIR server

- Create a new service principal, run `$ az ad sp create-for-rbac -o json`

Log into Azure first, run

```
$ az login
```

Output:

```
C:\Users\rilian>az login
```

Note, we have launched a browser for you to login. For old experience with device code, use "az login --use-device-code"  
You have logged in. Now let us find all the subscriptions to which you have access...

```

[
  {
    "cloudName": "AzureCloud",
    "id": "327ead23-9a0f-4d49-a37f-9c9dda2818d4",
    "isDefault": true,
    "name": "Microsoft_AIRS",
    "state": "Enabled",
    "tenantId": "72f988bf-86f1-41af-91ab-2d7cd011db47",
    "user": {
      "name": "rilian@microsoft.com",
      "type": "user"
    }
  }
]

```

Create new Service Principal, run

```
$ az ad sp create-for-rbac -o json
```

Output:

```
C:\Users\rilian>az ad sp create-for-rbac -o json
```

```
Retrying role assignment creation: 1/36
```

```
Retrying role assignment creation: 2/36
```

```

{
  "appId": "e8105de9-72f7-46ef-b787-e2be47a325d0",
  "displayName": "azure-cli-2019-03-22-12-18-42",
  "name": "http://azure-cli-2019-03-22-12-18-42",
  "password": "766268ed-a688-43f5-a0ff-d6ad24f27a74",
  "tenant": "72f988bf-86f1-41af-91ab-2d7cd011db47"
}

```

- Copy "appid" and "password" from output for use later

```
appid=e8105de9-72f7-46ef-b787-e2be47a325d0
```

password=766268ed-a688-43f5-a0ff-d6ad24f27a74

- Find "objectid" of this new Service Principal, run `$ az ad sp show --id {appid of the new SP} -o json | jq -r . objectId`

Install jq on Windows, run in PowerShell

<https://chocolatey.org/packages/jq>

`$ choco install jq`

Output:

`PS C:\WINDOWS\system32> choco install jq`

Chocolatey v0.10.13

2 validations performed. 1 success(es), 1 warning(s), and 0 error(s).

Validation Warnings:

- A pending system reboot request has been detected, however, this is being ignored due to the current Chocolatey configuration. If you want to halt when this occurs, then either set the global feature using:

`choco feature enable -name=exitOnRebootDetected`  
or pass the option `--exit-when-reboot-detected`.

Installing the following packages:

jq

By installing you accept licenses for the packages.

Progress: Downloading jq 1.5... 100%

jq v1.5 [Approved]

jq package files install completed. Performing other installation steps.

The package jq wants to run 'chocolateyinstall.ps1'.

Note: If you don't run this script, the installation will fail.

Note: To confirm automatically next time, use '-y' or consider:

`choco feature enable -n allowGlobalConfirmation`

Do you want to run the script?([Y]es/[N]o/[P]rint): Y

WARNING: Url has SSL/TLS available, switching to HTTPS for download

Downloading jq 64 bit

from '<https://github.com/stedolan/jq/releases/download/jq-1.5/jq-win64.exe>'

Progress: 100% - Completed download of C:\ProgramData\chocolatey\lib\jq\tools\jq.exe (2.22 MB).

Download of jq.exe (2.22 MB) completed.

C:\ProgramData\chocolatey\lib\jq\tools\jq.exe

WARNING: Write-ChocolateySuccess is deprecated and will be removed in v2. If you are the maintainer, please remove it from your package file.

ShimGen has successfully created a shim for jq.exe

The install of jq was successful.

Software install location not explicitly set, could be in package or default install location if installer.

Chocolatey installed 1/1 packages.

See the log for details (C:\ProgramData\chocolatey\logs\chocolatey.log).

`$ az ad sp show --id e8105de9-72f7-46ef-b787-e2be47a325d0 -o json | jq -r .objectId`

Output:

`PS C:\WINDOWS\system32> az ad sp show --id e8105de9-72f7-46ef-b787-e2be47a325d0 -o json | jq -r .objectId`  
`a7bdbd85-18b3-4b15-ac19-4c997854884a`

- Deploy ARM template for the new FHIR service:
  - Create resource group in "NorthCentralUS" (supported region for FHIR service preview), run `$ az group create --name myRG --location northcentralus`
- `$ az group create --name myRG --location northcentralus`

## Output:

```
PS C:\MyWork\TE\Clients\Athena Health\Hackathon\Student and Host Files\0.0 - Hack Setup> az group create --name myRG --location northcentralus
```

```
{
  "id": "/subscriptions/327ead23-9a0f-4d49-a37f-9c9dda2818d4/resourceGroups/myRG",
  "location": "northcentralus",
  "managedBy": null,
  "name": "myRG",
  "properties": {
    "provisioningState": "Succeeded"
  },
  "tags": null
}
```

- Run the template and use the "objectId" from above,  
\$ az group deployment create --template-file azuredeploy.json --parameters azuredeploy.parameters.json --parameters accessPolicyObjectId={objectId of SP} -g Athena-Hack --no-wait

**Note:** Need to update azuredeploy.parameters.json file to use all lower case name value ("myserver").

```
$ az group deployment create --template-file azuredeploy.json --parameters
azuredeploy.parameters.json --parameters accessPolicyObjectId=a7bdbd85-18b3-4b15-
ac19-4c997854884a -g Athena-Hack --no-wait
```

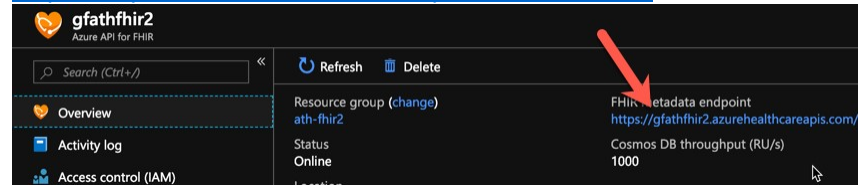
## Output:

```
PS C:\> cd "C:\MyWork\TE\Clients\Athena Health\Hackathon\Student and Host Files\0.0 - Hack Setup"
PS C:\MyWork\TE\Clients\Athena Health\Hackathon\Student and Host Files\0.0 - Hack Setup> az group deployment
create --template-file azuredeploy.json --parameters azuredeploy.parameters.json --parameters
accessPolicyObjectId=a7bdbd85-18b3-4b15-ac19-4c997854884a -g myRG --no-wait
PS C:\MyWork\TE\Clients\Athena Health\Hackathon\Student and Host Files\0.0 - Hack Setup>
```

- Monitor deployment progress from "Deployments" blade of the resource group
- Configure the node.js data generation app:
  - We'll be adding a new configuration section to the "config.json" file.
  - Copy and paste one of the pre-existing environments and change values that are different from the "default" configuration. Typically, these values include:
    - "tenant": The Azure AD tenant in which you created the service principal above > microsoft.onmicrosoft.com , tenant id: 72f988bf-86f1-41af-91ab-2d7cd011db47
    - "applicationId": The "appId" value we saved during the creation of the service principal. > e8105de9-72f7-46ef-b787-e2be47a325d0
    - "clientSecret": The "password" value we saved during the creation of the service principal. > 766268ed-a688-43f5-a0ff-d6ad24f27a74
    - "fhirApiUrl": The URL to the FHIR service created above. You can find

this on the Overview blade of the Azure API for FHIR resource:

<https://myserver.azurehealthcareapis.com/metadata>



Output:

```
"richard1": {
  "authorityHostUrl": "https://login.windows.net",
  "tenant": "72f988bf-86f1-41af-91ab-2d7cd011db47",
  "applicationId": "db0e3ced-58db-4750-9749-7817dd4ee4d5",
  "clientSecret": "766268ed-a688-43f5-a0ff-d6ad24f27a74",
  "resource": "https://azurehealthcareapis.com",
  "fhirApiUrl": "https://myserver.azurehealthcareapis.com/metadata"
}
```

### (Optional) Setup Postman to access FHIR API

Postman Global VAR: Azure REST API Env Vars

\*\*\*\*\*

tenant\_id: 72f988bf-86f1-41af-91ab-2d7cd011db47

grant\_type: client\_credentials

client\_id: fba6119e-4c34-4d70-b65c-57a9dac1edce

client\_secret: 004208d3-f172-4d00-972d-69aef37e2bc2

resource: <http://management.azure.com>

subscriptionid: 327ead23-9a0f-4d49-a37f-9c9dda2818d4

Auth URL: [https://login.microsoftonline.com/{TENANT-ID}/oauth2/authorize?](https://login.microsoftonline.com/{TENANT-ID}/oauth2/authorize?resource=<audience>)

[resource=<audience>](https://login.microsoftonline.com/{TENANT-ID}/oauth2/authorize?resource=<audience>)

{TenantID}=72f988bf-86f1-41af-91ab-2d7cd011db47

{audience} (Azure API for FHIR)=<https://azurehealthcareapis.com>

AuthURL=<https://login.microsoftonline.com/72f988bf-86f1-41af-91ab-2d7cd011db47/oauth2/authorize?resource=https://azurehealthcareapis.com>

Get New Access Token

Get new access token, click

## GET NEW ACCESS TOKEN



Request a new access token to add it to your list of tokens

On clicking Request Token, you will be redirected to the Auth URL where you can enter the user's credentials and request for a token

Callback URL `https://www.getpostman.com/oauth2/callback`

Set this as the callback URL in your app settings page.

Token Name `myserver FHIR REST API Bearer Token`

Auth URL

`https://login.microsoftonline.com/72f988bf-86f1-41af-91ab-2d7cd011db47/oauth2/authorize?resource=https://azurehealthcareapis.com`

Access Token URL

`https://login.microsoftonline.com/72f988bf-8...`

Client ID

`e8105de9-72f7-46ef-...`

Client Secret

`766268ed-a688-43f5-...`

Scope (Optional)

Grant Type

Authorization Code

☐

Request access token locally

Cancel

Request Token

- Prompt for sign into Azure



## Sign in

Email, phone, or Skype

---

[Can't access your account?](#)

Next

### Output:

*Request Id: 222bf9bc-a58f-4e71-b35d-2bea11753d00*

*Correlation Id: e744362f-44d3-4840-9416-8120340de117*

*Timestamp: 2019-03-25T13:02:21Z*

*Message: AADSTS700016: Application with identifier 'db0e3ced-58db-4750-9749-7817dd4ee4d5' was not found in the directory '72f988bf-86f1-41af-91ab-2d7cd011db47'. This can happen if the application has not been installed by the administrator of the tenant or consented to by any user in the tenant. You may have sent your authentication request to the wrong tenant.*

*Advanced diagnostics: Enable*

*If you plan on getting support for an issue, turn this on and try to reproduce the error. This will collect additional information that will help troubleshoot the issue.*

*Request Id: 21ea6475-5c9c-42f9-91b6-7a29b0cb5e00*

*Correlation Id: 04f61217-7595-4d2a-a1c8-22d7c8094115*

*Timestamp: 2019-03-25T13:09:46Z*

*Message: AADSTS650057: Invalid resource. The client has requested access to a resource which is not listed in the requested permissions in the client's application registration. Client app ID:*

*eadec9c3-4146-4f24-8045-2c6b05802b4f(<https://rilianfhir1.azurehealthcareapis.com>). Resource value from request: <https://azurehealthcareapis.com>. Resource app ID: 4f6778d8-5aef-43dc-a1ff-b073724b9495. List of valid resources from app registration: 00000003-0000-0000-c000-000000000000.*

- Run the data generation app to insert patient records
  - Download all dependencies, run **\$ npm install** at the root folder

Run in cmd in project working directory:

**\$ cd c:\MyWork\TE\Clients\Athena Health\Hackathon\Student and Host Files\0.0 - Hack Setup**

**\$ npm install**

Output:

*C:\MyWork\TE\Clients\Athena Health\Hackathon\Student and Host Files\0.0 - Hack Setup>npm install*

*npm WARN deprecated node-uuid@1.4.8: Use uuid module instead*

*npm WARN deprecated hoek@0.9.1: The major version is no longer supported. Please update to 4.x or newer*

*> sleep@6.0.0 install C:\MyWork\TE\Clients\Athena Health\Hackathon\Student and Host Files\0.0 - Hack Setup\node\_modules\sleep*

*> node-gyp rebuild*

*C:\MyWork\TE\Clients\Athena Health\Hackathon\Student and Host Files\0.0 - Hack Setup\node\_modules\sleep>if not defined*

```
npm_config_node_gyp (node "C:\Program Files\nodejs\node_modules\npm\node_modules\npm-lifecycle\node-gyp-bin\..\..\node_modules\node-gyp\bin\node-gyp.js" rebuild ) else (node "C:\Program Files\nodejs\node_modules\npm\node_modules\node-gyp\bin\node-gyp.js" rebuild )
```

Building the projects in this solution one at a time. To enable parallel build, please add the "/m" switch.

```
sleep_cpp11.cc
```

```
module_init.cc
```

```
sleep_posix.cc
```

```
sleep_win.cc
```

```
win_delay_load_hook.cc
```

```
Creating library C:\MyWork\TE\Clients\Athena Health\Hackathon\Student and Host Files\0.0 - Hack Setup\node_modules\sleep\build\Release\node_sleep.lib and object C:\MyWork\TE\Clients\Athena Health\Hackathon\Student and Host Files\0.0 - Hack Setup\node_modules\sleep\build\Release\node_sleep.exp
```

```
Generating code
```

```
All 118 functions were compiled because no usable IPDB/IOBJ from previous compilation was found.
```

```
Finished generating code
```

```
node_sleep.vcxproj -> C:\MyWork\TE\Clients\Athena Health\Hackathon\Student and Host Files\0.0 - Hack Setup\node_modules\sleep\build\Release\node_sleep.node
```

```
npm WARN fhirDataMgmt@1.0.0 No description
```

```
npm WARN fhirDataMgmt@1.0.0 No repository field.
```

```
added 41 packages from 45 contributors and audited 130 packages in 30.842s
```

```
found 8 moderate severity vulnerabilities
```

```
run `npm audit fix` to fix them, or `npm audit` for details
```

Setting NODE\_ENV:

```
$ Set NODE_ENV=<environmentname>
```

Output

```
C:\MyWork\TE\Clients\Athena Health\2019-03 - Austin PaaS Hackathon\Student and Host Files\0.0 - Hack Setup>set
```

```
NODE_ENV=richard2
```

(Optional) Install npm module dotenv to configure environment via .env file

- Create ".env" file in project directory with the following content:

```
"NODE_ENV=richard1"
```

- Install dotenv npm module:

```
$ npm install dotenv --save
```

Output:

```
PS C:\MyWork\TE\Clients\Athena Health\2019-03 - Austin PaaS Hackathon\Student and Host Files\0.0 - Hack Setup> npm install dotenv --save
```

```
npm WARN fhirDataMgmt@1.0.0 No description
```

```
npm WARN fhirDataMgmt@1.0.0 No repository field.
```

```
+ dotenv@7.0.0
```

```
updated 1 package and audited 131 packages in 4.182s
```

```
found 8 moderate severity vulnerabilities
```

```
run `npm audit fix` to fix them, or `npm audit` for details
```

- Install FHIR npm library

```
$ npm install fhir
```

Output:

```
PS C:\MyWork\TE\Clients\Athena Health\2019-03 - Austin PaaS Hackathon\Student and Host Files\0.0 - Hack Setup> npm install fhir
```

```
npm WARN fhirDataMgmt@1.0.0 No description
```

```
npm WARN fhirDataMgmt@1.0.0 No repository field.
```

```
+ fhir@4.5.0
```

```
added 13 packages from 27 contributors and audited 145 packages in 14.584s
```



found 8 moderate severity vulnerabilities  
run `npm audit fix` to fix them, or `npm audit` for details

- Run the **datagen.js** app to loop and create endless # of patients. Press Ctrl-C when enough records have been created

First, set environment to "richard2" in config.json

\$ Set NODE\_ENV=richard2

Run \$ node datagen.js

```
Created patient 0: c33672f4-4363-442a-9fba-70a4eb009060
Created patient 1: 6030d8d7-865a-48cf-9840-5658f864b2bc
Created patient 2: 63ad88e5-11c4-444f-beaa-decf8297cc87
Created patient 3: def529b7-49bb-4281-ba4c-6cb0f4efca56
Created patient 4: 98632425-5842-4731-bb97-b6c9197a35ae
Created patient 5: cb3c312d-c60c-4a3f-9ac4-8a3b8883dfad
Created patient 6: b8a2473d-083b-4e3b-91db-dcee5fc40642
Created patient 7: 07dbdd33-4873-428e-90d4-7e971f20efd2
Created patient 8: 6cb2ef62-2427-4666-ac5f-e0177df531f8
Created patient 9: 4e3c4501-d085-4905-afa2-3cc8f8e214d7
Created patient 10: d9e8d4c3-0ad6-454b-93a5-3d8a01db7e9a
Created patient 11: 33f9b0b2-2b40-4705-ad90-34e0bc036519
```

Output:

C:\MyWork\TE\Clients\Athena Health\Hackathon\Student and Host Files\0.0 - Hack Setup>node datagen.js

Created patient 0: 6c41dbcc-7de8-46af-88f6-42e10986ecb2

Created patient 1: 9f9ff091-88f0-4809-b2be-4914dfb03a49

Created patient 2: be9822cb-5bf3-4d88-ba03-8f15ff0dc30e

Created patient 3: 6a045c01-2cd4-493d-983d-6cc739fbd1a0

Created patient 4: 1bf674cc-a3b7-427e-a9c6-80de2d36008e

Created patient 5: 1c44e2cb-2620-4396-9240-40d658f72d29

...

Created patient 1049: 09e8aeb4-de49-49ae-850f-1f5bf71edba6

^C

C:\MyWork\TE\Clients\Athena Health\Hackathon\Student and Host Files\0.0 - Hack Setup>

- Run the dataread.js app to make sure the new patients can be read.

Note: These records are read in pages of 100 at a time.

Run \$ node dataread.js

```
0: 6030d8d7-865a-48cf-9840-5658f864b2bc: Larson, Trisha
1: def529b7-49bb-4281-ba4c-6cb0f4efca56: Fritsch, Alphonso
2: b8a2473d-083b-4e3b-91db-dcee5fc40642: Medhurst, Cleve
3: d9e8d4c3-0ad6-454b-93a5-3d8a01db7e9a: Smitham, Jalon
4: 33f9b0b2-2b40-4705-ad90-34e0bc036519: Schroeder, Edwardo
5: a2d3ccba-f36b-46dc-81e6-397de597da97: Schoen, Noel
6: 41c48110-f35c-4fd6-b787-49c53f596f3c: O'Conner, Madelyn
7: be4f9f92-5343-4309-a108-47326cae4d85: O'Conner, Willa
8: a81b5ebc-0c76-44a0-b9d6-1afa77884ba6: Powlowski, Lucas
9: 84f6be64-dc40-461b-b61f-8b00381afe33: Oberbrunner, Russel
10: c33672f4-4363-442a-9fba-70a4eb009060: Rowe, Darion
11: 1f3cd9f1-4e17-4b85-86af-74470f0d673e: Mertz, Rory
12: 4e3c4501-d085-4905-afa2-3cc8f8e214d7: Ankunding, Modesta
13: 07dbdd33-4873-428e-90d4-7e971f20efd2: Wolf, Rasheed
14: e7b8e138-5aae-4a91-8576-a08ccc44692: Olson, Gay
15: 009cad55-ffa8-4375-933e-77d6325ba824: Pollich, Ezekiel
16: cb3c312d-c60c-4a3f-9ac4-8a3b8883dfad: Kreiger, Carey
17: 63ad88e5-11c4-444f-beaa-decf8297cc87: Schimmel, Jonathon
```

Output:

C:\MyWork\TE\Clients\Athena Health\2019-03 - Austin PaaS Hackathon\Student and Host Files\0.0 - Hack Setup>node dataread.js

0: 8720e635-0cbf-4dc0-ab72-0b03957ecf6c: Schulist, Pat

1: 95b92321-8798-4636-a682-865f04d6162e: Gleichner, Andres

2: 2e3a045d-0d4b-44e1-8d84-cd352843759f: Schulist, Ova

...

98: d29d9818-837d-4fc2-bbe8-ee51b900a407: Collier, Sydnee

```

99: 891a071c-dde4-46de-af06-733aa8498110: Blanda, Rylan
----- PAGE: 1 -----
0: 845e71df-473f-4ad5-b078-db6e2c9d72c4: Bernier, Vanessa
1: 23188b17-10df-48f3-a4b3-a467865b80c0: Mosciski, Syble
...
97: 02beead4-5838-4a6a-93b5-1c3d80d75c48: Schulist, Wilburn
98: 8d4e14ad-d3c2-4e19-b8a0-90c5a5978d27: Walker, Alda
99: e09a4a61-c141-4fea-8b13-769b122e3a43: Corwin, Hester
----- PAGE: 2 -----
0: 1dfad034-6b5b-4482-8181-978a99093b98: Schroeder, Doyle
1: d9a2d0a0-6808-492e-aa0a-6c8c56c35f8c: O'Kon, Caden
2: 56a11b75-a86b-494c-846d-9e2a57a3a0f4: Barton, Chloe
3: 457e7f1a-38cf-467a-98e0-74d179dc2ffa: Ferry, Claudine
...
99: d2745bc1-aa08-475c-972a-816893eec3e3: Schowalter, Cristian
----- PAGE: 106 -----
0: 23a46216-ff9a-4772-8b43-e5a67422c76a: O'Connell, Vincent
1: 95aecc62-9ab4-4920-8abc-87f50738eede: Smitham, Tania
2: aec7fcfe-51d7-4e48-b1c2-c33d3485cf4a: VonRueden, Adolph
----- PAGE: 107 -----

```

C:\MyWork\TE\Clients\Athena Health\2019-03 - Austin PaaS Hackathon\Student and Host Files\0.0 - Hack Setup>

- We're now finished, make sure there at least **10,000 patient records** in the server before the hack starts.

## Challenge Set 0: Pre-reqs

Challenges: Install tool-set:

- Install the WSL: <https://docs.microsoft.com/en-us/windows/wsl/install-win10>, needed for shell script examples in hack or use cli on browser via shell.azure.com

Install Windows Subsystem for Linux, open PowerShell as Administrator and run:

\$ Enable-WindowsOptionalFeature -Online -FeatureName Microsoft-Windows-Subsystem-Linux

Output:

Windows PowerShell

Copyright (C) Microsoft Corporation. All rights reserved.

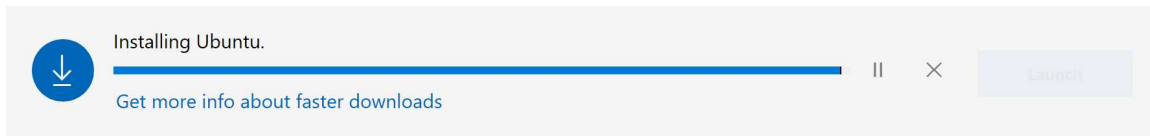
PS C:\WINDOWS\system32> Enable-WindowsOptionalFeature -Online -FeatureName Microsoft-Windows-Subsystem-Linux

Path :

Online : True

RestartNeeded : False

- Install your Linux Distro, download and install ubuntu from the Windows Store



## Ubuntu

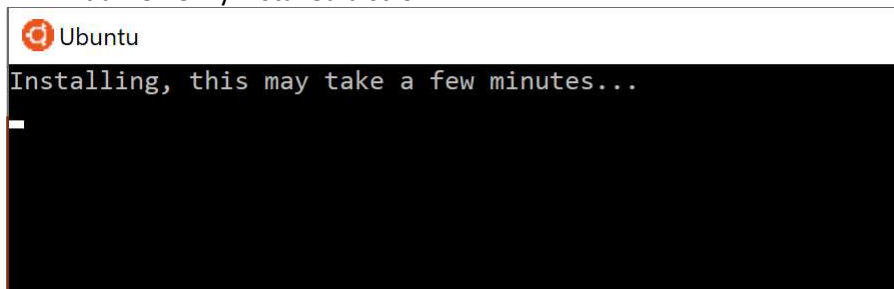
Canonical Group Limited • Developer tools > Utilities

★★★★★ 200  Share

Ubuntu on Windows allows one to use Ubuntu Terminal and run Ubuntu command line utilities including bash, ssh, git, apt and many more.

[More](#)

- Initialize newly installed distro



- Setup a new Linux user account for use with sudo > create a new user and password

Output:

*Installing, this may take a few minutes...*

*Please create a default UNIX user account. The username does not need to match your Windows username.*

*For more information visit: <https://aka.ms/wslusers>*

*Enter new UNIX username: rilian*

*Enter new UNIX password: richard1*

*passwd: password updated successfully*

*Installation successful!*

*To run a command as administrator (user "root"), use "sudo <command>".*

*See "man sudo\_root" for details.*

*rilian@RLIANGSB:~\$*

- Install the Azure CLI in the WSL: <https://docs.microsoft.com/en-us/cli/azure/install-azurecli?view=azure-cli-latest>
- Install VS Code: <https://code.visualstudio.com/>
- Optionally install Azure Storage Explorer: <http://storageexplorer.com>

## Challenge Set1: Load demographic data into PaaS Data Store

- 1.1: Deploy CosmosDB instance supporting SQL interface through the portal.
- 1.2: Deploy an Azure Function that reads from FHIR server and writes to the SQL interface of CosmosDB.
  - Provide sample code for reading from FHIR

- This is the entirety of dataread.js + config.json
- Trigger your function manually for now
- Hint: @azure/cosmos NPM library

<https://docs.microsoft.com/en-us/azure/cosmos-db/sql-api-nodejs-get-started>

Setup your Node.js app:

Before you start writing code to build the application, you can build the framework for your app. Run the following steps to set up your Node.js application that has the framework code:

Open your favorite terminal.

Locate the folder or directory where you'd like to save your Node.js application.

Create two empty JavaScript files with the following commands:

Windows:

```
fsutil file createnew app.js 0
```

```
fsutil file createnew config.js 0
```

Output:

```
C:\MyWork\TE\Clients\Athena Health\2019-03 - Austin PaaS Hackathon\Student and Host Files\1.2 - FHIR to CosmosDB
\Student>fsutil file createnew app.js 0
File C:\MyWork\TE\Clients\Athena Health\2019-03 - Austin PaaS Hackathon\Student and Host Files\1.2 - FHIR to CosmosDB
\Student\app.js is created
```

```
C:\MyWork\TE\Clients\Athena Health\2019-03 - Austin PaaS Hackathon\Student and Host Files\1.2 - FHIR to CosmosDB
\Student>fsutil file createnew config.js 0
File C:\MyWork\TE\Clients\Athena Health\2019-03 - Austin PaaS Hackathon\Student and Host Files\1.2 - FHIR to CosmosDB
\Student\config.js is created
```

Create and initialize a package.json file if not already exist. Use the following command:  
npm init -y

Install the @azure/cosmos module via npm. Use the following command:

```
npm install @azure/cosmos --save
```

Output:

```
C:\MyWork\TE\Clients\Athena Health\2019-03 - Austin PaaS Hackathon\Student and Host Files\1.2 - FHIR to CosmosDB
\Student>npm install @azure/cosmos --save
npm notice created a lockfile as package-lock.json. You should commit this file.
npm WARN fhirDataMgmt@1.0.0 No description
npm WARN fhirDataMgmt@1.0.0 No repository field.
```

```
+ @azure/cosmos@2.1.5
added 25 packages from 17 contributors and audited 53 packages in 11.385s
found 0 vulnerabilities
```

Set your app's configurations  
update config.js:

```
// ADD THIS PART TO YOUR CODE  
var config = {}
```

```
config.endpoint = "~your Azure Cosmos DB endpoint uri here~";  
config.primaryKey = "~your primary key here~";
```

### Create a database:

#### Output:

```
C:\MyWork\TE\Clients\Athena Health\2019-03 - Austin PaaS Hackathon\Student and Host Files\1.2 - FHIR to CosmosDB  
\Student>node app.js  
Created database:  
FamilyDatabase
```

```
Reading database:  
FamilyDatabase
```

```
Completed successfully  
Press any key to exit
```

### Create a container:

#### Output:

```
C:\MyWork\TE\Clients\Athena Health\2019-03 - Austin PaaS Hackathon\Student and Host Files\1.2 - FHIR to CosmosDB  
\Student>node app.js  
Created database:  
FamilyDatabase
```

```
Reading database:  
FamilyDatabase
```

```
Created container:  
FamilyContainer
```

```
Reading container:  
FamilyContainer
```

```
Completed successfully  
Press any key to exit
```

### Create an item:

#### Output:

```
C:\MyWork\TE\Clients\Athena Health\2019-03 - Austin PaaS Hackathon\Student and Host Files\1.2 - FHIR to CosmosDB  
\Student>node app.js  
Created database:  
FamilyDatabase
```

```
Created database:  
FamilyDatabase
```

```
Reading database:  
FamilyDatabase
```

```
Reading database:  
FamilyDatabase
```

```
Created container:  
FamilyContainer
```

```
Created container:  
FamilyContainer
```

*Reading container:  
FamilyContainer*

*Reading container:  
FamilyContainer*

*Completed successfully  
Press any key to exit  
Created family item with id:  
Anderson.1*

*Created family item with id:  
Wakefield.7*

*Completed successfully  
Press any key to exit*

*Query Azure Comos DB resources:*

*Output:*

*C:\MyWork\TE\Clients\Athena Health\2019-03 - Austin PaaS Hackathon\Student and Host  
Files\1.2 - FHIR to CosmosDB\Student>node app.js*

*Created database:  
FamilyDatabase*

*Created database:  
FamilyDatabase*

*Reading database:  
FamilyDatabase*

*Reading database:  
FamilyDatabase*

*Created container:  
FamilyContainer*

*Created container:  
FamilyContainer*

*Reading container:  
FamilyContainer*

*Completed successfully  
Press any key to exit  
Reading container:  
FamilyContainer*

*Item with family id Anderson.1 already exists*

*Item with family id Wakefield.7 already exists*

Querying container:

FamilyContainer

Query returned [{"firstName":"Henriette  
Thaulow","gender":"female","grade":5,"pets":[{"givenName":"Fluffy"}]}

Completed successfully

Press any key to exit

Replace an item:

Output:

C:\MyWork\TE\Clients\Athena Health\2019-03 - Austin PaaS Hackathon\Student and Host Files\1.2 - FHIR to CosmosDB  
\Student>node app.js

Created database:

FamilyDatabase

Reading database:

FamilyDatabase

Created container:

FamilyContainer

Created database:

FamilyDatabase

Reading container:

FamilyContainer

Completed successfully

Press any key to exit

Reading database:

FamilyDatabase

Created container:

FamilyContainer

Reading container:

FamilyContainer

Item with family id Anderson.1 already exists

Item with family id Wakefield.7 already exists

Querying container:

FamilyContainer

Query returned [{"firstName":"Henriette Thaulow","gender":"female","grade":5,"pets":[{"givenName":"Fluffy"}]}

Replacing item:

Anderson.1

Querying container:

FamilyContainer

Query returned [{"firstName":"Henriette Thaulow","gender":"female","grade":6,"pets":[{"givenName":"Fluffy"}]}

Completed successfully

Press any key to exit

Delete an item:

## Output:

C:\MyWork\TE\Clients\Athena Health\2019-03 - Austin PaaS Hackathon\Student and Host Files\1.2 - FHIR to CosmosDB

\Student>node app.js

Created database:

FamilyDatabase

Created database:

FamilyDatabase

Reading database:

FamilyDatabase

Reading database:

FamilyDatabase

Created container:

FamilyContainer

Created container:

FamilyContainer

Reading container:

FamilyContainer

Reading container:

FamilyContainer

Item with family id Anderson.1 already exists

Item with family id Anderson.1 already exists

Item with family id Wakefield.7 already exists

Querying container:

FamilyContainer

Item with family id Wakefield.7 already exists

Querying container:

FamilyContainer

Query returned [{"firstName":"Henriette Thaulow","gender":"female","grade":6,"pets":[{"givenName":"Fluffy"}]}]

Replacing item:

Anderson.1

Query returned [{"firstName":"Henriette Thaulow","gender":"female","grade":6,"pets":[{"givenName":"Fluffy"}]}]

Replacing item:

Anderson.1

Querying container:

FamilyContainer

Querying container:

FamilyContainer

Query returned [{"firstName":"Henriette Thaulow","gender":"female","grade":6,"pets":[{"givenName":"Fluffy"}]}]

Completed successfully

Press any key to exit

Query returned [{"firstName":"Henriette Thaulow","gender":"female","grade":6,"pets":[{"givenName":"Fluffy"}]}]

Deleted item:

Anderson.1

Completed successfully

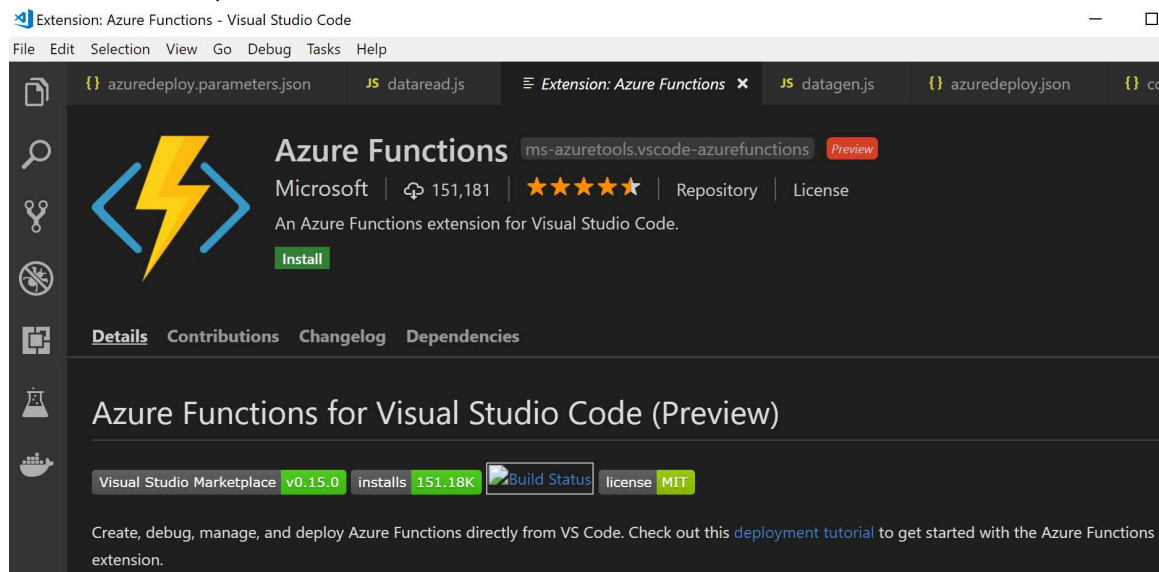


Press any key to exit

Install Azure Functions extension for VS Code:

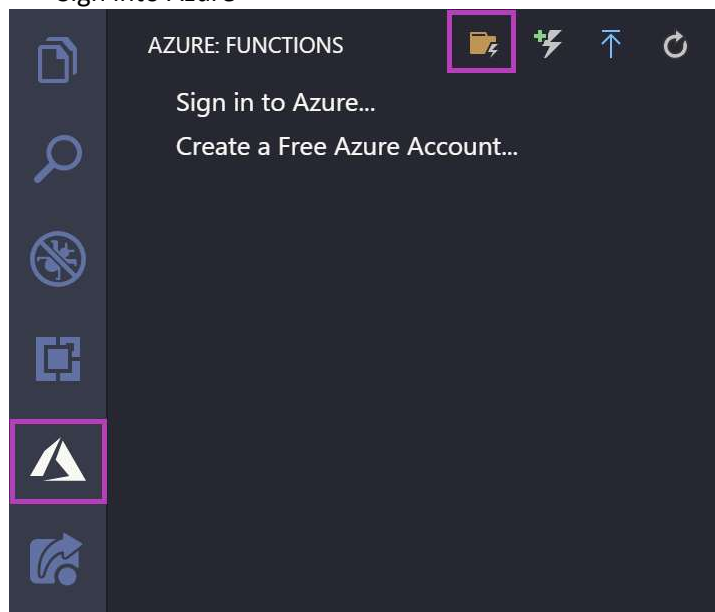
<https://code.visualstudio.com/tutorials/functions-extension/getting-started>

vscode:extension/ms-azuretools.vscode-azurefunctions




After installing Azure Function extension for VS Code:



- Click Azure logo > Azure Functions explorer Create new project
- Sign into Azure




Click  to create a new Azure Function project


Login into Azure if you have not done so...


 To sign in, use a web browser to open the page  
<https://microsoft.com/devicelogin> and enter the code  
A9LLJNUUG to authenticate.

Source: Azure Account (Extension)  



- Browse to <https://microsoft.com/devicelogin>
- Enter code (see above) at the Device Login window (below)

<https://login.microsoftonline.com/common/oauth2/deviceauth> 





## Device Login

Enter the code that you received from the application on your device

## Visual Studio Code

Click Cancel if this isn't the application you were trying to sign in to on your device.

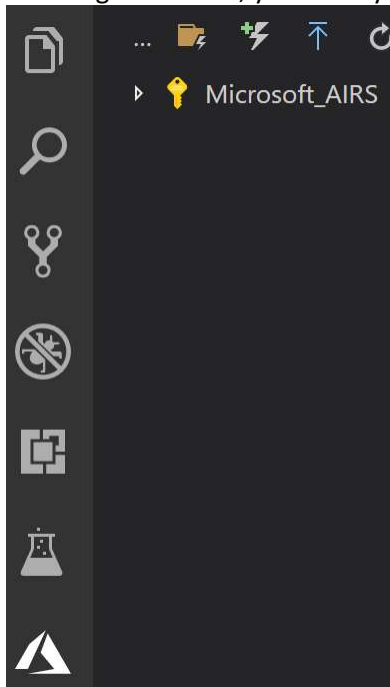
 

- At login window, change to another account and type in your work email address and password if prompted.
- If successful, a confirmation window will display (below)

## Visual Studio Code

You have signed in to the Visual Studio Code application on your device. You may now close this window.

After log into Azure, you'll see your subscription id:



- Choose a project folder for your app and choose JavaScript
- When prompted, choose "Open in current window"
- Add a "HTTP trigger Function" to your app

Install VS Code Azure Function core tools via PowerShell:

```
$ npm i -g azure-functions-core-tools@2
```

Output:

```
PS C:\WINDOWS\system32> npm i -g azure-functions-core-tools@2
C:\Program Files\nodejs\azfun -> C:\Program Files\nodejs\node_modules\azure-functions-core-tools\lib\main.js
C:\Program Files\nodejs\func -> C:\Program Files\nodejs\node_modules\azure-functions-core-tools\lib\main.js
```




**new** Create a new function from a template. Aliases: *new, create*  
 --language [-l] Template programming language, such as C#, F#, JavaScript, etc.  
 --template [-t] Template name  
 --name [-n] Function name  
 --csx use old style csx dotnet functions

**init** Create a new Function App in the current folder. Initializes git repo.  
 --source-control Run git init. Default is false.  
 --worker-runtime Runtime framework for the functions. Options are: *dotnet, node, python*  
 --force Force initializing  
 --docker Create a Dockerfile based on the selected worker runtime  
 --csx use csx dotnet functions  
 --language Initialize a language specific project. Currently supported when --worker-runtime set to node. Options are - "typescript" and "javascript"

**logs** Gets logs of Functions running on custom backends  
 --platform Hosting platform for the function app. Valid options: *kubernetes*  
 --name Function name  
 --config [Optional] Config file

Create a Function to your app:



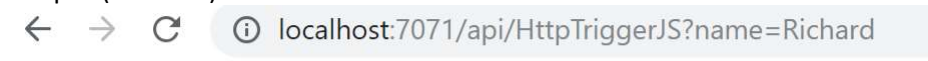
- Click  in VS Code Azure Function explorer
- Choose HTTP Trigger > name = HTTPTriggerJS
- Authentication = Anonymous

<https://code.visualstudio.com/tutorials/functions-extension/create-function>

Run and test app locally:

- To run the project locally, press F5
- Browse to <http://localhost:7071/api/HttpTriggerJS?name=Richard>

Output (browser):

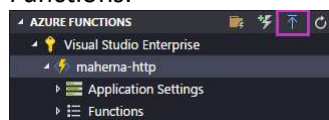


Hello Richard

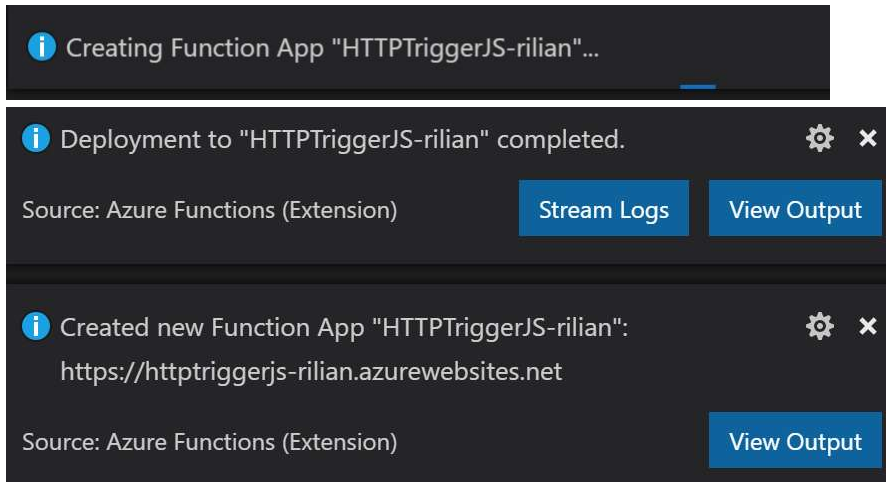
Deploy sample app to Azure Function:

<https://code.visualstudio.com/tutorials/functions-extension/deploy-app>

In the AZURE FUNCTIONS explorer, click the blue up arrow icon to deploy your app to Azure Functions.



- Input global unique name for function
- Select location
- Choose create new storage account and hit "Enter"



VS Code > Output: Azure Function

Output:

Creating storage account "httptriggerjsrilian" in location "North Central US" with sku "Standard\_LRS" ...  
Successfully created storage account "httptriggerjsrilian".  
Creating Function App "HTTPTriggerJS-rilian" ...  
Created new Function App "HTTPTriggerJS-rilian": <https://httptriggerjs-rilian.azurewebsites.net>

8:53:35 PM HTTPTriggerJS-rilian: Creating zip package...  
8:53:36 PM HTTPTriggerJS-rilian: Starting deployment...  
8:53:40 PM HTTPTriggerJS-rilian: Updating submodules.  
8:53:40 PM HTTPTriggerJS-rilian: Preparing deployment for commit id '5ad98d6d19'.  
8:53:41 PM HTTPTriggerJS-rilian: Skipping build. Project type: Run-From-Zip  
8:53:41 PM HTTPTriggerJS-rilian: Skipping post build. Project type: Run-From-Zip  
8:53:44 PM HTTPTriggerJS-rilian: Syncing 2 function triggers with payload size 160 bytes successful.  
8:53:44 PM HTTPTriggerJS-rilian: Updating D:\home\data\SitePackages\packagename.txt with deployment 20190324005337.zip  
8:53:44 PM HTTPTriggerJS-rilian: Deployment successful.  
Deployment to "HTTPTriggerJS-rilian" completed.  
HTTP Trigger Urls:  
HttpTriggerJS: <https://httptriggerjs-rilian.azurewebsites.net/api/HttpTriggerJS>

EventHub Trigger function:

Restoring packages for C:\MyWork\TE\Clients\Athena Health\Hackathon\Student and Host Files\1.2 - FHIR to CosmosDB\Lab\extensions.csproj...

Restore completed in 715.03 ms for C:\MyWork\TE\Clients\Athena Health\Hackathon\Student and Host Files\1.2 - FHIR to CosmosDB\Lab\extensions.csproj.

extensions -> C:\MyWork\TE\Clients\Athena Health\Hackathon\Student and Host Files\1.2 - FHIR to CosmosDB\Lab\bin\extensions.dll

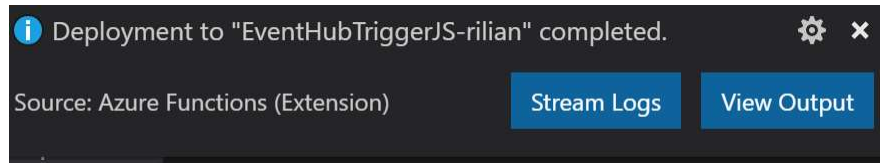
Build succeeded.

0 Warning(s)

0 Error(s)

Time Elapsed 00:00:05.02

Terminal will be reused by tasks, press any key to close it.



#### Output:

Creating storage account "httptriggerjsrilian" in location "North Central US" with sku "Standard\_LRS"...

Successfully created storage account "httptriggerjsrilian".

Creating Function App "HTTPTriggerJS-rilian"...

Created new Function App "HTTPTriggerJS-rilian": <https://httptriggerjs-rilian.azurewebsites.net>

8:53:35 PM HTTPTriggerJS-rilian: Creating zip package...

8:53:36 PM HTTPTriggerJS-rilian: Starting deployment...

8:53:40 PM HTTPTriggerJS-rilian: Updating submodules.

8:53:40 PM HTTPTriggerJS-rilian: Preparing deployment for commit id '5ad98d6d19'.

8:53:41 PM HTTPTriggerJS-rilian: Skipping build. Project type: Run-From-Zip

8:53:41 PM HTTPTriggerJS-rilian: Skipping post build. Project type: Run-From-Zip

8:53:44 PM HTTPTriggerJS-rilian: Syncing 2 function triggers with payload size 160 bytes successful.

8:53:44 PM HTTPTriggerJS-rilian: Updating D:\home\data\SitePackages\packagename.txt with deployment 20190324005337.zip

8:53:44 PM HTTPTriggerJS-rilian: Deployment successful.

Deployment to "HTTPTriggerJS-rilian" completed.

HTTP Trigger Urls:

HttpTriggerJS: <https://httptriggerjs-rilian.azurewebsites.net/api/HttpTriggerJS>

Creating storage account "eventhubtriggerjsrilian" in location "northcentralus" with sku "Standard\_LRS"...

Successfully created storage account "eventhubtriggerjsrilian".

Creating Function App "EventHubTriggerJS-rilian"...

Created new Function App "EventHubTriggerJS-rilian": <https://eventhubtriggerjs-rilian.azurewebsites.net>

9:14:06 PM EventHubTriggerJS-rilian: Creating zip package...

9:14:09 PM EventHubTriggerJS-rilian: Starting deployment...

9:14:13 PM EventHubTriggerJS-rilian: Updating submodules.

9:14:14 PM EventHubTriggerJS-rilian: Preparing deployment for commit id '69e9ded511'.

9:14:15 PM EventHubTriggerJS-rilian: Skipping build. Project type: Run-From-Zip

9:14:15 PM EventHubTriggerJS-rilian: Skipping post build. Project type: Run-From-Zip

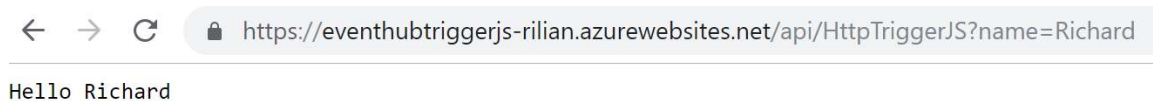
9:14:18 PM EventHubTriggerJS-rilian: Syncing 3 function triggers with payload size 265 bytes successful.

9:14:19 PM EventHubTriggerJS-rilian: Updating D:\home\data\SitePackages\packagename.txt with deployment 20190324011411.zip

9:14:19 PM EventHubTriggerJS-rilian: Deployment successful.

Deployment to "EventHubTriggerJS-rilian" completed.

Browse to: <https://eventhubtriggerjs-rilian.azurewebsites.net/api/HttpTriggerJS?name=Richard>



- Deploy dataread.js nodejs to Azure Function
- 
- Deploy CosmosDB instance supporting Cassandra or SQL interface through the portal

- Deploy Azure Function that reads from FHIR server and writes to the Cassandra interface of CosmosDB

- Sample code for reading from FHIR > dataread.js + config.json

Sample code to develop node.js app with Cassandra API using Cosmos DB

<https://github.com/Azure-Samples/azure-cosmos-db-cassandra-nodejs-getting-started>

- Need active Azure Cassandra API account  
<https://docs.microsoft.com/en-us/azure/cosmos-db/create-cassandra-nodejs>
- Install Node.js version is v0.10.29 or higher
  - Check node version, run in PowerShell: \$ npm install check-node-version
  - Install nodejs and tools: <https://nodejs.org/en/>

Output (after installed):

```
=====
Tools for Node.js Native Modules Installation Script
=====
```

*This script will install Python and the Visual Studio Build Tools, necessary to compile Node.js native modules. Note that Chocolatey and required Windows updates will also be installed.*

*This will require about 3 Gb of free disk space, plus any space necessary to install Windows updates. This will take a while to run.*

*Please close all open programs for the duration of the installation. If the installation fails, please ensure Windows is fully updated, reboot your computer and try to run this again. This script can be found in the Start menu under Node.js.*

*You can close this window to stop now. Detailed instructions to install these tools manually are available at <https://github.com/nodejs/node-gyp#on-windows>*

*Press any key to continue . . .*

*Using this script downloads third party software*

*This script will direct to Chocolatey to install packages. By using Chocolatey to install a package, you are accepting the license for the application, executable(s), or other artifacts delivered to your machine as a result of a Chocolatey install. This acceptance occurs whether you know the license terms or not. Read and understand the license terms of the packages being installed and their dependencies prior to installation:*

*- <https://chocolatey.org/packages/chocolatey>*

*- <https://chocolatey.org/packages/python2>*

*- <https://chocolatey.org/packages/visualstudio2017-workload-vctools>*

*This script is provided AS-IS without any warranties of any kind*

*Chocolatey has implemented security safeguards in their process to help protect the community from malicious or pirated software, but any use of this script is at your own risk. Please read the Chocolatey's legal terms of use as well as how the community repository for Chocolatey.org is maintained.*

*Press any key to continue . . .*

In PowerShell Output:

Getting latest version of the Chocolatey package for download.

Getting Chocolatey from <https://chocolatey.org/api/v2/package/chocolatey/0.10.13>.

Downloading 7-Zip commandline tool prior to extraction.

Extracting C:\Users\rilian\AppData\Local\Temp\chocolatey\chocInstall\chocolatey.zip to C:\Users\rilian\AppData\Local\Temp\chocolatey\chocInstall...

Installing chocolatey on this machine



Chocolatey v0.10.13  
2 validations performed. 1 success(es), 1 warning(s), and 0 error(s).

Upgrading the following packages:  
python2;visualstudio2017-workload-vctools  
By upgrading you accept licenses for the packages.  
python2 is not installed. Installing...  
Progress: Downloading python2 2.7.16... 100%  
Progress: Downloading python2 2.7.16... 100%

python2 v2.7.16 [Approved]  
python2 package files upgrade completed. Performing other installation steps.  
Installing 64-bit python2...

- Install Git: <http://git-scm.com/>
- Install node.js driver for apache cassandra: run \$ npm install cassandra-driver
- Clone git repo: run  
\$ git clone git@github.com:Azure-Samples/Azure-Samples/azure-cosmos-db-cassandra-node-getting-started.git cosmosdb
  - cd cosmosdb
- Update config.js with ComosDB connectstring values:
  - config.username
  - config.password
  - config.contactPoint
- Run npm install to install required modules
- Run node upprofile.js to start your node app
- Trigger your function manually
- Build a data model for Cassandra for the fields to search for. Hint: need Keyspace and Table

## Challenge Set 2: Event Driven Demographics

2.1: Deploy Eventhub

2.2: Update Azure Function to read from FHIR server and drop to Eventhub

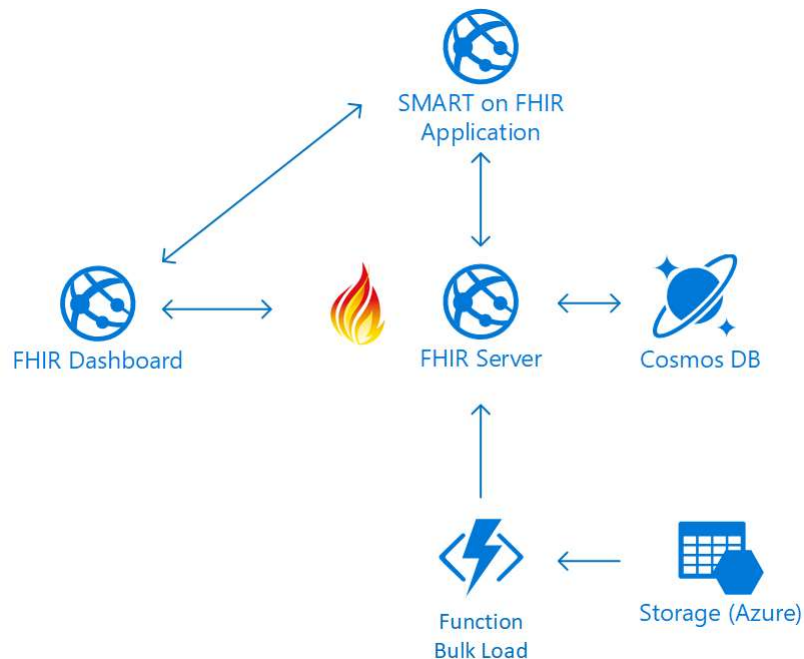
2.3: Deploy new Azure Function triggered by Event Hub pushing data to CosmosDB

2.1: Deploy Event Hub

<https://github.com/Azure/azure-quickstart-templates/tree/master/201-event-hubs-create-event-hub-and-consumer-group/>

Create a EventHubs namespace with EventHub and ConsumerGroup using an ARM template:

<http://azure.microsoft.com/documentation/articles/service-bus-resource-manager-namespace->



[event-hub/](#)



<https://portal.azure.com/#create/Microsoft.Template/uri/https%3A%2F%2Fraw.githubusercontent.com%2FAzure%2Fazure-quickstart-templates%2Fmaster%2F201-event-hubs-create-event-hub-and-consumer-group%2Fazuredeploy.json>

22: Update Azure Function to read from FHIR server and drop to Eventhub

FHIR server sample code:

<https://github.com/Microsoft/fhir-server-samples>

Update Powershell "Az" package:

```
PS C:\WINDOWS\system32> Install-Module -Name Az -AllowClobber
```

Untrusted repository

You are installing the modules from an untrusted repository. If you trust this repository, change its InstallationPolicy value by running the Set-PSRepository cmdlet. Are you sure you want to install the modules from 'PSGallery'?

[Y] Yes [A] Yes to All [N] No [L] No to All [S] Suspend [?] Help (default is "N"): A

Update PowerShell modules:

\$ Update-Module <module name>

Output:

```
PS C:\projects\fhir-server-samples\deploy\scripts> Update-Module
```

Untrusted repository

You are installing the modules from an untrusted repository. If you trust this repository, change its InstallationPolicy value by running the Set-PSRepository cmdlet. Are you sure you want to install the modules from

'PSGallery'?  
[Y] Yes [A] Yes to All [N] No [L] No to All [S] Suspend [?] Help (default is "N"): A

## Output:

PS C:\projects\fhir-server-samples\deploy\scripts> az login

Note, we have launched a browser for you to login. For old experience with device code, use "az login --use-device-code"

-----  
Exception happened during processing of request from ('127.0.0.1', 20579)

Traceback (most recent call last):

File "socketserver.py", line 317, in \_handle\_request\_noblock

File "socketserver.py", line 348, in process\_request

File "socketserver.py", line 361, in finish\_request

File "socketserver.py", line 696, in \_\_init\_\_

File "http\server.py", line 418, in handle

File "http\server.py", line 386, in handle\_one\_request

File "socket.py", line 586, in readinto

ConnectionResetError: [WinError 10054] An existing connection was forcibly closed by the remote host

-----  
You have logged in. Now let us find all the subscriptions to which you have access...

```
[
  {
    "cloudName": "AzureCloud",
    "id": "327ead23-9a0f-4d49-a37f-9c9dda2818d4",
    "isDefault": true,
    "name": "Microsoft_AIRS",
    "state": "Enabled",
    "tenantId": "72f988bf-86f1-41af-91ab-2d7cd011db47",
    "user": {
      "name": "rilian@microsoft.com",
      "type": "user"
    }
  }
]
```

PS C:\projects\fhir-server-samples\deploy\scripts> Login-AzureRmAccount -TenantId 72f988bf-86f1-41af-91ab-2d7cd011db47

Account : rilian@microsoft.com

SubscriptionName : Microsoft\_AIRS

SubscriptionId : 327ead23-9a0f-4d49-a37f-9c9dda2818d4

TenantId : 72f988bf-86f1-41af-91ab-2d7cd011db47

Environment : AzureCloud

PS C:\projects\fhir-server-samples\deploy\scripts> .\Create-FhirServerSamplesEnvironment.ps1 -EnvironmentName dev-athena-hack -UsePaaS \$true

## Setup event-hubs for nodjes:

Install @azure/event-hubs npm module:

\$ npm install @azure/event-hubs

## Output:

C:\MyWork\TE\Clients\Athena Health\2019-03 - Austin PaaS Hackathon\Student and Host Files\1.2 - FHIR to CosmosDB\Student>

npm install @azure/event-hubs

npm WARN fhirdata@1.0.0 No description

npm WARN fhirdata@1.0.0 No repository field.

+ @azure/event-hubs@1.0.8

added 15 packages from 24 contributors and audited 642 packages in 5.392s

found 8 moderate severity vulnerabilities

run `npm audit fix` to fix them, or `npm audit` for details

## Client creation

The simplest usage is to use the static factory method

`EventHubClient.createFromConnectionString(_connection-string_, _event-hub-path_)`. Once you have a client, you can use it for:

## Sending events

You can send a single event using `client.send()` method.

You can even batch multiple events together using `client.sendBatch()` method.

## Receiving events

There are two ways to receive events using the EventHub Client

Send an event with partition key:

```
const { EventHubClient, EventPosition } = require('@azure/event-hubs');
```

```
const client =
```

```
EventHubClient.createFromConnectionString(process.env["EVENTHUB_CONNECTION_STRING"], process.env["EVENTHUB_NAME"]);
```

```
async function main() {
```

```
  // NOTE: For receiving events from Azure Stream Analytics, please send Events to an EventHub where the body is a JSON object.
```

```
  // const eventData = { body: { "message": "Hello World" }, partitionKey: "pk12345"};
```

```
  const eventData = { body: "Hello World", partitionKey: "pk12345"};
```

```
  const delivery = await client.send(eventData);
```

```
  console.log("message sent successfully.");
```

```
}
```

```
main().catch((err) => {
```

```
  console.log(err);
```

```
});
```

Send multiple events as a batch

```
const { EventHubClient, EventPosition } = require('@azure/event-hubs');
```

```
const client =
```

```
EventHubClient.createFromConnectionString(process.env["EVENTHUB_CONNECTION_STRING"], process.env["EVENTHUB_NAME"]);
```

```
async function main() {
```

```
  const datas = [
```

```
    { body: "Hello World 1", applicationProperties: { id: "Some id" }, partitionKey: "pk786" },
```

```
    { body: "Hello World 2" },
```

```
    { body: "Hello World 3" }
```

```
  ];
```

```
  // NOTE: For receiving events from Azure Stream Analytics, please send Events to an EventHub
```

```

// where the body is a JSON object/array.
// const datas = [
//   { body: { "message": "Hello World 1" }, applicationProperties: { id: "Some id" },
partitionKey: "pk786" },
//   { body: { "message": "Hello World 2" } },
//   { body: { "message": "Hello World 3" } }
// ];
const delivery = await client.sendBatch(datas);
console.log("message sent successfully.");
}

main().catch((err) => {
  console.log(err);
});

```

2.3: Deploy new Azure Function triggered by Event Hub pushing data to CosmosDB