Cognitive Services, Serverless Compute in Azure & Azure Stack

In 30 minutes we are going to show you some rapid innovation on both Azure and Azure Stack -

- Build out an Severless PaaS web application in Azure
    - https://azure.microsoft.com/en-us/services/functions/

- Deploy some open source code that uses Azure AI to analyse text and return a sentiment score
    - https://azure.microsoft.com/en-us/services/cognitive-services/

- Redeploy this exact some code to Azure Stack, showing you how interoperable the two environments are.
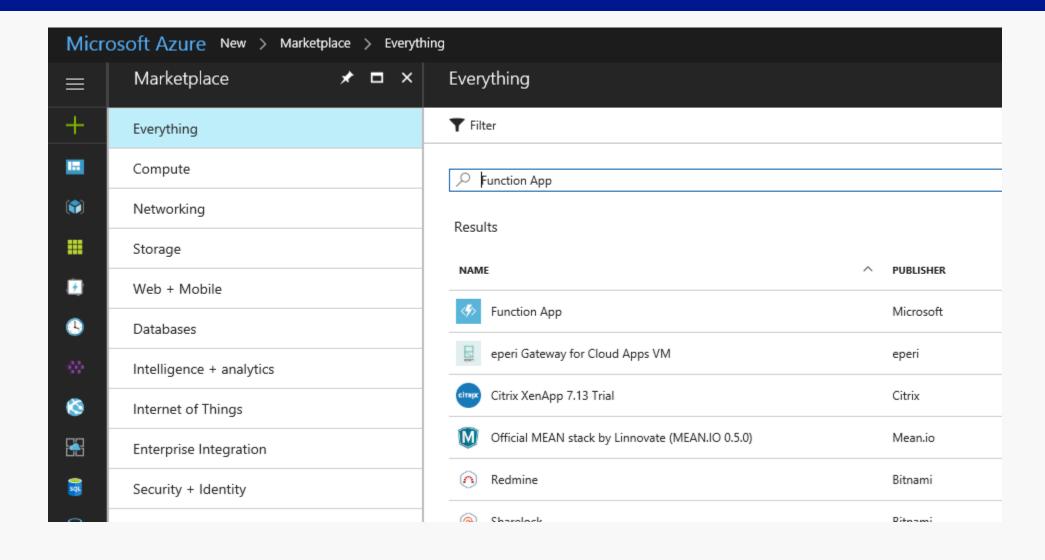
AI, Serverless Compute in Azure & Azure Stack

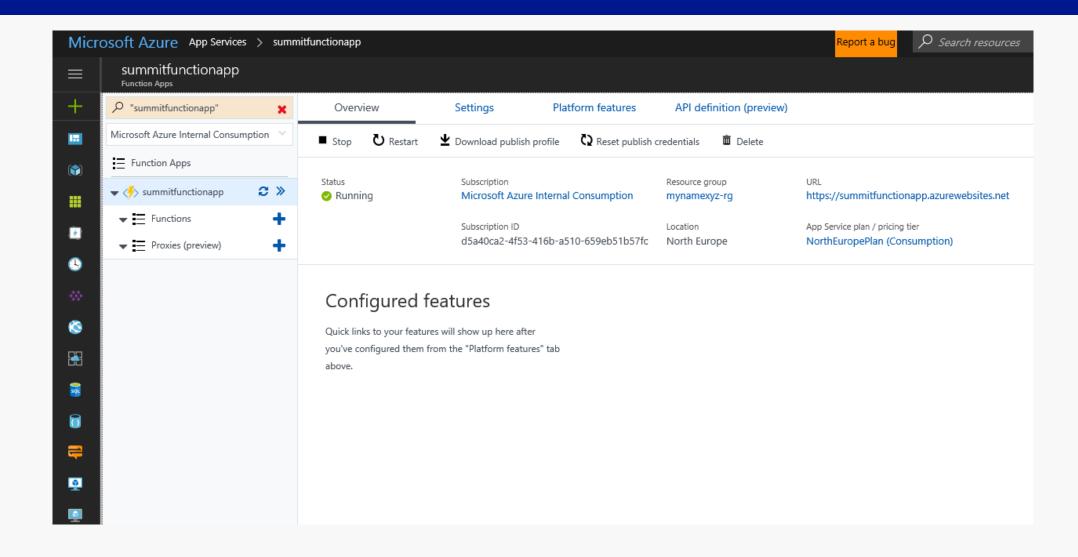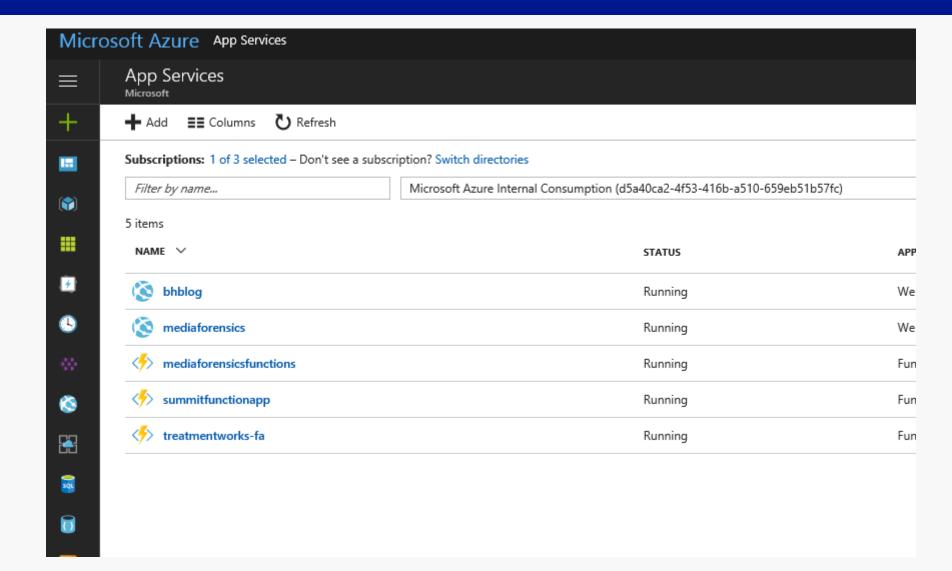- Navigate to https://github.com/ben-houghton/summitcode

Navigate to your Function App in Azure

Set Up your Function App in Azure

Create your Function in Azure

Add the Sentimentfunction.js Code

Microsoft Azure   App Services > summitfunctionapp - HttpTriggerJS1

Report a bug   Search resources

summitfunctionapp - HttpTriggerJS1
Function Apps

"summitfunctionapp"

Microsoft Azure Internal Consumption

Function Apps

summitfunctionapp
  Functions
    HttpTriggerJS1
      Integrate
      Manage
      Monitor
  Proxies (preview)

index.js   Save   ▶ Save and run

```
1
2  var request = require('request');
3
4
5  module.exports = function (context, req) {
6      context.log('Node.js HTTP trigger function processed a request. RequestUri=%s', req.originalUrl);
7
8      // Make a call out to Cognitive Services
9      if (req.query.text)
10     {
11
12         request.post({
13             url: 'https://westus.api.cognitive.microsoft.com/text/analytics/v2.0/sentiment?',
14             headers: {
15                 "Content-Type": "application/json",
16                 "Ocp-Apim-Subscription-Key": "[Cogntive Services Key]"
17             },
18             json: {
19                 "documents": [
20                     {
21                         "language": "en",
22                         "id": "id1",
23                         "text": req.query.text
24                     }
25                 ]
26             }
27         },
28         function (err, res, body) {
29             context.log("Response from Cog API (err, res, body)");
30             context.log(JSON.stringify(err, null, " "));
31             context.log(JSON.stringify(res, null, " "));
32             context.log(JSON.stringify(body, null, " "));
33
34             // Check to see if we succeeded.
35             if (err || res.statusCode != 200) {
36                 context.log(err);
37                 context.res = {
38                     status: 500,
39                     body: err
```

Add the Cognitive Services Subscription Key ☁

https://github.com/ben-houghton/summitcode

```javascript
if (req.query.text) {

    request.post({
        url: 'https://westus.api.cognitive.microsoft.com/text/analytics/v2.0/sentiment?',
        headers: {
            "Content-Type": "application/json",
            "Ocp-Apim-Subscription-Key": "[Cogntive Services Key]"
        },
        json: {
            "documents": [
              {
```

Install the Node.js Request Package

summitfunctionapp
Function Apps

Overview | Settings | **Platform features** | API definition (preview)

🔍 Search features

GENERAL SETTINGS

- ≡ Application settings
- ⦀ Properties
- ☁ Backups
- ◷ All settings

CODE DEPLOYMENT

- ⚙ Deployment options
- ▦ Deployment credentials

DEVELOPMENT TOOLS

- ▸_ Console
- ⚒ Advanced tools (Kudu)
- </> App Service Editor
- ≣ Resource Explorer
- ⊞ Extensions

NETWORKING

- ⟷ Networking
- ⛊ SSL
- ▦ Custom domains
- 🔑 Authentication / Authorization
- ❗ Push notifications

MONITORING

- ✓ Diagnostic logs
- 🔲 Log streaming
- 🖥 Process explorer
- 🔲 Security scanning

API

- ❗ CORS
- ▤ API definition

APP SERVICE PLAN

- ▦ App Service plan
- 📊 Quotas

RESOURCE MANAGEMENT

- ▦ Activity log
- 👥 Access control (IAM)
- 🏷 Tags
- 🔒 Locks
- ▦ Automation script

Search: "summitfunctionapp"
Microsoft Azure Internal Consumption

- ≡ Function Apps
- ⚡ summitfunctionapp
  - ≡ Functions
    - ƒ HttpTriggerJS1
      - ⚡ Integrate
      - ⚙ Manage
      - 🔍 Monitor
  - ≡ Proxies (preview)

Click on Advanced Tools (Kudu)

Install the Node.js Request Package

Kudu   Environment   Debug console ▾   Process explorer   Tools ▾   Site extensions

... / HttpTriggerJS1 ➕ | 3 items   🏠 🌐 🖴

	Name	Modified	Size
⬇➖	📂 node_modules	24/04/2017 11:09:36	
⬇✏➖	📄 function.json	24/04/2017 10:04:06	1 KB
⬇✏➖	📄 index.js	24/04/2017 10:59:31	2 KB

⌄ ⌃

Use old console

```
D:\home>
D:\home\site>
D:\home\site\wwwroot>
D:\home\site\wwwroot\HttpTriggerJS1>npm install request
D:\home\site\wwwroot\HttpTriggerJS1
`-- request@2.81.0
  +-- aws-sign2@0.6.0
  +-- aws4@1.6.0
  +-- caseless@0.12.0
  +-- combined-stream@1.0.5
  | `-- delayed-stream@1.0.0
  +-- extend@3.0.0
  +-- forever-agent@0.6.1
```

Debug Console >CMD

Above the console click –

Site
Wwwroot
[The name of your function]

Type
npm install request

Test your Azure Function

index.js Save ▶ Run </> Get function URL

```
1
2  var request = require('request');
3
4
5  module.exports = function (context, req) {
6      context.log('Node.js HTTP trigger function processed a request. RequestUri=%s', req.originalUrl);
7
8      // Make a call out to Cognitive Services
9      if (req.query.text)
10     {
11
12         request.post({
13             url: 'https://westus.api.cognitive.microsoft.com/text/analytics/v2.0/sentiment?',
14             headers: {
15                 "Content-Type": "application/json",
16                 "Ocp-Apim-Subscription-Key": "c4e04e40d16e4752a62f318f2f43cea2"
17             },
18             json: {
19                 "documents": [
20                     {
21                         "language": "en",
22                         "id": "id1",
23                         "text": req.query.text
24                     }
25                 ]
26             }
27         },
```

Logs   ❚❚ Pause 🗑 Clear 📋 Copy logs ↗ Expand ⌄

View files   Test   Keys   ›

HTTP method

GET ⌄

Query

text   what a lovely day   ✕

+ Add parameter

Headers

There are no headers

+ Add header

Request body

```
1
```

Output   ✓ Status: 200 OK

{"score":0.990714707977006}

▶ Run

Get Postman so we can make Requests

https://www.getpostman.com/

File   Edit   View   Favorites   Tools   Help

Postman | Supercharge your ...

Products   Pricing   Documentation

P O S T M A N

Developing APIs is hard.
Postman makes it easy.

Download the free Postman App:

| MacOS | Windows ⌄ | Linux ⌄ | Chrome |

Get Your Function URL

Click on the Get Function URL link and copy the Function URL
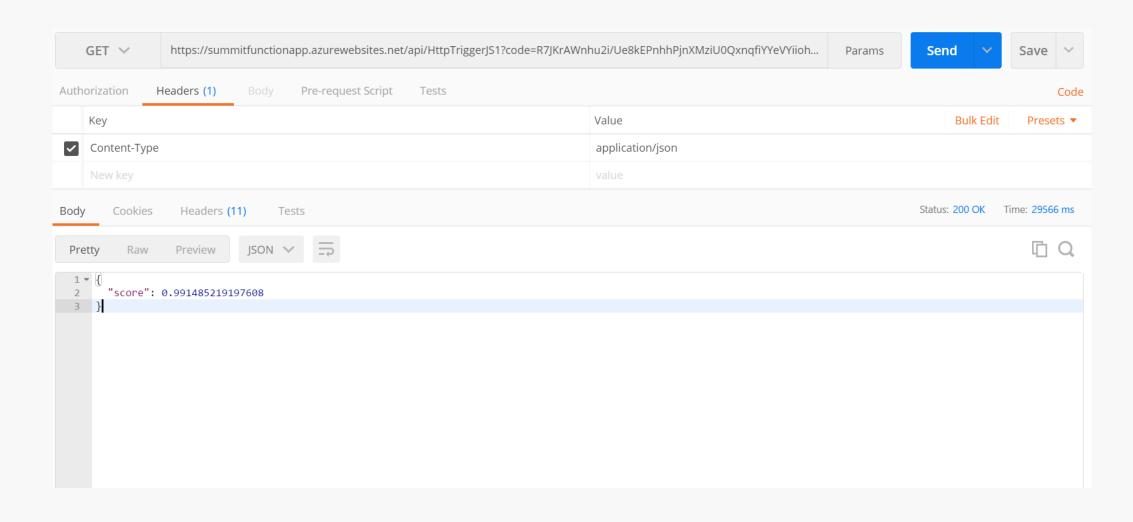
Copy the Function URL link and add any query string or body parameters to Postman
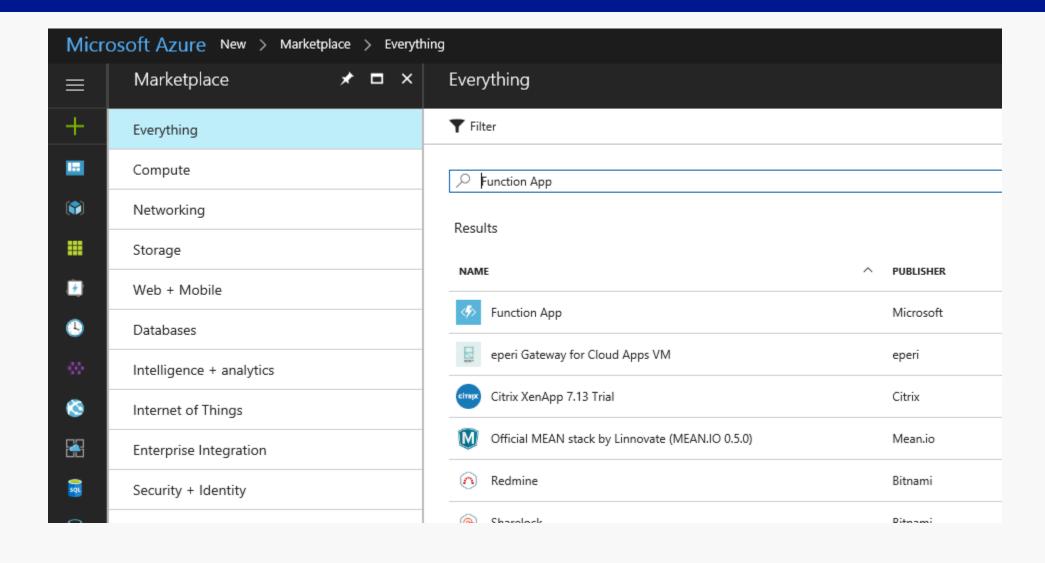
Make Remote Requests to the Azure Function

| GET ⌄ | https://summitfunctionapp.azurewebsites.net/api/HttpTriggerJS1?code=R7JKrAWnhu2i/Ue8kEPnhhPjnXMziU0QxnqfiYYeVYiioh... | Params | **Send** ⌄ | Save ⌄ |

Authorization   **Headers (1)**   Body   Pre-request Script   Tests   Code

	Key	Value	Bulk Edit Presets ⌄
☑	Content-Type	application/json	
	New key	value	

Body   Cookies   Headers (11)   Tests   Status: 200 OK   Time: 29566 ms

Pretty   Raw   Preview   JSON ⌄

```
1  {
2      "score": 0.991485219197608
3  }
```

Select Function App in Azure Stack

Provision your Function App in Azure

Navigate to your Function App in Azure
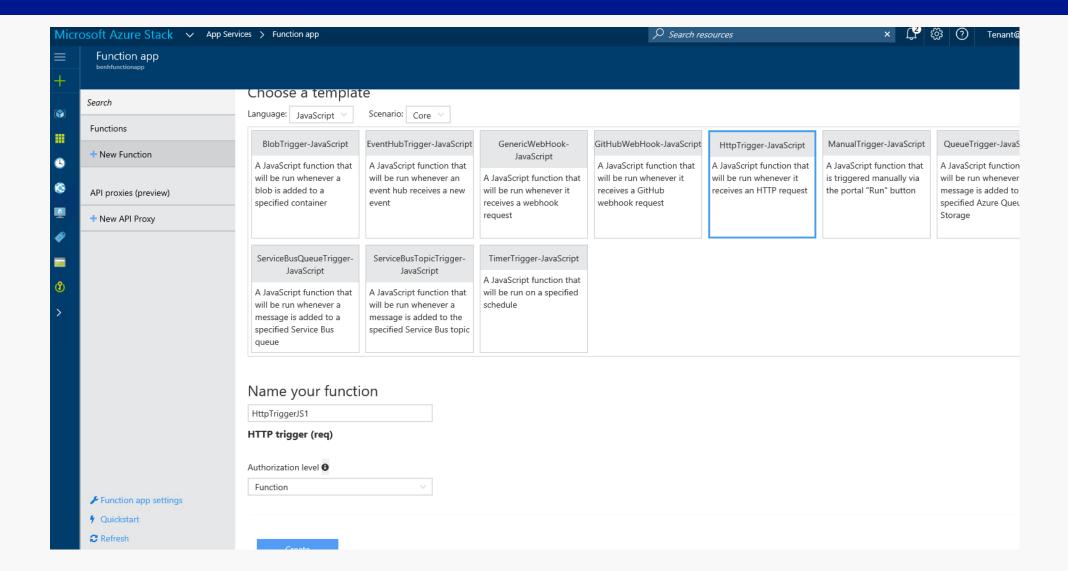
Set Up your Function App in Azure Stack
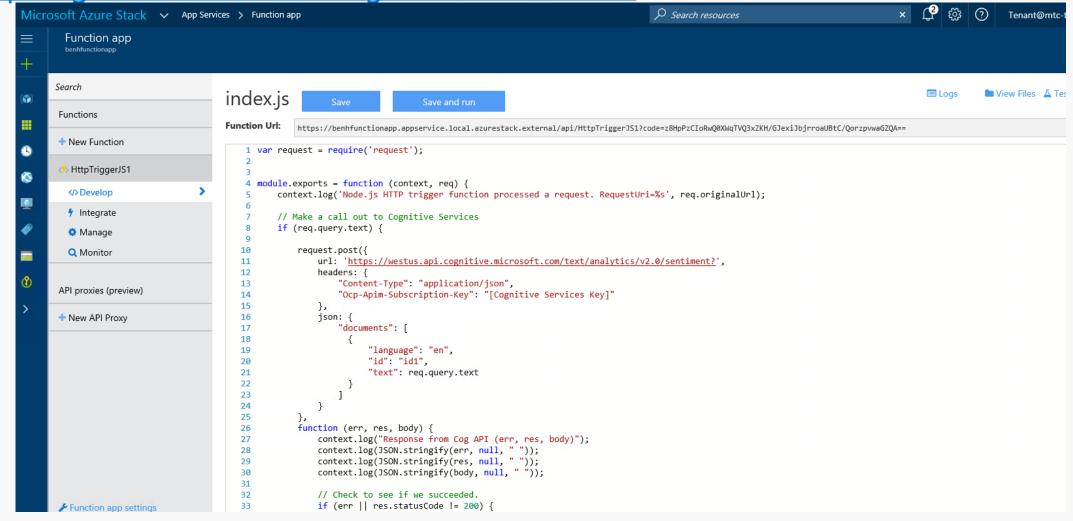
Add the Sentimentfunction.js Code
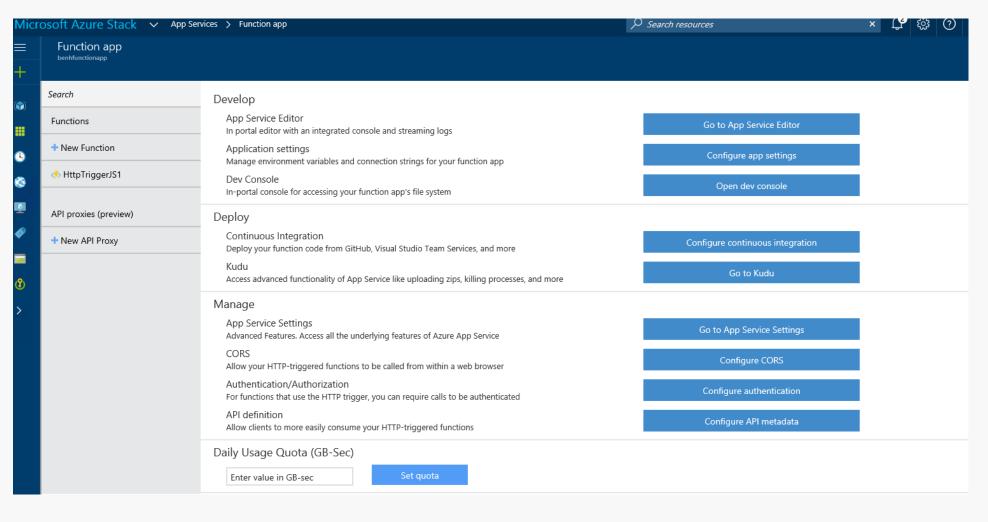
Install the Node.js Request Package on Azure Stack

Click on Advanced Tools (Kudu)
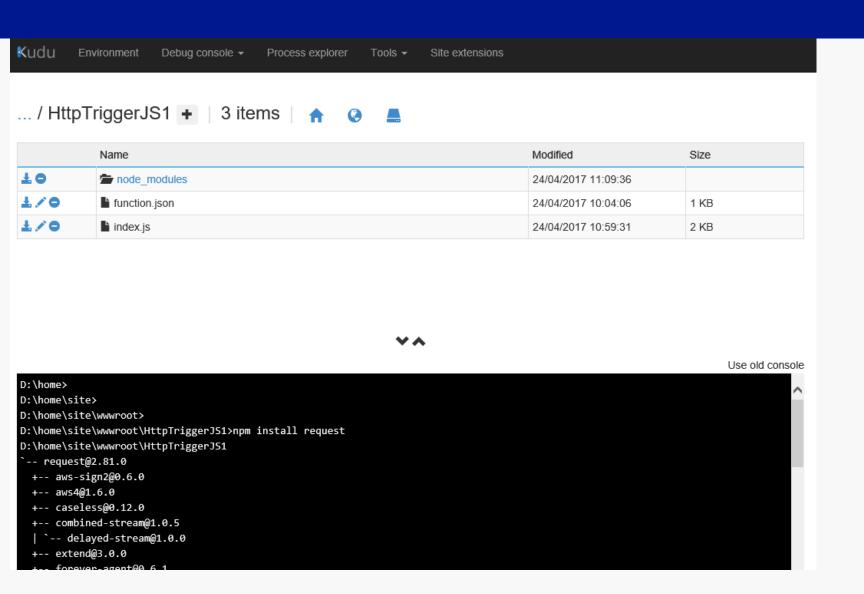
Install the Node.js Request Package on Azure Stack

Debug Console >CMD
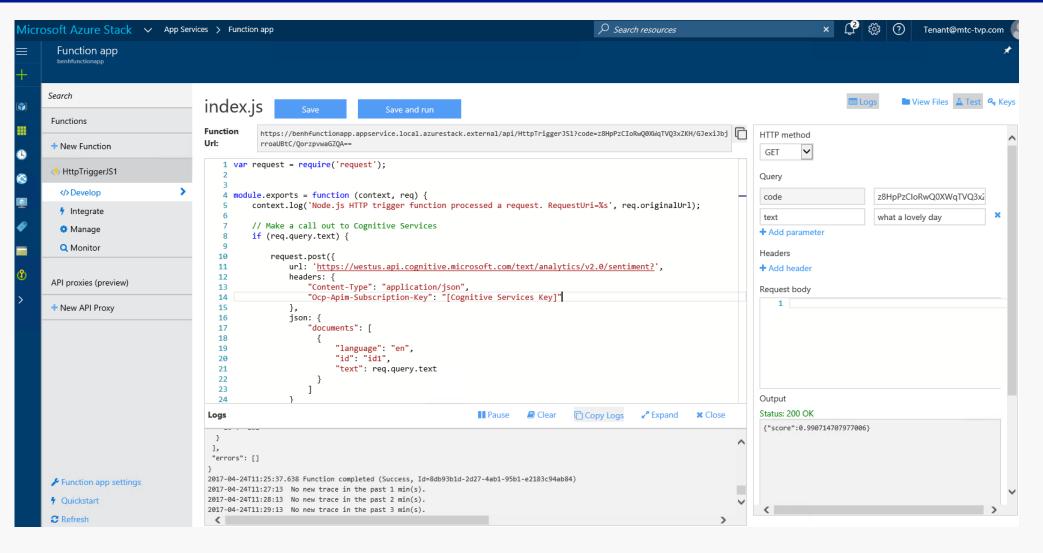
Above the console click –

Site
Wwwroot
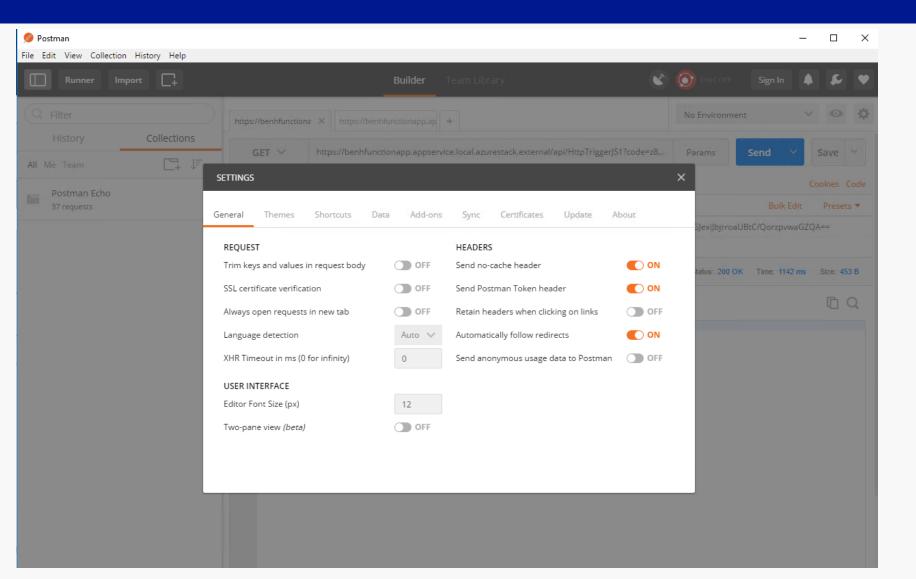*[The name of your function]*

Type
npm install request

Test your Azure Stack Function

Once you've configured the Function and test settings, click 'Run'
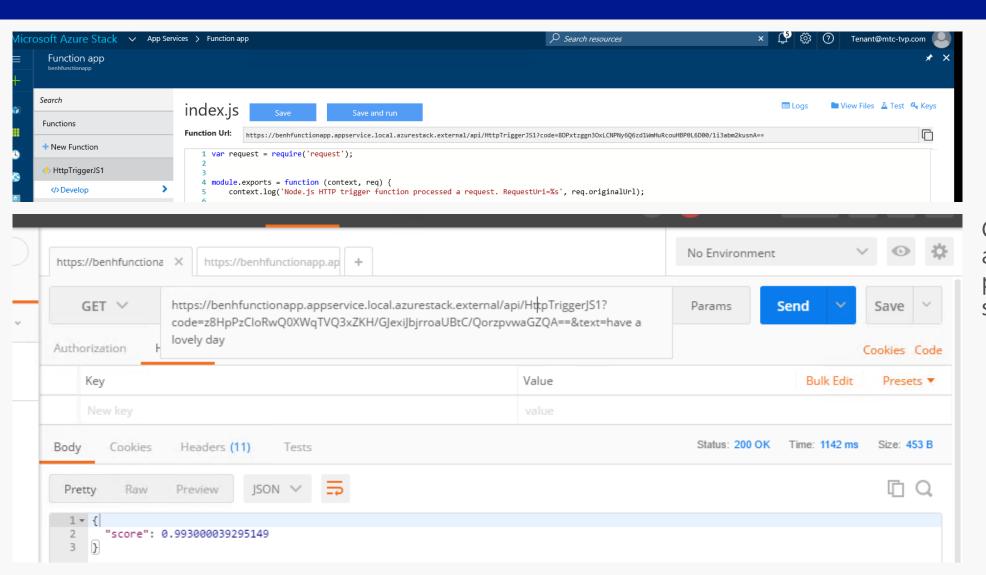
Configure Postman for Azure Stack

On the Azure Stack server, in Postman you may need to set "SSL certificate verification" to "Off"

Test your Azure Stack Function with Postman

Get the Function URL from the function app UI

Copy the Function URL and append any required parameters to the query string if a GET request.

Try other Cognitive Service & Functions

This code works on Function Apps for Azure and Azure Stack

Go to my Github Repo https://github.com/ben-houghton/summitcode

Name	URL	Configuration
Image Analysis	imageanalysisfunction.js	Replace the Function code with the code from the imageanalysisfunction.js file Use a POST request Use a POST body like {url: "http://linktoimage.jpg"} Use "Content-type: application/json" header
Key Word Analysis	keyphrasefunction.js	Replace the Function code with the code from the keyphrasefunction.js file Use a GET or POST request with a query string parameter of 'text'