Mobiliya

Security & Surveillance

Setup and installation guide

Table of Contents

1. About this Document

This document explains steps needed to setup and configure Security and Surveillance solution.

2. Intended Audience

This document is intended for IT administrators who will be setting up required on-premise modules and cloud services.

3. Introduction

Security and surveillance solution is based on 3 pillars

1) Aggregator – Connects with existing infrastructure. Fetches Camera streams and integrates them in system. A solution can consist of one or more aggregators.
2) ComputeEngine – Core module responsible for detection and tracking. A solution can consist of one or more compute-engines.
3) Backend Server – REST api server which acts as a coordinator and gatekeeper for overall solution.

4. Prerequisite

Below prerequisites should be addressed before starting deployment and configuration.

1) Nvidia Jetson board
2) Linux Machine
3) Jetson jetpack
4) Azure account
5) Camera rtsp urls

5. Take one step at a time

Overall setup and configuration can be divided as per below steps.

1) Create packages for cloud services
2) Video Indexer
3) Azure B2C Setup
4) Azure cloud Setup
5) PowerBI Deployment
6) Preparing Jetson board
7) Installing Aggregator module
8) Installing ComputeEngine

5.1 Create packages for cloud services

We are going to upload our Webapp, BackendServer (BS) and CloudComputeEngine(CCE) to azure.

Note: This step can be ignored if Webapp, BackendServer and CloudComputeEngine packages are already uploaded on publicly accessible blob.

Prerequisite:

1) Azure Account
2) Code Repository access
3) Visual Studio 2017 with NodeJS
4) NodeJS
5) Angular cli for package creation

Please refer to 'Step 1 Package Creation 1.0.0' to complete this step.

Note:

a) Same procedure needs to be followed for packaging and uploading CCE and BS (any node module)
b) Keep note of deployment urls, we will need it while deploying to azure.

5.2 Azure B2C Setup

We are going to configure and deploy Azure Active Directory B2C.

Prerequisite:

1) Azure Account

Please refer to 'Step 2 B2C Setup 1.0.0' to complete this step.

Note:

We will need following information in further steps.

i. B2C client Id
ii. B2C Policy
iii. B2C Tenant name
iv. B2C Scope

5.3 Video Indexer

We are going to enable video indexer service in this step.

Prerequisite:

1) Azure Account

Please refer to 'Step 3 Video Indexer 1.0.0' to complete this step.

Note:

We will need following information in further steps.

i. Video Indexer key

5.4 Azure Cloud Setup

We are going to configure and deploy required Azure resources in this step. We will also deploy packages uploaded in previous step.

Prerequisite:

1) Azure Account
2) Video Indexer key. [refer document for instruction, if not available]
3) Deployment urls for webapp, cloud compute engine, and backend server.

Please follow companion document titled 'Step 4 ARM Deployment 1.0.0' to complete this step.

Note:

We will need following values in further steps:

i. IoT Hub connection string
ii. Storage account name
iii. Storage account access key
iv. Backend url
v. Webapp url

5.5 PowerBI Deployment

We are going to deploy PowerBI reports in this step.

Prerequisite:

1) Credentials for signing to https://app.powerbi.com
2) PowerBI Desktop (> April 2018)
3) PowerBI Template file ($(repo_url)/SnSPowerBI/Templates)

Please refer to 'Step 5 PowerBI Deployment 1.0.0' to complete this step.

Note:

We will need following values in further steps:

i. Report and FaceDetection Embed Urls
ii. PowerBI reportIds (reportId from Embed Url)
iii. Account Username
iv. Account password
v. Client Id
vi. Client Secret

5.6 Preparing Jetson board

In this step, we are going to flash Nvidia Jetson board.

Prerequisite:

1) Nvidia Jetson board
2) Host Linux machine
3) Nvidia Jetson Jetpack
4) Router
5) Micro B usb cable

Please refer to 'Step 6 Jetson Flashing 1.0.0' to complete this step.

5.7 Installing Aggregator Module

We will setup aggregator module in this step.

Prerequisite:

1) Linux machine/Jetson flashed with Ubuntu
2) NodeJS 4.0 or above [refer instructions below]
3) Python 2.7 or above
4) OpenCV 3 or above [refer instructions below]
5) git

5.7a Installation

Install NodeJS (if not available)

1. $ curl -sL https://deb.nodesource.com/setup_6.x | sudo -E bash -
2. $ sudo apt-get install -y nodejs

Install OpenCV (if not available)

1. $ git clone $(repo_url)/jetson-device-client
2. $ cd jetson-device-client
3. $ mv install-opencv.sh ~/
4. $ cd ~/
5. $ chmod +x install-opencv.sh
6. $ ./install-opencv.sh

Please follow below steps (in terminal) to install aggregator.

1) **Clone Repository**
   i. $ git clone $(repo_url)/Aggregator. [Note: use ‘jetson-only’ branch if we want to run aggregator on jetson]
2) **Install npm packages**
   i. $ cd Aggregator
   ii. $ npm run pythonPackages
   iii. $ npm install

5.7b Configuration

1. Switch to Aggregator folder: $ cd Aggregator
2. Open config.js file
3. Update config.aggregatorName and config.location
4. Update following placeholder with values acquired in previous steps
   a. <IOTHubConnectionString> => IoT Hub connection string
   b. <storageAccountName> => Storage Account Name
   c. <storageAccountAccessKey> => Storage Account Access Key
   d. <backendUrl> => Backend Url (should start with https://)

5.7c Get Set Go!

3) **Start Aggregator (follow for development setup)**
   i. $ node aggregatorServer.js

4) **Start Aggregator in background (follow for production setup)**
   i. $ sudo npm install forever -g
   ii. $ forever start aggregatorServer.js
   iii. $ forever list (optional: to debug)
   iv. $ tail -100f ${log file path from above list} (optional: to debug)

5.8 Installing Compute Engine

In this step, we will install basic compute engine responsible to Human and object detection.

Prerequisite:

1) Nvidia Jetson with ubuntu
2) git

5.8a Installation

Please follow below steps (Jetson command-line) to install Compute Engine.

1) **Clone Repository**
   i. $ git clone $(repo_url)/ComputeEngine [Note: use 'jetson-only' branch if we want to run aggregator on jetson]
2) **Install required packages**
   i. $ sudo apt-get install cmake curl
   ii. $ curl -sL https://deb.nodesource.com/setup_6.x | sudo -E bash –
   iii. $ sudo apt-get install -y nodejs
   iv. $ sudo apt-get install -y libjson0 libjson0-dev
   v. $ sudo apt-get install -y libjson0-dbg
   vi. $ sudo apt-get install -y libcurl4-gnutls-dev
3) **Install Base64 library**
   i. $ git clone https://github.com/bartobri/base64-simple.git
   ii. $ cd base64-simple
   iii. $ make
   iv. $ sudo make install
4) **Build Executable**
   i. $ cd ComputeEngine
   ii. $ cd darknet
   iii. $ make
5) **Install node modules**
   i. $ cd ../jetsonNodeServer
   ii. $ sudo npm install

5.8b Configuration

1. Switch to ComputeEngine/jetsonNodeServer folder: $ cd ComputeEngine/jetsonNodeServer
2. Open settings.js file
3. Update config.name and config.location
4. Update following placeholder with values acquired in previous steps
   a. <IOTHubConnectionString> => IoT Hub connection string
   b. <backendUrl> => Backend Url (should start with https://)

5.8c Get Set Go!

6) **Start server (follow for development setup)**
   i. $ node jetsonserver.js
7) **Start server in background (follow for production setup)**
   i. $ sudo npm install forever -g
   ii. $ forever start jetsonserver.js
   iii. $ forever list (optional: to debug)
   iv. $ tail -100f ${log file path from above list} (optional: to debug)

6. Verification

Please refer [Demo video]/[User guide] for Solution overview.

Note: User guide is not part of setup and configuration documents.

7. Troubleshooting

1. Bad request on sign in

   Cause: Request too long.

   Res: Clear Browsing data and retry

2. Loader stays for long time after sign in
   Cause:
   i. Backend not responding
   ii. Connectivity issues with backend server
3. Dashboard bar chart not displaying for

   Cause: DB query timed out

   Res: Increase timeout or increase RUs for cosmos db

4. Report tab not loading

   Cause: PowerBI not configured

   Res: configure PowerBI reports and add correct url in report 'settings'

5. Raw image not displayed

   Cause:

   a. Aggregator not able to connect with camera
   b. Aggregator not working
   c. Backend not responding

   Res: Restart Aggregator

6. Not able to connect camera

   Cause:

   a. Aggregator not able to connect with camera
   b. Backend not responding
   c. Aggregator not responding

   Res:

    d. Verify rtsp url
    e. Restart aggregator
    f. Restart Backend server

7. Start streaming not working. No live feed

   Cause:

    a. Aggregator not able to communicate with cameras
    b. Aggregator cannot communicate to Compute Engine
    c. Compute Engine not responding
    d. Backend server not responding

   Res:

   Aggregator is working fine if raw images are updated. Reason could be c, and d

   Aggregator not responding if raw images are not updating

8. Not able to see video retention result

   Cause: Blob storage not accessible

   Res: Verify blob storage credentials in aggregator configuration file.

9. Image not getting displayed consistently in live results for IP Camera

   Cause: IP Camera not able to stream feed

   Res: Restart IP Camera.