

IMAGE RECOGNITION SYSTEM USING IBM CLOUD VISUAL RECOGNITION.

To create a image recognition using IBM Cloud, including setting up an IBM Cloud account, creating a Visual Recognition service, creating IBM Cloud. In this integration, IBM Cloud Functions serves as the event-driven architecture where you can trigger image recognition functions in response to events such image uploads. When an event is triggered, it invokes a serverless function, which then utilizes Visual Recognition to analyze and interpret the images.

The application demonstrates an IBM Cloud Functions (based on Apache OpenWhisk) that gets an image from the Cloudant database and classifies it through Watson Visual Recognition. The use case demonstrates how actions work with data services and execute logic in response to Cloudant events.

Steps :

1. Clone the repo

\$ git clone <https://github.com/IBM/serverless-image-recognition>

2. Create IBM Cloud Services

Create a [Cloudant](#) instance and choose Use both legacy credentials and IAM for the *Available authentication method* option.

- Create credentials for this instance and copy the username and password in the local.env file in the value of CLOUDANT_USERNAME and CLOUDANT_PASSWORD.

- Launch the Cloudant web console and create a database named images and tags. Create Cloudant credentials using the IBM Cloud dashboard and place them in the local.env file.

Create a [Watson Visual Recognition](#) instance.

- Copy the API Key in the Credentials section and paste it in the local.env file in the value of WATSON_VISUAL_APIKEY

3. Deploy Cloud Functions

Create 2 databases in cloudant:

1. images
2. tags

Deploy through the IBM Cloud Functions console user interface

Choose ["Start Creating"](#) in the IBM Cloud Functions Dashboard. [Then proceed to this deployment instructions using the UI.](#)

You can also deploy them directly from the CLI by following the steps in the next section.

Deploy using the wskdeploy command line tool

This approach deploy the Cloud Functions with one command driven by the runtime-specific manifest file available in this repository.

Make sure you have the right environment variables in the local.env file. Export them in your terminal then deploy the Cloud Functions using wskdeploy. This uses the manifest.yaml file in this root directory.

```
$ source local.env
```

```
$ wskdeploy
```

Alternative Deployment methods

Deploy manually with the ibmcloud wsk command line tool

This approach shows you how to deploy individual the packages, actions, triggers, and rules with CLI commands. It helps you understand and control the underlying deployment artifacts.

- Export credentials

```
$ source local.env
```

- Create Cloudant Binding

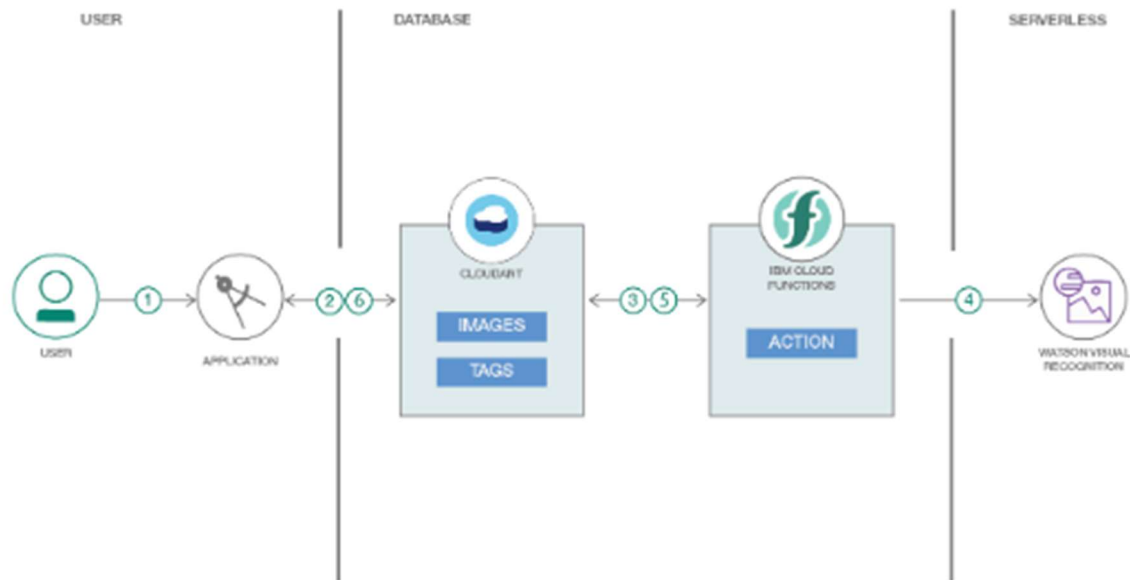
```
$ ibmcloud wsk package bind /whisk.system/cloudant  
serverless-pattern-cloudant-package \  
-p username $CLOUDANT_USERNAME \  
-p password $CLOUDANT_PASSWORD \  
-p host ${CLOUDANT_USERNAME}.cloudant.com
```

- Create the Cloudant Trigger

The trigger will listen to changes in the `images` database.

```
$ ibmcloud wsk trigger create update-trigger --feed serverless-  
pattern-cloudant-package/changes \  
--param dbname images
```

Structure:



Program :

cache: pip

before_install:

- sudo apt-get install shellcheck
- sudo pip install yamllint
- git clone <https://github.com/IBM/pattern-ci>

before_script:

- `"/pattern-ci/tests/shellcheck-lint.sh"`
- `"/pattern-ci/tests/yaml-lint.sh"`

jobs:

include:

- script: echo "Lints passed."

```
$ ibmcloud wsk package delete serverless-pattern-cloudant-package
```

```
$ ibmcloud wsk trigger delete update-trigger
```

```
$ ibmcloud wsk action delete update-document-with-watson
```

```
$ ibmcloud wsk rule delete update-trigger-rule
```

package :

```
{
  "name": "serverless-image-recognition",
  "version": "1.0.0",
  "description": "serverless",
  "main": "main.js",
  "directories": {
    "doc": "docs"
  },
  "scripts": {
    "start": "electron ."
  },
  "repository": {
    "type": "git",
    "url": "git+https://github.com/AnthonyAmanse/serverless-image-
recognition.git"
  },
  "author": "",
  "license": "ISC",
  "bugs": {
    "url": "https://github.com/AnthonyAmanse/serverless-image-
recognition/issues"
  },
  "homepage": "https://github.com/AnthonyAmanse/serverless-image-
recognition#readme",
  "devDependencies": {
```

```
    "electron": "^7.1.2"  
  }  
}
```

Conclusion :

Creating an image recognition system using the IBM Cloud Visual Recognition in a cloud computing project allows for powerful image analysis capabilities that can be integrated into various applications and solutions.

Image recognition is a crucial application of artificial intelligence that involves the interpretation of images to extract meaningful information. IBM Cloud Visual Recognition is a state-of-the-art AI service that allows for image analysis, classification, and extraction of important features from images.

In conclusion, implementing an image recognition system using IBM Cloud Visual Recognition in a cloud computing project offers a powerful solution for automating image analysis, which can find applications in a wide range of domains, ultimately improving processes, decision-making and the user experiences.