Cognitive Home - Watson Visual Recognition Lab

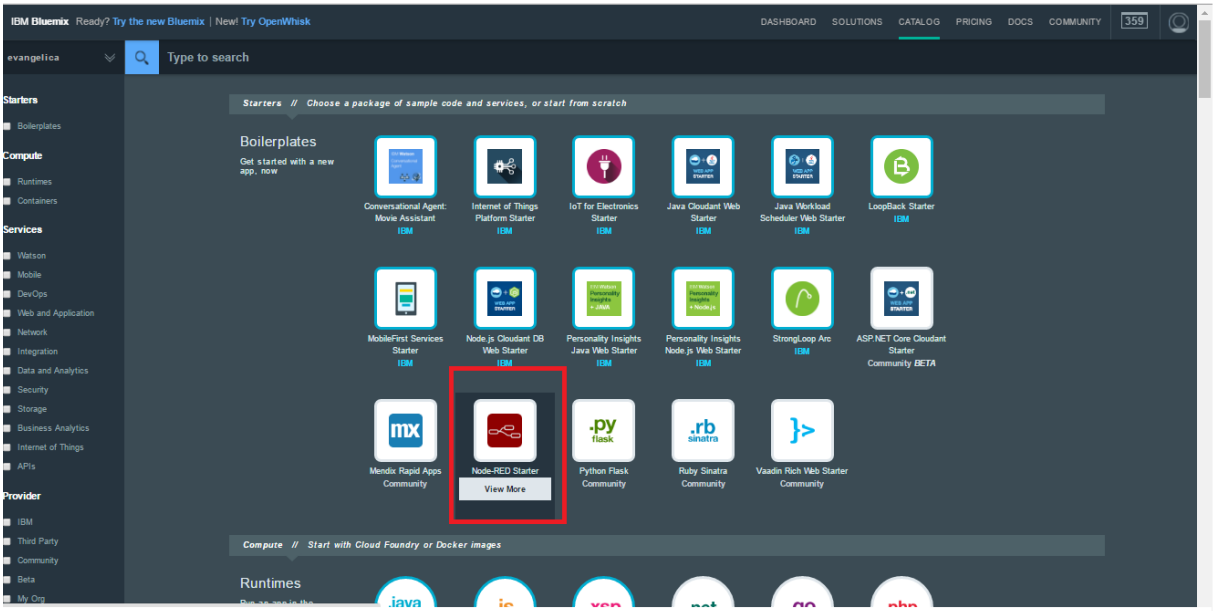
Created by Shubhradeep Nandi (AI, Cognitive and DataScience expert)

1 OBJECTIVE

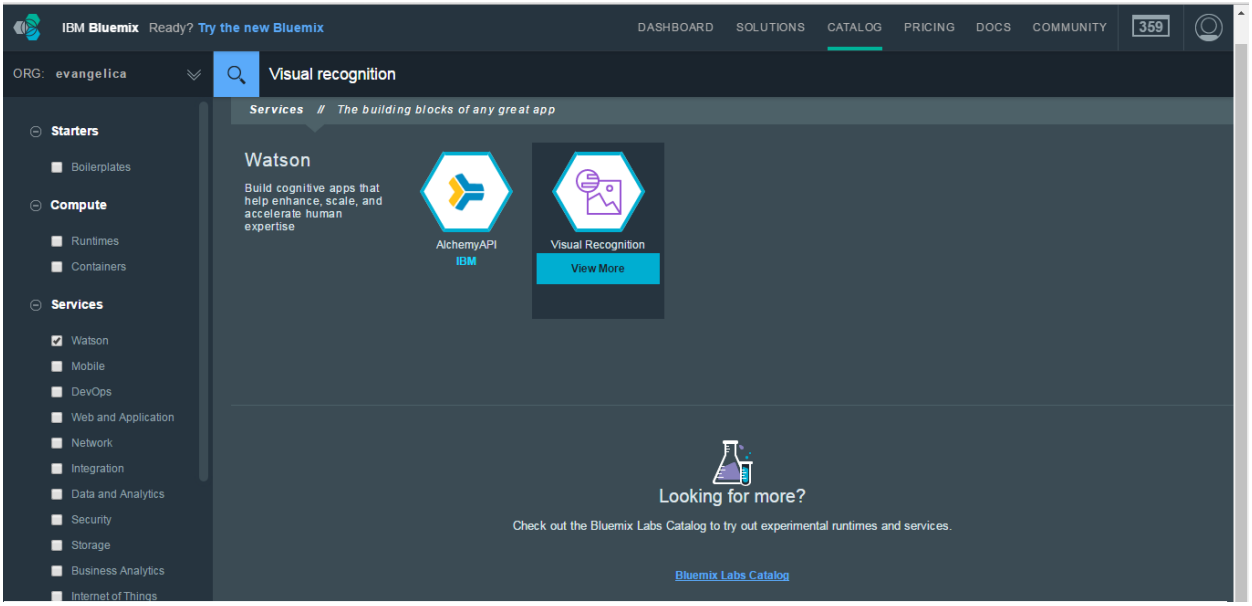
To build an intelligent security system that can visually recognize individuals and give them access.

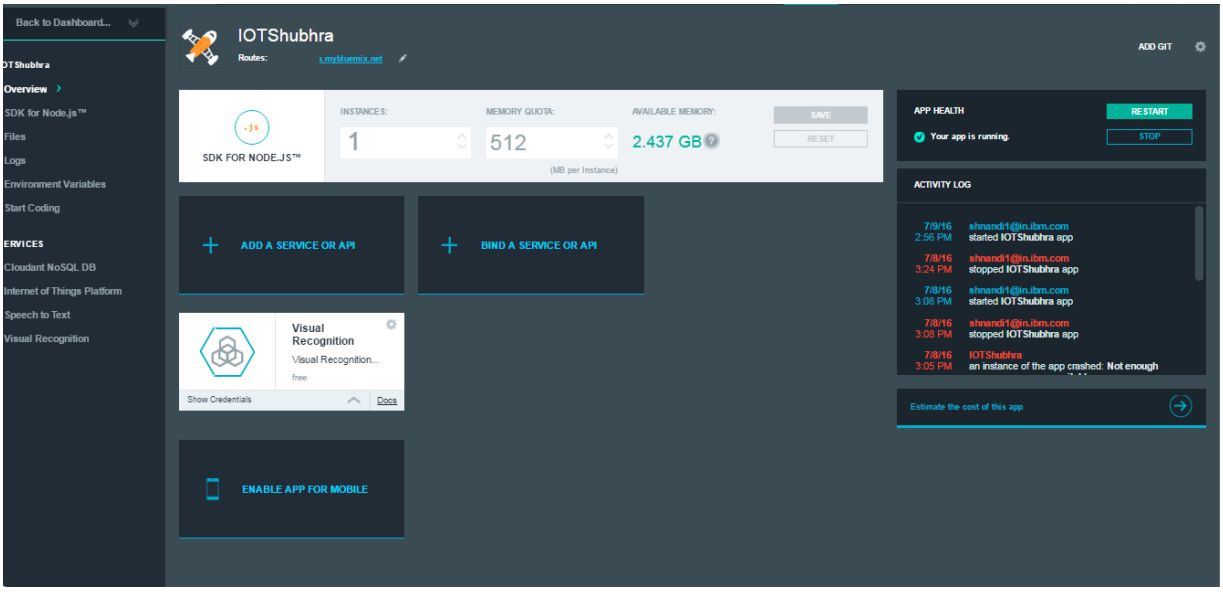
2 CREATE THE APPLICATION BASE

1. Click on the https://console.ng.bluemix.net/catalog/starters/node-red-starter/
2. Create a Node-Red Application



3. Add a Watson Visual Recognition service to the application.

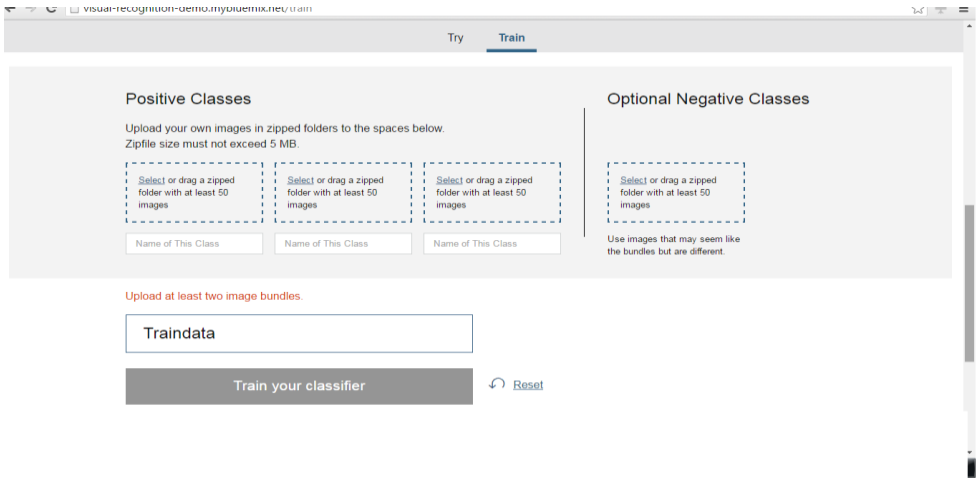




3 TRAINING THE SYSTEM

1. Take around 15 selfies each from at least two of your team members.
2. Create a zip file each for two set of images.
3. Go to http://visual-recognition-demo.mybluemix.net/train

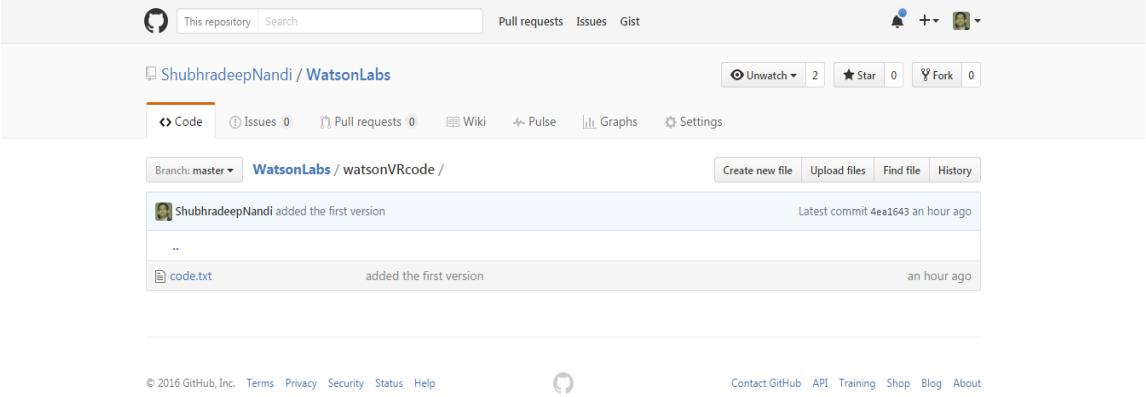
- Try to upload your zip files and create a custom classifier.



4. If step 3 does not work for you then follow step 5-7
5. (Optional) Alternatively, Open up the command prompt.
6. (Optional) Go to your folder where the selfie zips are stored.
7. (Optional) Run the below curl command to train.
   - curl -X POST -F "<person1-name>_positive_examples=@<person1-name>.zip" -F "<person2_positive_examples=@person2.zip" -F "negative_examples=@negative.zip" -F "name=selfiesec" "https://gateway-a.watsonplatform.net/visual-recognition/api/v3/classifiers?api_key=<Your_API_KEY>&version=<TRAINER WILL PROVIDE>"
8. This will take up to a maximum 15 minutes for the system to get trained with your customer data.
9. Note the classifier id for future references.

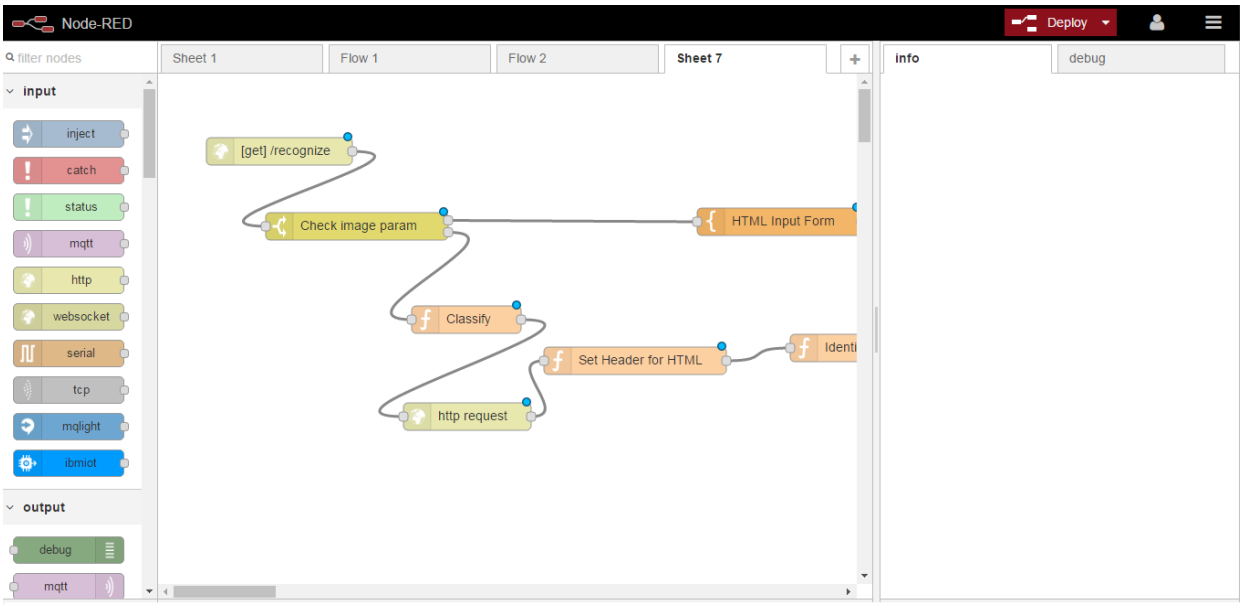
4 CREATING THE YOUR VISUAL AUTHENTICATION SYSTEM.

1. Go to GitHub repo https://github.com/ShubhradeepNandi/WatsonLabs.git and click on watch + star + fork



2. Get the code.txt from **watsonVRcode**

3. Import the code text in to Node Red application Sheet space.



4. Modify the below items :-

   - In the **Classify** function node modify this line *"msg.url="https://gateway-a.watsonplatform.net/visual-recognition/api/v3/classify?&api_key=1234myapikey&url="+encodeURI(msg.payload.url)+"&classifier_ids=**<ADD YOUR OWN CLASSIFIER ID>**&owners=me,IBM&threshold=0.2&version=**<ADD THE API VERSION PROVIDED >**";"*

   - In the **Identification** function node modify this line *"if(msg.class === '**<Add your first class>**' || msg.class === '**<Add your second class>**'){ "*

5. Deploy the code and you are done.

5 HANDSHAKE WITH IOT

1. IOT code should make a GET call to the below API :-
   - ***<Your Node Red App url> /recognize?url=<Image url>***
   - A parameter '**url**' should be appended to the api call where you need to send an Image URL [*How and where to upload an Image will be informed during the Lab*]
2. This application will send a JSON response in the below format :-

   *{"custom_classes":2,"images":[{"classifiers":[{"classes":[{"class":"shubhra","score":0.237042}],"classifier_id":"myclassifier","name":"selfiesec"}],"resolved_url":"https://****/$$$$/shubhra2.jpg","source_url":"https://****/$$$$/shubhra2.jpg"}],"images_processed":1,"**authorize**":**1**}*

   'authorize' is the parameter to let your IOT system know how to respond.