

Developing a machine learning IoT app (with Node-Red and Tensorflow)

1. Setting up the “Node-Red” environment:

There are different ways to run NodeRed.

You can run NodeRed :

- Local install on a laptop or desktop (Mac, Windows, Linux)
- Raspberry Pi
- In a Cloud environment such as <http://cloud.ibm.com>

In this workshop we will use the IBM Cloud environment to run Node-Red.

- Login to IBM Cloud : <http://cloud.ibm.com>
- Open a Cloud Shell (in the top right corner)



This will open a Cloud Shell window within the ‘Frankfurt’ region

- Issue the command to switch to the ‘UK-Region’ where you should normally have your development space (you can verify this via “Manage” -> “Account” -> “Cloud Foundry orgs” -> “<your org>”):

```
# ic target -r eu-gb
```

```
# ic target -cf
```

You should now see that all fields in the results have been given a value (see example below):

```
Targeted Cloud Foundry (https://api.eu-gb.cf.cloud.ibm.com)
```

```
Targeted org ibmuser@gmail.com
```

```
Targeted space dev
```

```
API endpoint:      https://cloud.ibm.com
Region:           eu-gb
User:             ibmuser@gmail.com
Account:          IBM User's Account (bc8344c378a535a1db6b7f42db4bbfc8)
Resource group:   No resource group targeted, use 'ibmcloud target -g
RESOURCE_GROUP'
CF API endpoint:   https://api.eu-gb.cf.cloud.ibm.com (API version: 2.161.0)
Org:              ibmuser@gmail.com
Space:            dev
```


- Issue the following command to deploy a NodeRed instance :

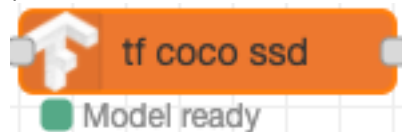
```
# ic cf push nodered --docker-image ydebeer/nodered_tensorflow -d eu-gb.mybluemix.net --random-route
```

This will deploy a pre-built docker image which contains NodeRed and it will generate a random unique URL (route) for you to access the application. Copy the route URL and paste it in a browser to open the NodeRed development environment.

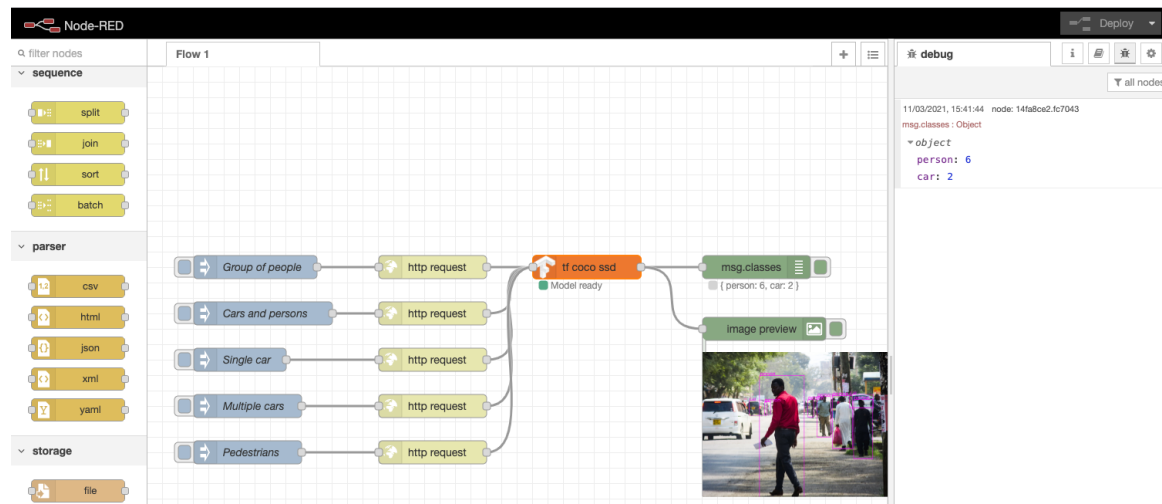
2. Exercise 1 - Deploy your first NodeRed flow with Tensorflow:

- Open the following URL in a new browser tab/window:
<https://flows.nodered.org/node/node-red-contrib-tfjs-coco-ssd>
- Deploy the Basic example from this web page:

- o Copy the JSON code to your clipboard
- o Go to the 'Import' tab in NodeRed (hamburger menu – top right corner)
- o Paste the JSON code and hit 'Import'
- o Click to fix the flow on the Development pane
- o Modify the parameters of the  node
 - Double click the node
 - Replace the 'Model Url'-value with `http://<your application route>/coco/model.json` – where <your application route> is the route url from your NodeRed which you obtained in step 1.
 - Click 'Done'
- o Click 'Deploy' (Red Button top right) to activate the flow (make sure the 'tf coco ssd' node indicates 'Model Ready')



- o Now you are ready to test the flow by clicking the inject node in front of the flow. Each inject will retrieve an image via a HTTP Request and analyze it to detect common objects.

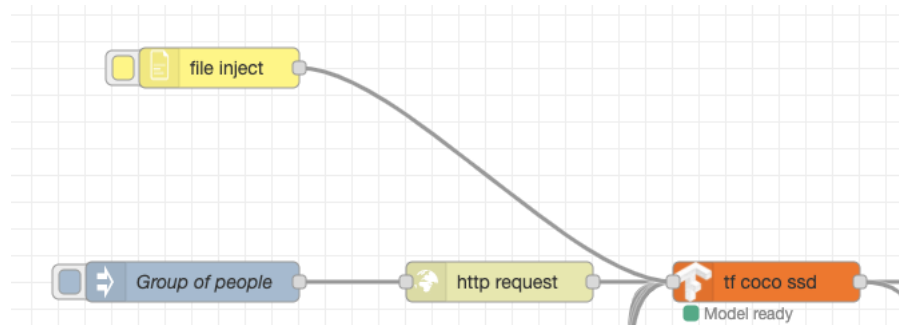


- The results are shown in the Image preview node as well as in the Debug tab

- Test with your own images

- Add some browser utilities to the Node-Red palette:

- Goto 'Manage Palette' in NodeRed (hamburger menu – top right corner)
- Click the 'Install' tab and search for module 'node-red-contrib-browser-utils' and click install. This will install 3 nodes: camera, file inject and microphone
- You can now drag and drop the 'file inject' node onto the canvas and connect it to the 'tf coco ssd' node.



- Click 'Deploy' in the top right corner to activate the modified flow.
- Click the button in front of the 'file inject' node to add an image from your local machine and have it analysed.

3. Tensorflow labs using Jupyter notebooks:

First deploy another Docker image into your Cloud environment using the IBM Cloud Shell.

Issue the following command:

```
# ic cf push tensorflow --docker-image tensorflow/tensorflow:latest-jupyter -d eu-gb.mybluemix.net --random-route -k2G -m2G
```

This will install and deploy a complete Jupyter environment for you to run some Tensorflow labs.

In order to login to the application you will need to retrieve the login token generated by the application.

Issue the following command in your IBM Cloud Shell to get the token:

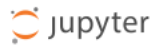
```
# ic cf logs tensorflow --recent
```

You should see something like the following:

```
2021-03-15T11:09:54.24+0000 [PROXY/0] OUT Exit status 137
2021-03-15T11:09:54.58+0000 [CELL/0] OUT Cell b9c18201-34e4-4668-b23d-ad1dbd064b89 successfully destroyed container for instance b124ddce-8fd5-4e45-40bc-b6df
2021-03-15T11:10:54.57+0000 [CELL/0] OUT Cell c907a583-d5e3-406b-b41c-87eb7c32508f successfully created container for instance abe5bfe1-fd8d-4cbd-6d96-1c27
2021-03-15T11:10:54.81+0000 [CELL/0] OUT Starting health monitoring of container
2021-03-15T11:10:56.64+0000 [APP/PROC/WEB/0] ERR [I 11:10:56.642 NotebookApp] Writing notebook server cookie secret to /root/.local/share/jupyter/runtime/notebook_cookie_secret
2021-03-15T11:10:56.98+0000 [APP/PROC/WEB/0] ERR [I 11:10:56.987 NotebookApp] Serving notebooks from local directory: /tf
2021-03-15T11:10:56.98+0000 [APP/PROC/WEB/0] ERR [I 11:10:56.987 NotebookApp] Jupyter Notebook 6.2.0 is running at:
2021-03-15T11:10:56.98+0000 [APP/PROC/WEB/0] ERR [I 11:10:56.987 NotebookApp] http://abe5bfe1-fd8d-4cbd-6d96-1c27:8888/?token=b7ca0fb8105b4fbb92b70e06fccdf5a702d111ea16415013
2021-03-15T11:10:56.98+0000 [APP/PROC/WEB/0] ERR [I 11:10:56.988 NotebookApp] or http://127.0.0.1:8888/?token=b7ca0fb8105b4fbb92b70e06fccdf5a702d111ea16415013
2021-03-15T11:10:56.99+0000 [APP/PROC/WEB/0] ERR [I 11:10:56.988 NotebookApp] Use Control-C to stop this server and shut down all kernels (twice to skip confirmation).
2021-03-15T11:10:56.99+0000 [APP/PROC/WEB/0] ERR [C 11:10:56.993 NotebookApp]
2021-03-15T11:10:56.99+0000 [APP/PROC/WEB/0] ERR To access the notebook, open this file in a browser:
2021-03-15T11:10:56.99+0000 [APP/PROC/WEB/0] ERR file:///root/.local/share/jupyter/runtime/nbserver-19-open.html
2021-03-15T11:10:56.99+0000 [APP/PROC/WEB/0] ERR Or copy and paste one of these URLs:
2021-03-15T11:10:56.99+0000 [APP/PROC/WEB/0] ERR http://abe5bfe1-fd8d-4cbd-6d96-1c27:8888/?token=b7ca0fb8105b4fbb92b70e06fccdf5a702d111ea16415013
2021-03-15T11:10:56.99+0000 [APP/PROC/WEB/0] ERR or http://127.0.0.1:8888/?token=b7ca0fb8105b4fbb92b70e06fccdf5a702d111ea16415013
2021-03-15T11:10:58.60+0000 [CELL/0] OUT Container became healthy
```

Copy the token obtained from the logs of the application (Red Text – search for token=....)

Open the URL (generated by the previous deployment) in a new browser tab – you should see the following page:



Password or token:

Log in

Token authentication is enabled

If no password has been configured, you need to open the notebook server with its login token in the URL, or paste it above. This requirement will be lifted if you [enable a password](#).

The command:

```
jupyter notebook list
```

will show you the URLs of running servers with their tokens, which you can copy and paste into your browser. For example:

```
Currently running servers:  
http://localhost:8888/?token=c8de56fa... :: /Users/you/notebooks
```

or you can paste just the token value into the password field on this page.

See [the documentation on how to enable a password](#) in place of token authentication, if you would like to avoid dealing with random tokens.

Cookies are required for authenticated access to notebooks.

Setup a Password

You can also setup a password by entering your token and a new password on the fields below:

Token

New Password

Log in and set new password

Use the token obtained from the logs of the application (Red Text – search for token=....)

Once you are logged in you should see:



Files

Running

Clusters

Select items to perform actions on them.

☐ 0



/

☐



tensorflow-tutorials

Start the 'regression' exercise first and finish by doing the steps described in the 'classification' tutorial.