
VAMStack
Better Platform.
Better AI.

VAMStack
Design House

Name : VAMStack Co., Ltd.

TAX ID : 0105561080465

Founded : May, 10 2561

Paid-up Capital : 7,000,000 ฿

Member of :TESA, DEPA, DGTi

Products :

- Video Analytic Platform Provider
- Image Processing, ML, DL and AI Modules
- AI for Retail, Manufacturers, Security, Surveillance and Medical Solution

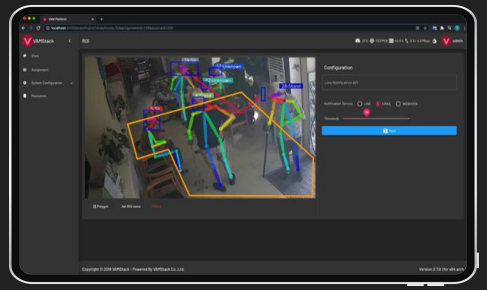
Sale Team
(Bangkok)

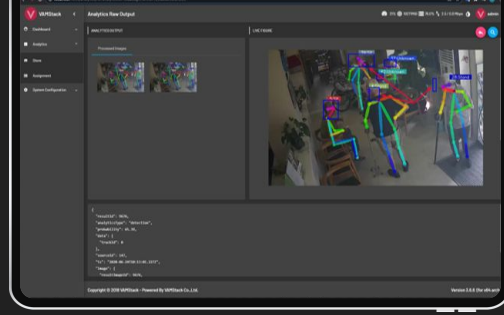
Developer Team
(Bangsaen Design House)

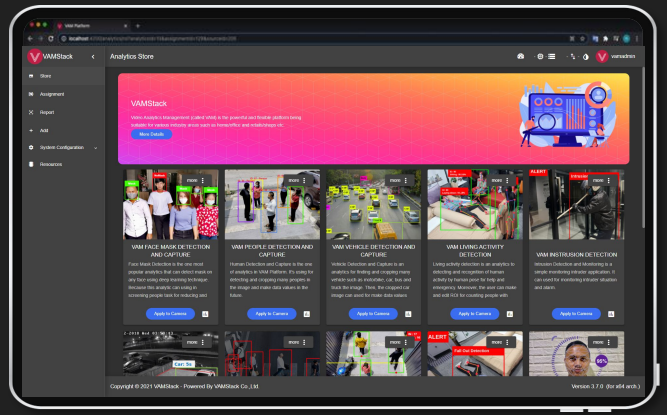
VAM Platform

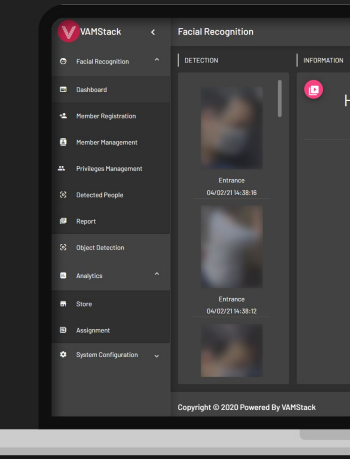
"เปิดโลกใหม่ด้วย AI ที่สามารถใช้งานในชีวิตจริงบน AI Analytics Platform"

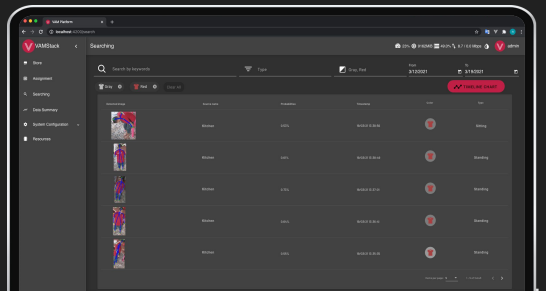
VAMSTACK

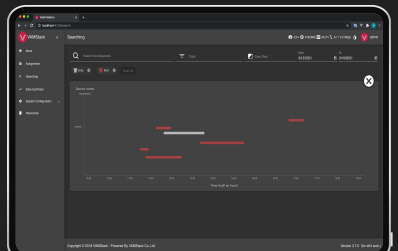












Core Technology

Home

Fall Out Detection

Intrusion Detection

Activity Recognition

Fire and Smoke Detection

Loitering Detection

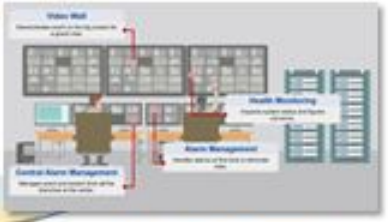
Vehicle Monitoring

People Monitoring

Building

Hotel Walkway
- Suspect Searching and Tracking
- Action Recognition

Office Department
- Face Recognition
- Access Control
- People Monitoring
- Workforce Management



Hotel Care Center
- Management and Service

Public Area
- Intrusion Detection
- Alarm Detector

Lobby
- Face Recognition
- Member Registration
- People Counting / in-out

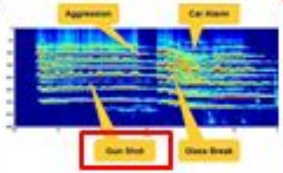
Parking
- Vehicle detection and Monitoring
- Parking Available (Outdoor)

City Surveillance

Intrusion Detection

Mobile Station

Central Command



Firearm and Bomb Sound Detection

Fire and Smoke Detection

People Screening and tracking

Face Recognition and Searching

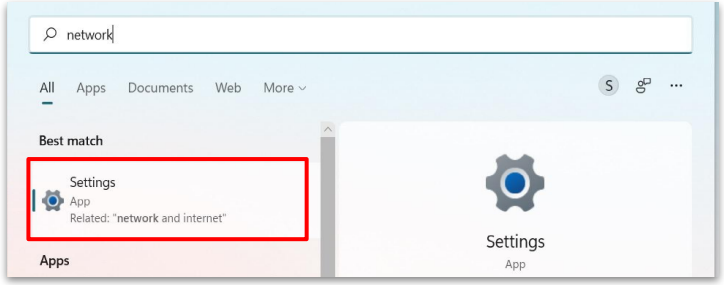
Object Recognition

Agenda

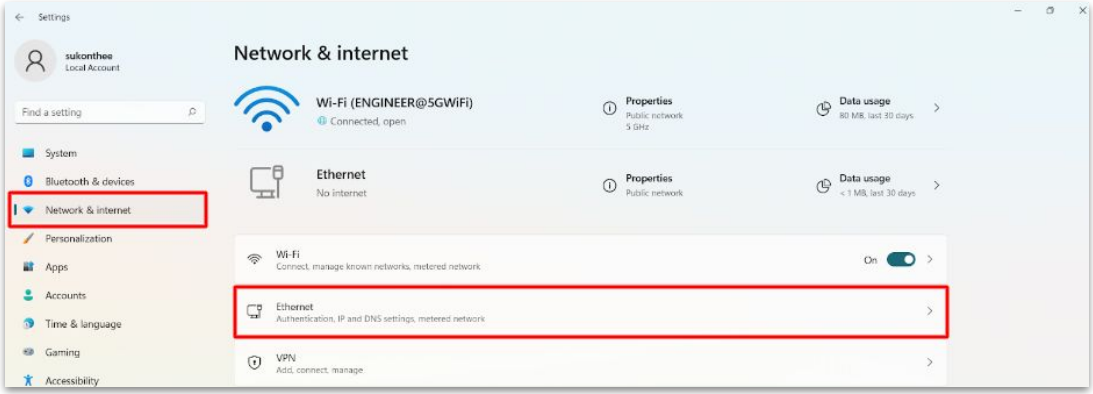
1. Connect NUC
    - 1.1. Windows OS
    - 1.2. Linux OS
    - 1.3. Mac OS
2. VAM Platform
3. VAM Studio
4. VAM SDK
    - 4.1. ฟังก์ชันสำหรับการรับข้อมูลของกล้องที่อยู่ใน VAM Platform
        - 4.1.1. listSource
        - 4.1.2. get_vamsnapshot
        - 4.1.3. get_vamroi
        - 4.1.4. draw_roi
    - 4.2. ฟังก์ชันพื้นฐานสำหรับการประมวลผลและวิเคราะห์ข้อมูล
        - 4.2.1. people_detection
        - 4.2.2. people_counting
    - 4.3. ฟังก์ชันสำหรับการส่งข้อมูลที่ประมวลผลเข้า VAM Platform
        - 4.3.1. set_vamlivefigure
        - 4.3.2. set_data2vamMetadata
5. VAM AIaaS
    - 5.1. Face Registration
        - 5.1.1. EnrollFace
    - 5.2. Face Recognition
        - 5.2.1. FaceFeature
    - 5.3. Face Detection and Quality Verification
        - 5.3.1. CompareFace

Connect NUC - Windows OS

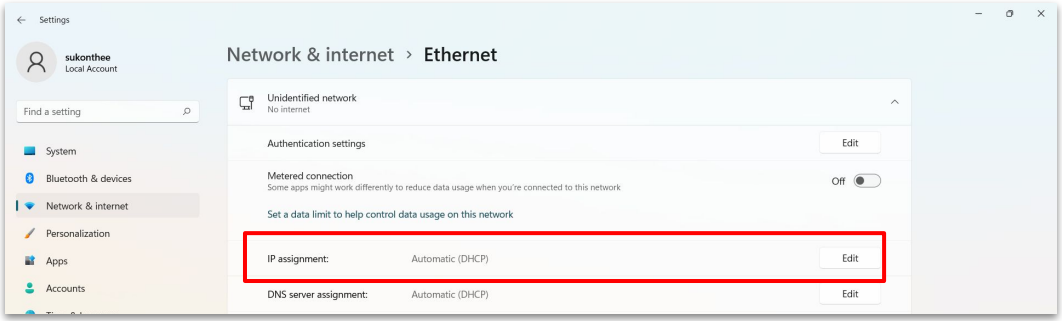
1. ตรวจสอบการเชื่อมต่อสายแลนระหว่างเครื่องของผู้เรียนกับ POE Switch เดียวกันกับเครื่องประมวลผล จากนั้นพิมพ์คำว่า “network” ในช่องค้นหาของระบบปฏิบัติการ windows และเลือก “Settings”



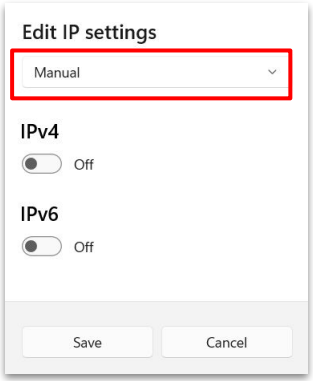
2. คลิก “Network & Internet” บริเวณแถบด้านซ้าย และเลือก “Ethernet”



3. กดปุ่ม “Edit” ในตัวเลือก “IP assignment”

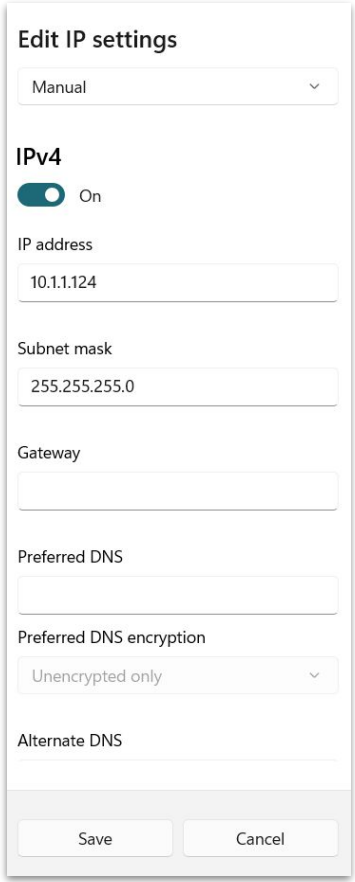


4. เปลี่ยนตัวเลือกจาก “DHCP” เป็น “Manual”



Connect NUC - Windows OS

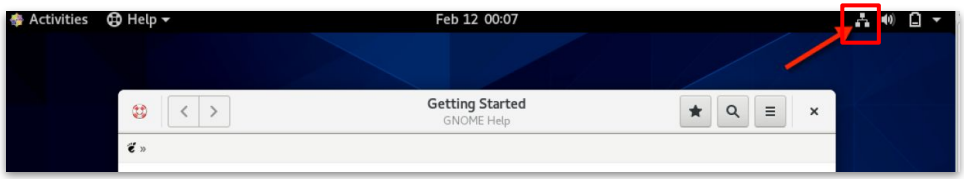
5. ปรับ IPv4 ให้เป็น “On”
   5.1. กำหนด “IP Address” ของเครื่องผู้เรียนเป็นวงแลน 10.1.1.xxx
   5.2. กำหนด “Subnet mask” เป็น 255.255.255.0
   5.3. คลิก “save”



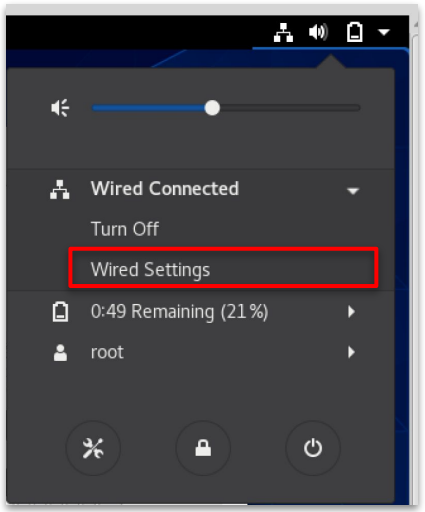
**หมายเหตุ

10.1.1.xxx หมายถึง ให้ผู้เรียนใส่ชุดหมายเลขเลขใด ๆ โดยไม่ซ้ำกันและห้ามใช้ชุดหมายเลข 10.1.1.1, 10.1.1.212, 10.1.1.165, 10.1.1.166, และ 10.1.1.224 เนื่องจากชุดหมายเลขดังกล่าวถูกนำไปใช้ในการเชื่อมต่อเครื่องประมวลผลและกล้อง

1. ตรวจสอบการเชื่อมต่อสายแลนระหว่างเครื่องของผู้เรียนกับ POE Switch เดียวกันกับเครื่องประมวลผล จากนั้นคลิกไอคอน “Network” บริเวณมุมบนด้านขวา



2. คลิกที่ “Wire Connected” แล้วเลือก “Wire Settings”



3. คลิกปุ่มรูปฟันเฟืองในตัวเลือก “Wired”

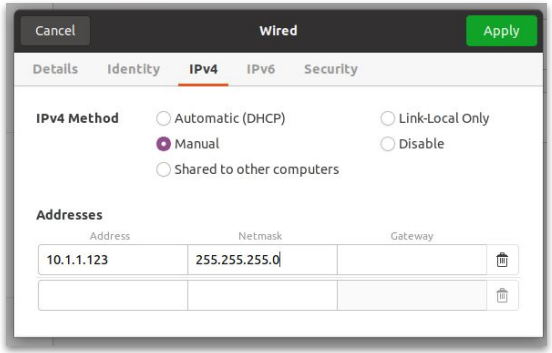
4. คลิกเลือก “IPv4”

   4.1. เปลี่ยน checkbox จาก “Automatic (DHCP)” เป็น “Manual”

   4.2. กำหนด “Address” ของเครื่องผู้เรียนเป็นวงแลน 10.1.1.xxx

   4.3. กำหนด “Netmask” เป็น 255.255.255.0

   4.4. คลิก “Apply”



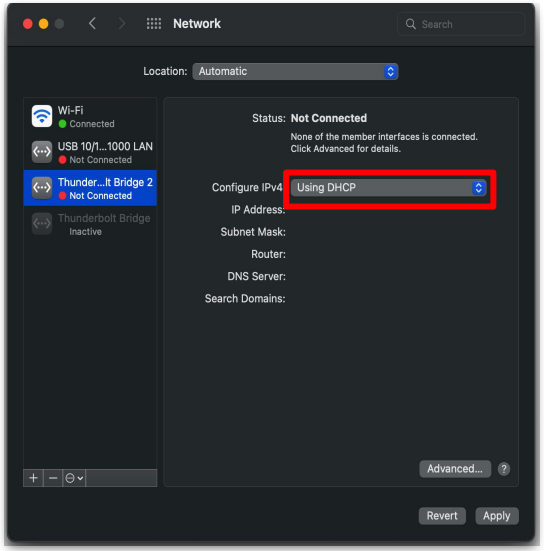
**หมายเหตุ

10.1.1.xxx หมายถึง ให้ผู้เรียนใส่ชุดหมายเลขเลขใด ๆ โดยไม่ซ้ำกันและห้ามใช้ชุดหมายเลข 10.1.1.1, 10.1.1.212, 10.1.1.165, 10.1.1.166, และ 10.1.1.224 เนื่องจากชุดหมายเลขดังกล่าวถูกนำไปใช้ในการเชื่อมต่อเครื่องประมวลผลและกล้อง

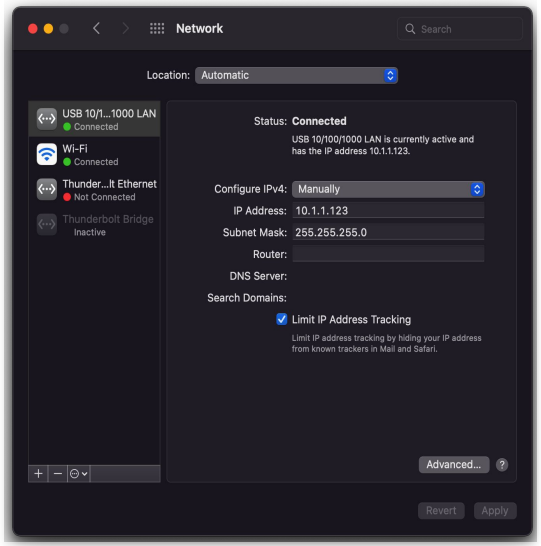
Connect NUC - Mac OS

1. ตรวจสอบการเชื่อมต่อสายแลนระหว่างเครื่องของผู้เรียนกับ POE Switch เดียวกันกับเครื่องประมวลผล จากนั้นค้นหาคำว่า “Network” ด้วย Spotlight บน Mac และคลิกเลือก Network

2. เลือกการเชื่อมต่อผ่านสาย LAN และเปลี่ยนตัวเลือก Configure IPv4 ให้เป็น “Manually”



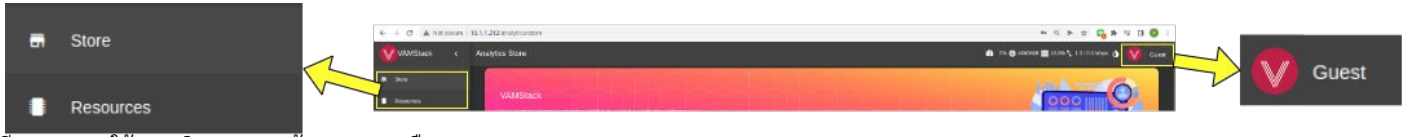
3. กำหนด “IP Address” ของเครื่องผู้เรียนเป็นวงแลน 10.1.1.xxx และ “Subnet Mask” เป็น 255.255.255.0 จากนั้นคลิก “Apply”



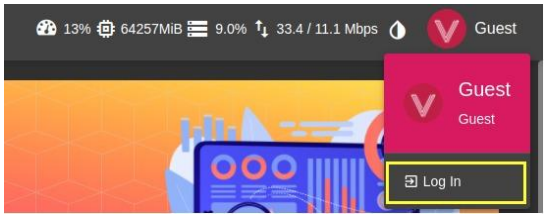
**หมายเหตุ

10.1.1.xxx หมายถึง ให้ผู้เรียนใส่ชุดหมายเลขเลขใด ๆ โดยไม่ซ้ำกันและห้ามใช้ชุดหมายเลข 10.1.1.1, 10.1.1.212, 10.1.1.165, 10.1.1.166, และ 10.1.1.224 เนื่องจากชุดหมายเลขดังกล่าวถูกนำไปใช้ในการเชื่อมต่อเครื่องประมวลผลและกล้อง

1. เข้าสู่หน้า VAM Platform โดยพิมพ์ “10.1.1.212” บนเว็บเบราว์เซอร์ จะปรากฏหน้าเริ่มต้นการเข้าใช้งานของ VAM Platform ในสถานะผู้เยี่ยมชม (Guest)

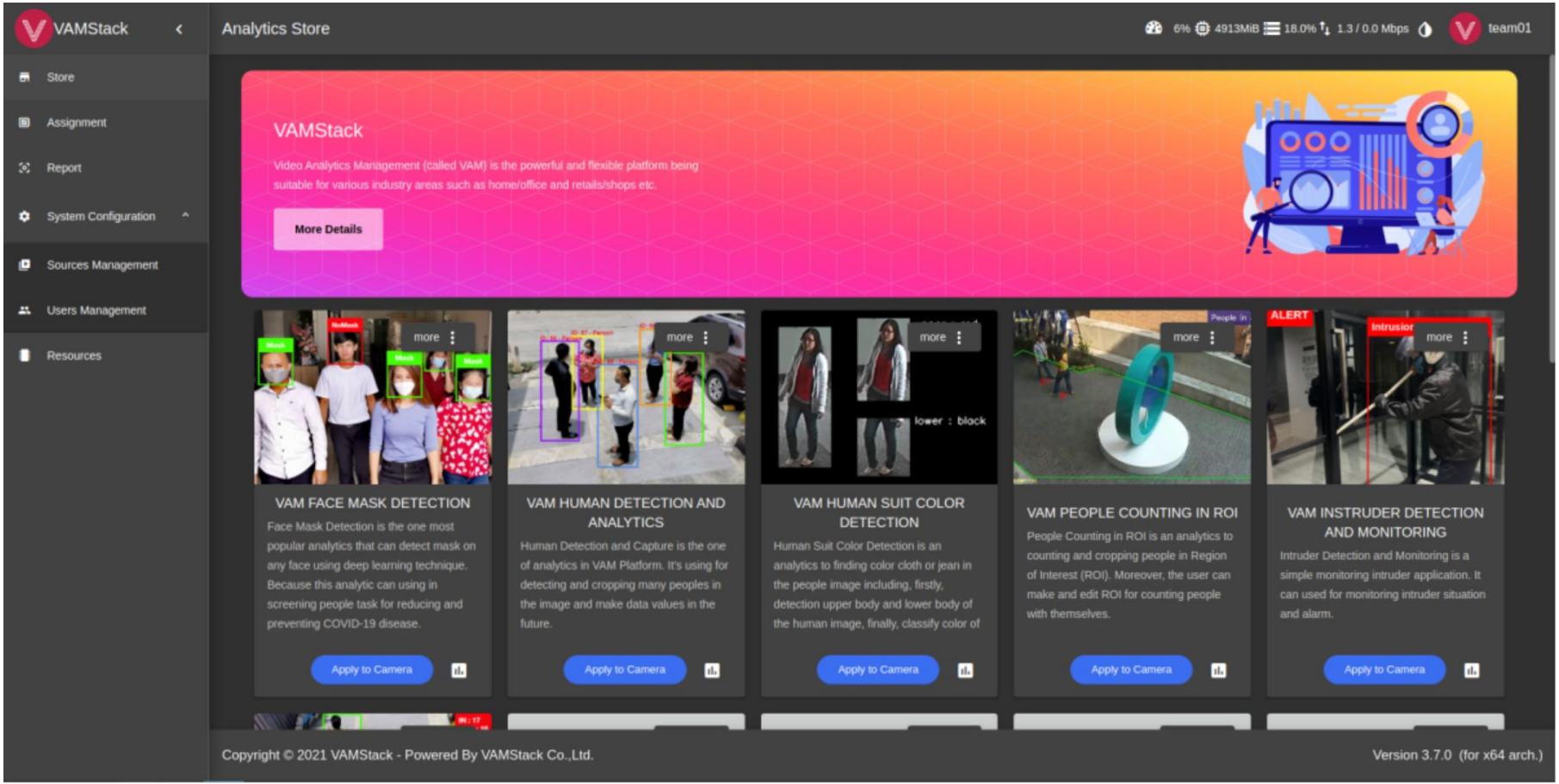


2. คลิกที่สถานะการใช้งานบริเวณมุมบนด้านขวาและเลือก “Log In”

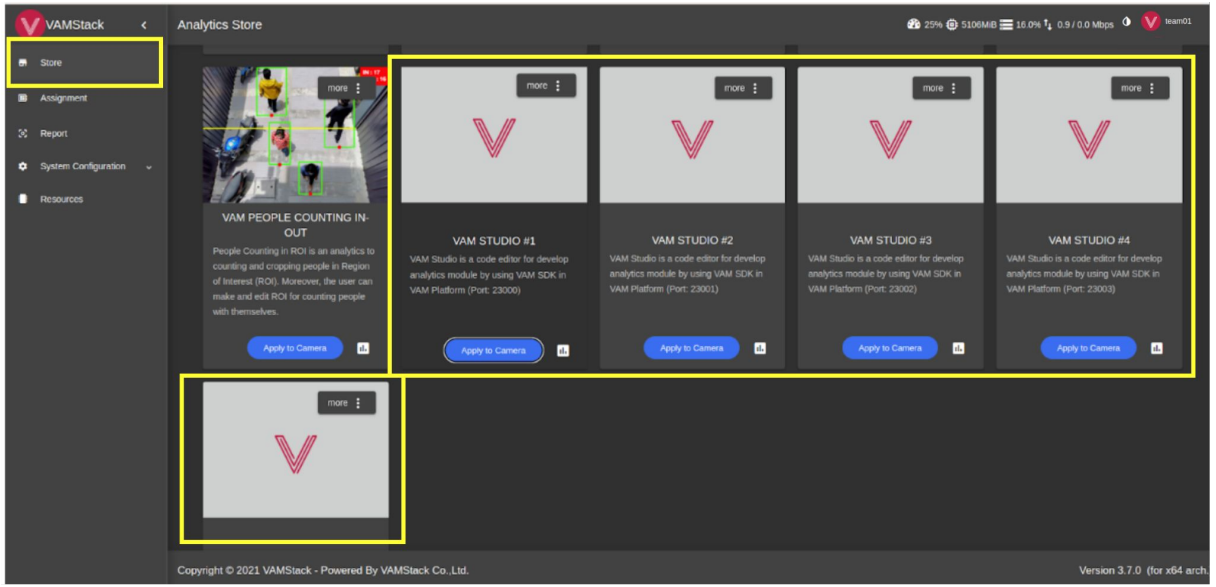


3. เข้าสู่ระบบโดยการใช้บัญชีที่ทางทีมผู้พัฒนาจัดเตรียมให้และคลิก “Log in”
   - 3.1. username: team01, password: study01
   - 3.2. username: team02, password: study02
   - 3.3. username: team03, password: study03
   - 3.4. username: team04, password: study04
   - 3.5. username: team05, password: study05

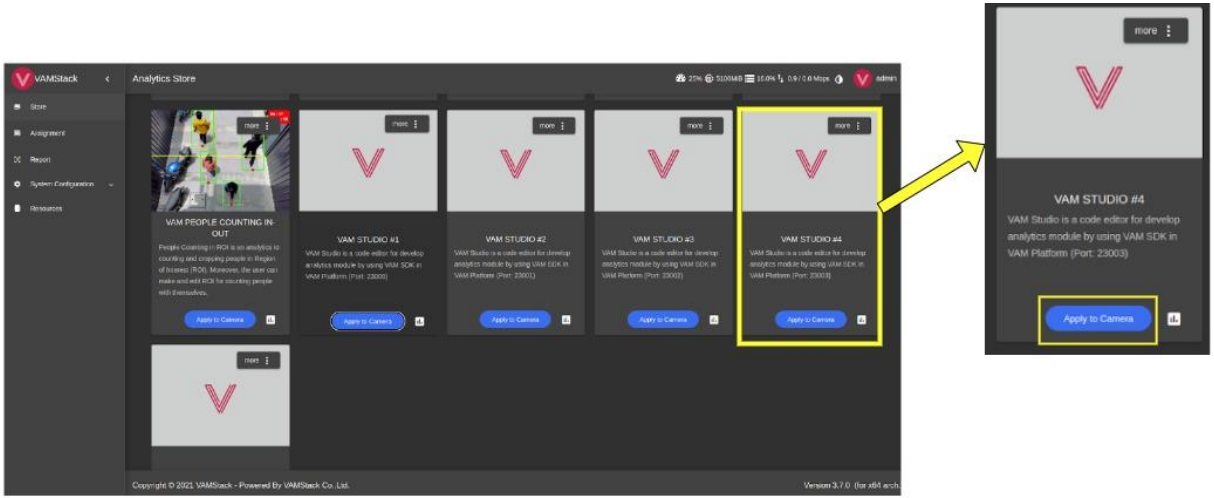
VAM Platform



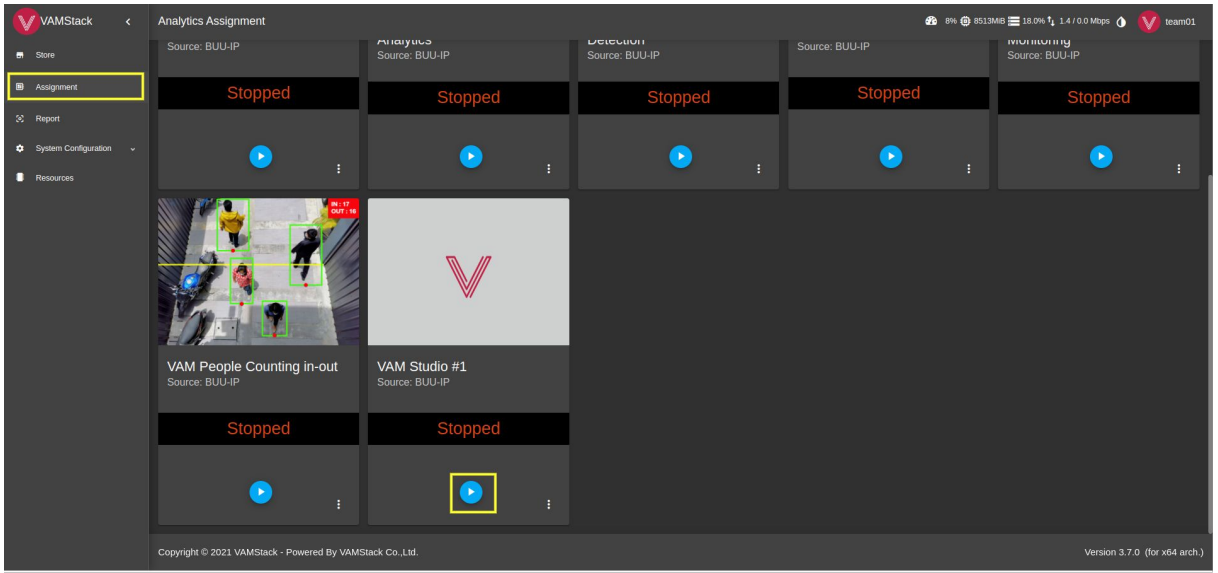
1. เข้าสู่หน้า VAM Platform และให้คลิก “Store” เพื่อค้นหาการ์ด “VAM STUDIO#X”



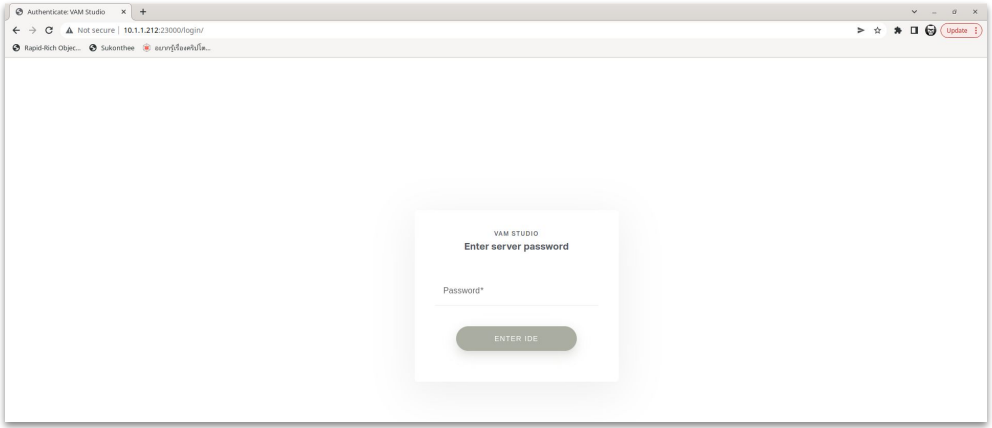
2. คลิกปุ่ม “Apply to Camera” ของการ์ด “VAM STUDIO#X” เพื่อผูกกล้อง โดยมีเงื่อนไขดังนี้
   2.1. บัญชี team01 ใช้การ์ด “VAM STUDIO#1”
   2.2. บัญชี team02 ใช้การ์ด “VAM STUDIO#2”
   2.3. บัญชี team03 ใช้การ์ด “VAM STUDIO#3”
   2.4. บัญชี team04 ใช้การ์ด “VAM STUDIO#4”
   2.5. บัญชี team05 ใช้การ์ด “VAM STUDIO#5”



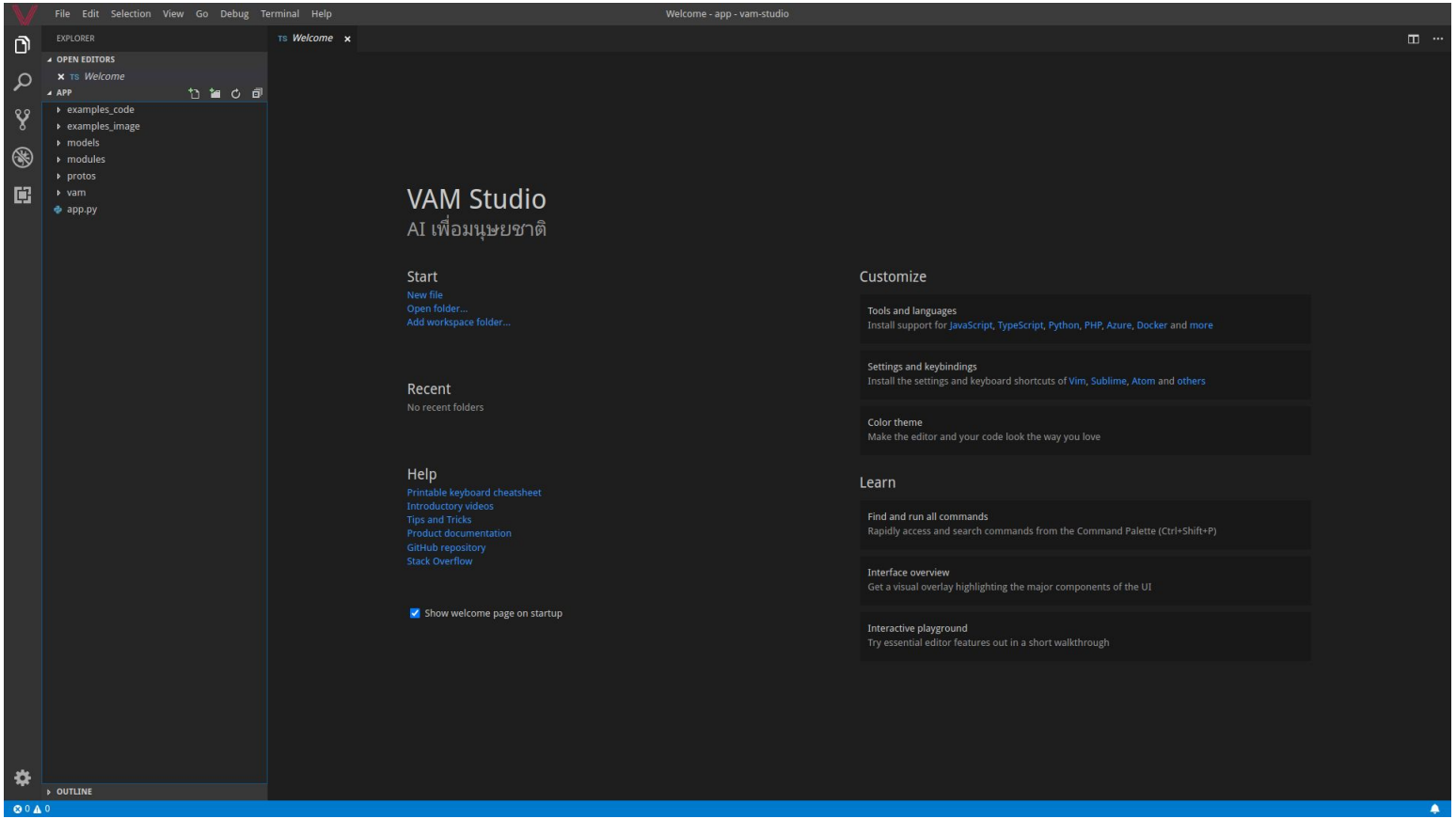
3. หลังผูกกล้องสำเร็จ ให้คลิกเมนู “Assignment” และคลิกที่ปุ่มการทำงานของ “VAM STUDIO#X” เพื่อเปลี่ยนสถานะการทำงาน



4. เมื่อ “VAM STUDIO#X” เริ่มทำงาน ให้เปิดหน้าต่างในเบราเซอร์ใหม่อีก 1 หน้าและทำการพิมพ์ url เพื่อเข้าสู่ระบบ VAM Studio ดังนี้
   - 4.1. การ์ด “VAM STUDIO#1” ใช้ localhost:23000 หรือ 10.1.1.212:23000
   - 4.2. การ์ด “VAM STUDIO#2” ใช้ localhost:23001 หรือ 10.1.1.212:23001
   - 4.3. การ์ด “VAM STUDIO#3” ใช้ localhost:23002 หรือ 10.1.1.212:23002
   - 4.4. การ์ด “VAM STUDIO#4” ใช้ localhost:23003 หรือ 10.1.1.212:23003
   - 4.5. การ์ด “VAM STUDIO#5” ใช้ localhost:23004 หรือ 10.1.1.212:23004
5. เมื่อเข้าสู่หน้า url สำเร็จ จะปรากฏหน้าให้ใส่รหัสผ่านเข้าใช้งาน ซึ่งทุก url จะใช้รหัสเดียวกันคือ P@ssw0rd



VAM Studio


File Edit Selection View Go Debug Terminal Help
Welcome - app - vam-studio
EXPLORER
OPEN EDITORS
Welcome
APP
examples_code
examples_image
models
modules
protos
vam
app.py
OUTLINE
Welcome
VAM Studio
AI เพื่อมนุษยชาติ
Start
New file
Open folder...
Add workspace folder...
Recent
No recent folders
Help
Printable keyboard cheatsheet
Introductory videos
Tips and Tricks
Product documentation
GitHub repository
Stack Overflow
Show welcome page on startup
Customize
Tools and languages
Install support for JavaScript, TypeScript, Python, PHP, Azure, Docker and more
Settings and keybindings
Install the settings and keyboard shortcuts of Vim, Sublime, Atom and others
Color theme
Make the editor and your code look the way you love
Learn
Find and run all commands
Rapidly access and search commands from the Command Palette (Ctrl+Shift+P)
Interface overview
Get a visual overlay highlighting the major components of the UI
Interactive playground
Try essential editor features out in a short walkthrough

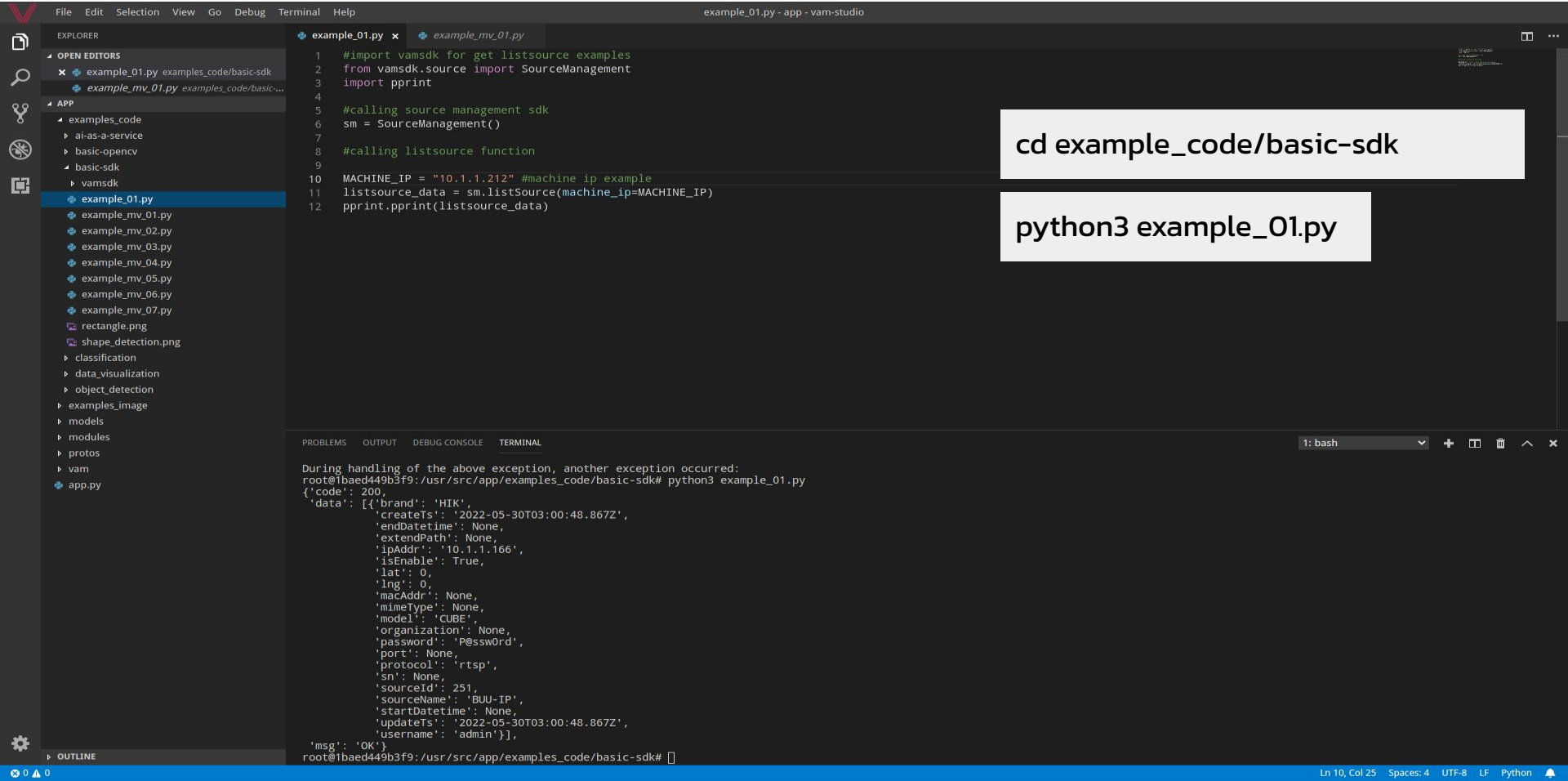
พักซักหน่อย มาเล่นเกมกันเถอะ

```
# import necessary package
import random

# a random number between 0-99
number = random.randint(0,99)

count = 1           # count of guess
max_guess = 7       # count of max guess
message = ''
win = False

while not win:
   guess_num = int(input('Please input your guess :'))
   if guess_num > number:
       message = 'Too big!'
   elif guess_num < number:
       message = 'Too small!'
   elif guess_num == number:
       message = '\nYou win!'
       win = True
   print(message)

   if not win and count == max_guess:
       print('\nYou Lose!')
       print('answer is ',number)

       break
   count += 1
```

VAM SDK

1. ฟังก์ชันสำหรับการรับข้อมูลของกล้องที่อยู่ใน VAM Platform
    1.1. listSource
    1.2. get_vamsnapshot
    1.3. get_vamroi
    1.4. draw_roi
2. ฟังก์ชันพื้นฐานสำหรับการประมวลผลและวิเคราะห์ข้อมูล
    2.1. people_detection
    2.2. people_counting
3. ฟังก์ชันสำหรับการส่งข้อมูลที่ประมวลผลเข้า VAM Platform
    3.1. set_vamlivefigure
    3.2. set_data2vamMetadata

listSource

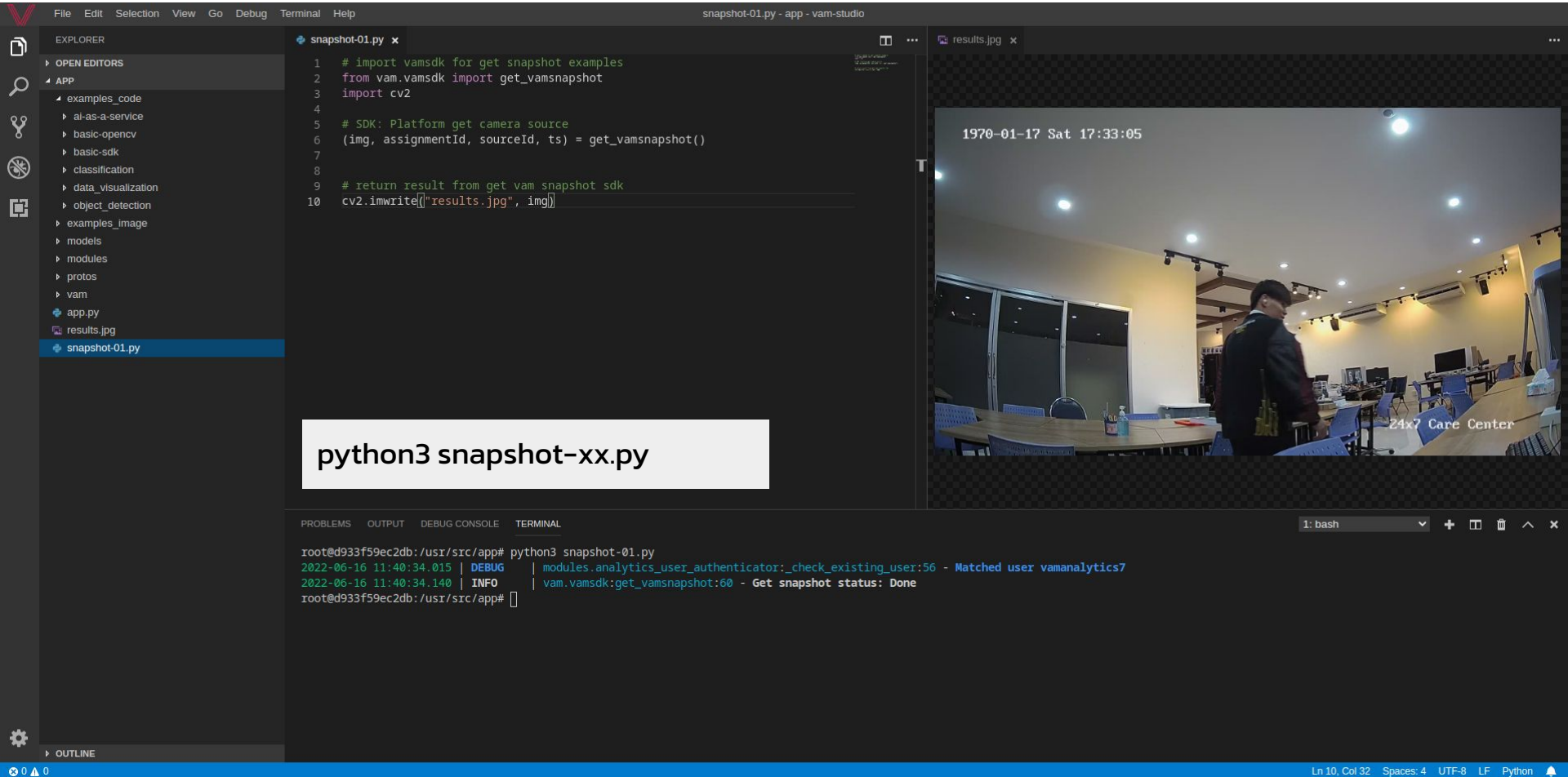

File Edit Selection View Go Debug Terminal Help
example_01.py - app - vam-studio
EXPLORER
OPEN EDITORS
example_01.py examples_code/basic-sdk
example_mv_01.py examples_code/basic-...
APP
examples_code
ai-as-a-service
basic-opencv
basic-sdk
vamsdk
example_01.py
example_mv_01.py
example_mv_02.py
example_mv_03.py
example_mv_04.py
example_mv_05.py
example_mv_06.py
example_mv_07.py
rectangle.png
shape_detection.png
classification
data_visualization
object_detection
examples_image
models
modules
protos
vam
app.py
OUTLINE
example_01.py
example_mv_01.py
1 #import vamsdk for get listsource examples
2 from vamsdk.source import SourceManagement
3 import pprint
4
5 #calling source management sdk
6 sm = SourceManagement()
7
8 #calling listsource function
9
10 MACHINE_IP = "10.1.1.212" #machine ip example
11 listsource_data = sm.listSource(machine_ip=MACHINE_IP)
12 pprint.pprint(listsource_data)
cd example_code/basic-sdk
python3 example_01.py
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL
1: bash
During handling of the above exception, another exception occurred:
root@1baed449b3f9:/usr/src/app/examples_code/basic-sdk# python3 example_01.py
{'code': 200,
'data': [{'brand': 'HIK',
'createTs': '2022-05-30T03:00:48.867Z',
'endDatetime': None,
'extendPath': None,
'ipAddr': '10.1.1.166',
'isEnable': True,
'lat': 0,
'lng': 0,
'macAddr': None,
'mimeType': None,
'model': 'CUBE',
'organization': None,
'password': 'P@ssw0rd',
'port': None,
'protocol': 'rtsp',
'sn': None,
'sourceId': 251,
'sourceName': 'BUU-IP',
'startDatetime': None,
'updateTs': '2022-05-30T03:00:48.867Z',
'username': 'admin'}],
'msg': 'OK'}
root@1baed449b3f9:/usr/src/app/examples_code/basic-sdk#
Ln 10, Col 25 Spaces: 4 UTF-8 LF Python

VAM SDK - get_vamsnapshot

1. สร้างไฟล์แล้วตั้งชื่อเป็น snapshot-xx.py โดยให้แทน xx ด้วยหมายเลขกลุ่ม เช่น snapshot-01.py และห้ามตั้งชื่อซ้ำกัน
2. คัดลอกชุดโค้ดของโปรแกรมดังต่อไปนี้ลงในไฟล์ที่สร้างไว้

```
# import vamsdk for get snapshot examples
from vam.vamsdk import get_vamsnapshot
import cv2

# SDK: Platform get camera source
(img, assignmentId, sourceId, ts) = get_vamsnapshot()

# return result from get vam snapshot sdk
cv2.imwrite("results.jpg", img)
```

get_vamsnapshot



VAM SDK - get_vamroi & draw_roi

1. ตั้งค่าพื้นที่ที่สนใจบนกล้องที่ผูกอยู่กับการ์ด “VAM STUDIO#X” ในหน้า “Assignment”
2. สร้างไฟล์แล้วตั้งชื่อเป็น drawROI-xx.py โดยให้แทน xx ด้วยหมายเลขกลุ่ม เช่น drawROI-01.py
3. คัดลอกชุดโค้ดของโปรแกรมดังต่อไปนี้ลงในไฟล์ที่สร้างไว้

```
# import vamsdk for get roi examples
from vam.vamsdk import get_vamsnapshot
from vam.vamsdk import get_vamroi
from vam.vamsdk import draw_roi
import cv2
import os

# SDK: Platform get camera source
(img, assignmentId, sourceId, ts) = get_vamsnapshot()

# SDK: Calling roi data from platform
(confRoiStatus, confRoiList) = get_vamroi(assignmentId=assignmentId)
print(confRoiList)

# draw roi sdk
roi_frame = draw_roi(img, confRoiList)
cv2.imwrite("results-ROI.jpg", roi_frame)
```

get_vamroi & draw_roi

File Edit Selection View Go Debug Terminal Help — results-ROI.jpg - app - vam-studio

EXPLORER

- OPEN EDITORS
- APP
  - examples_code
    - ai-as-a-service
    - basic-opencv
    - basic-sdk
    - classification
    - data_visualization
    - object_detection
  - examples_image
  - models
  - modules
  - protos
  - vam
  - app.py
  - drawROI-01.py
  - results-ROI.jpg
  - results.jpg
  - snapshot-01.py

drawROI-01.py

```python
# import vamsdk for get roi examples
from vam.vamsdk import get_vamsnapshot
from vam.vamsdk import get_vamroi
from vam.vamsdk import draw_roi

import cv2
import os

# SDK: Platform get camera source
(img, assignmentId, sourceId, ts) = get_vamsnapshot()

# SDK: Calling roi data from platform
(confRoiStatus, confRoiList) = get_vamroi(assignmentId=assignmentId)
print(confRoiList)

# draw roi sdk
roi_frame = draw_roi(img, confRoiList)
cv2.imwrite("results-ROI.jpg", roi_frame)

```

results-ROI.jpg

python3 drawROI-xx.py

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL — 1: bash

```
root@d933f59ec2db:/usr/src/app# python3 drawROI-01.py
2022-06-16 11:54:10.558 | DEBUG    | modules.analytics_user_authenticator:_check_existing_user:56 - Matched user vamanalytics7
2022-06-16 11:54:10.727 | INFO     | vam.vamsdk:get_vamsnapshot:60 - Get snapshot status: Done
2022-06-16 11:54:10.948 | DEBUG    | modules.analytics_user_authenticator:_check_existing_user:56 - Matched user vamanalytics7
[{'source_id': 251, 'assigment_id': 21, 'coordinate': [[[35, 505], [34, 80], [936, 94], [924, 513]]], 'name': 'Zone: 1', 'width': 970, 'height': 582}]
root@d933f59ec2db:/usr/src/app#
```

OUTLINE

0 ⚠ 0 — Whole Image 1920x1080 339.86KB

VAM SDK - people_detection

1. สร้างไฟล์แล้วตั้งชื่อเป็น people-detection-xx.py โดยให้แทน xx ด้วยหมายเลขกลุ่ม เช่น people-detection-01.py
2. คัดลอกชุดโค้ดของโปรแกรมดังต่อไปนี้ลงในไฟล์ที่สร้างไว้

```
# import vamsdk for set get analytics results examples
from vam.vamsdk import people_detection
import cv2

# select image
IMAGE_PATH = "examples_image/person.png"
image = cv2.imread(IMAGE_PATH)

# calling detection people function
people_bboxs, results_frame_people = people_detection(frame=image, conf=0.6)
print(people_bboxs)

# returning people_bboxs
# [(x1, y1, x2, y2, conf)]

# where (x1, y1) is topleft of boundingbox
# where (x2, y2) is bottomright of boundingbox
# where conf is confident of an object

# imwrite people detection results
cv2.imwrite("test.png", results_frame_people)
```

people_detection

File Edit Selection View Go Debug Terminal Help — detection-01.py - app - vam-studio

EXPLORER

OPEN EDITORS

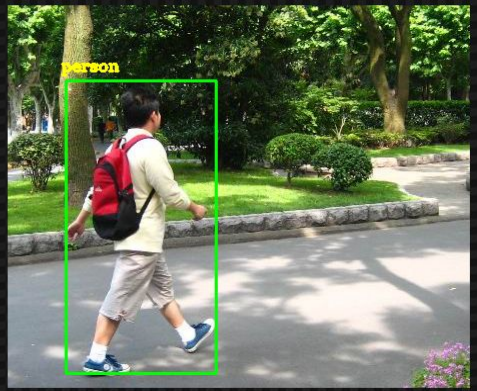
APP
- examples_code
- examples_image
- models
- modules
- protos
- vam
- app.py
- detection-01.py
- drawROI-01.py
- results-ROI.jpg
- results.jpg
- snapshot-01.py
- test.png

OUTLINE

detection-01.py

```python
# import vamsdk for set get analytics results examples
from vam.vamsdk import people_detection

import cv2
# select image
IMAGE_PATH = "/../../../examples_image/person.png"
image = cv2.imread(IMAGE_PATH)

# calling detection people function
# SDK: People detection
people_bboxs, results_frame_people = people_detection(frame=image, conf=0.6)
print(people_bboxs)

# imwrite people detection results
cv2.imwrite("test.png", results_frame_people)
```

test.png



python3 people-detection-xx.py

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

1: bash

```
root@d933f59ec2db:/usr/src/app# python3 detection-01.py
[(58, 76, 208, 372, 0.872821033000946)]
root@d933f59ec2db:/usr/src/app#
```

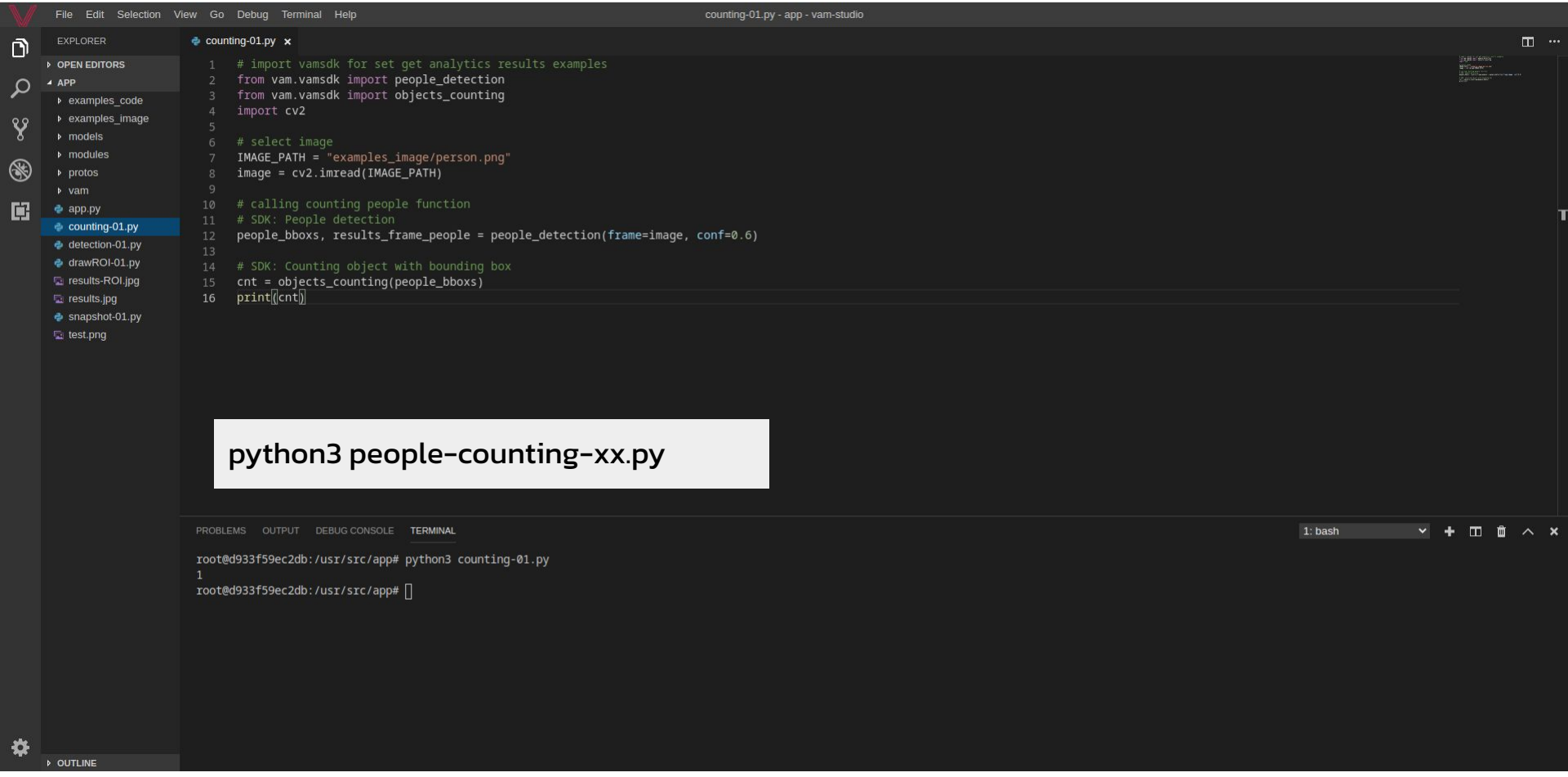
0 0 — Ln 15, Col 46 Spaces: 4 UTF-8 LF Python

VAM SDK - people_counting

1. สร้างไฟล์แล้วตั้งชื่อเป็น people-cnt-xx.py โดยให้แทน xx ด้วยหมายเลขกลุ่ม เช่น people-cnt-01.py
2. คัดลอกชุดโค้ดของโปรแกรมดังต่อไปนี้ลงในไฟล์ที่สร้างไว้

```
# import vamsdk for set get analytics results examples
from vam.vamsdk import people_detection
from vam.vamsdk import objects_counting
import cv2

# select image
IMAGE_PATH = "examples_image/person.png"
image = cv2.imread(IMAGE_PATH)

# SDK: People detection
people_bboxs, results_frame_people = people_detection(frame=image, conf=0.6)

# imwrite people detection results
cv2.imwrite("test.png", results_frame_people)

# SDK: Counting object with bounding box
cnt = objects_counting(people_bboxs)
print(cnt)
```

people_counting

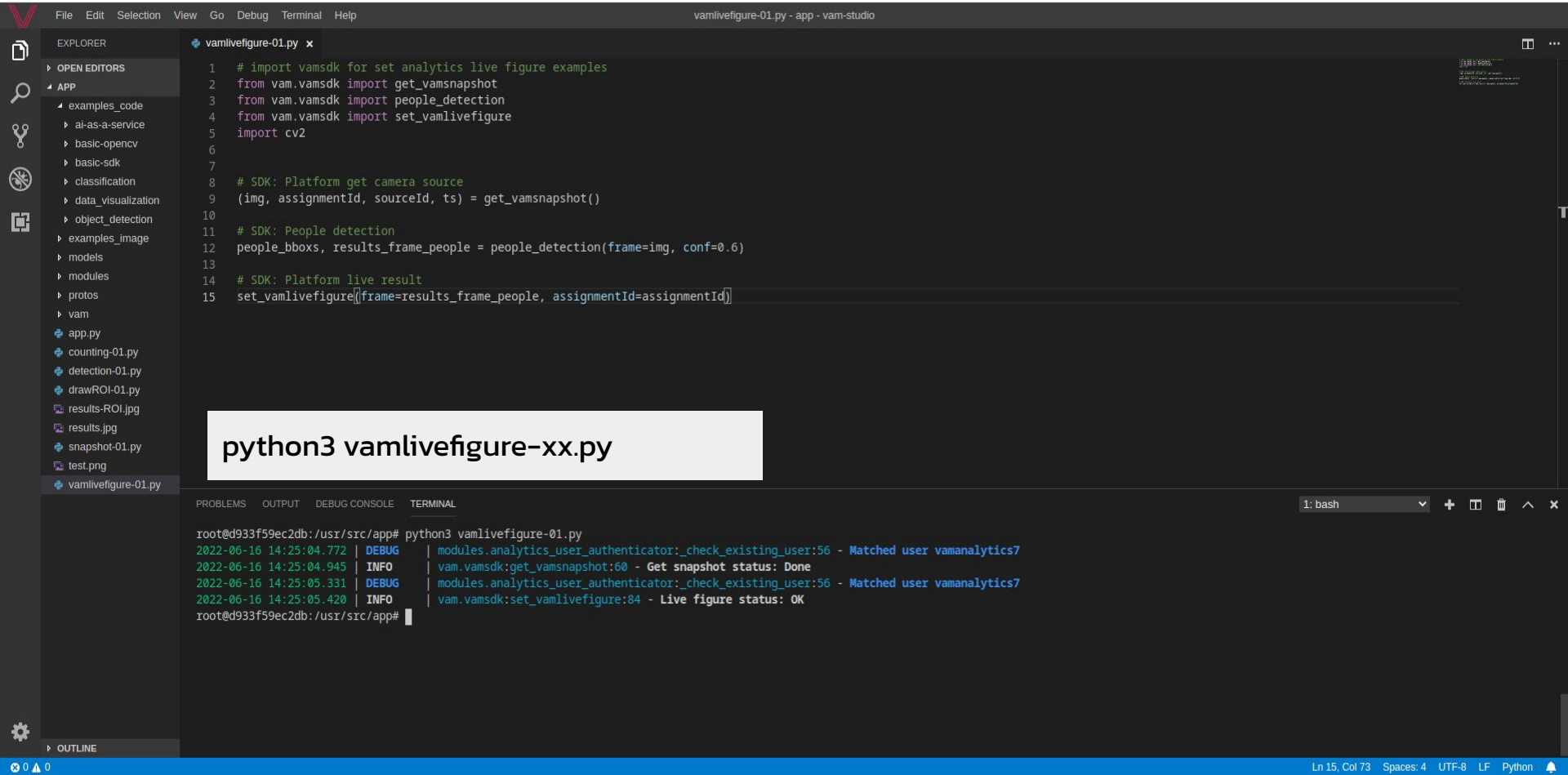


VAM SDK - set_vamlivefigure

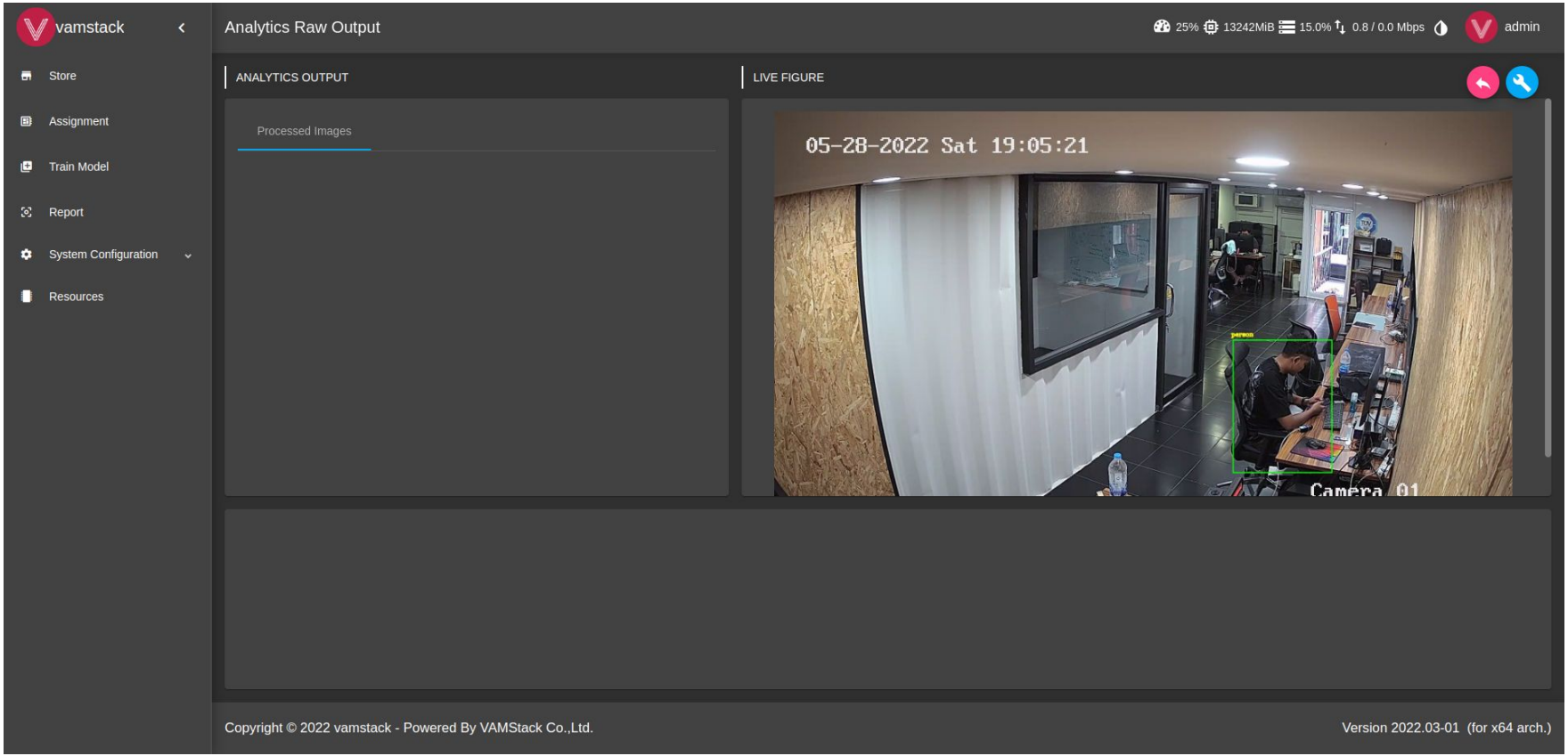
1. สร้างไฟล์แล้วตั้งชื่อเป็น vamlivefigure-xx.py โดยให้แทน xx ด้วยหมายเลขกลุ่ม เช่น vamlivefigure-01.py
2. คัดลอกชุดโค้ดของโปรแกรมดังต่อไปนี้ลงในไฟล์ที่สร้างไว้

```
# import vamsdk for set analytics live figure examples
from vam.vamsdk import get_vamsnapshot
from vam.vamsdk import people_detection
from vam.vamsdk import set_vamlivefigure
import cv2

# SDK: Platform get camera source
(img, assignmentId, sourceId, ts) = get_vamsnapshot()

# SDK: People detection
people_bboxs, results_frame_people = people_detection(frame=img, conf=0.6)

# SDK: Platform live result
set_vamlivefigure(frame=results_frame_people, assignmentId=assignmentId)
```

set_vamlivefigure



VAM SDK - set_vamlivefigure

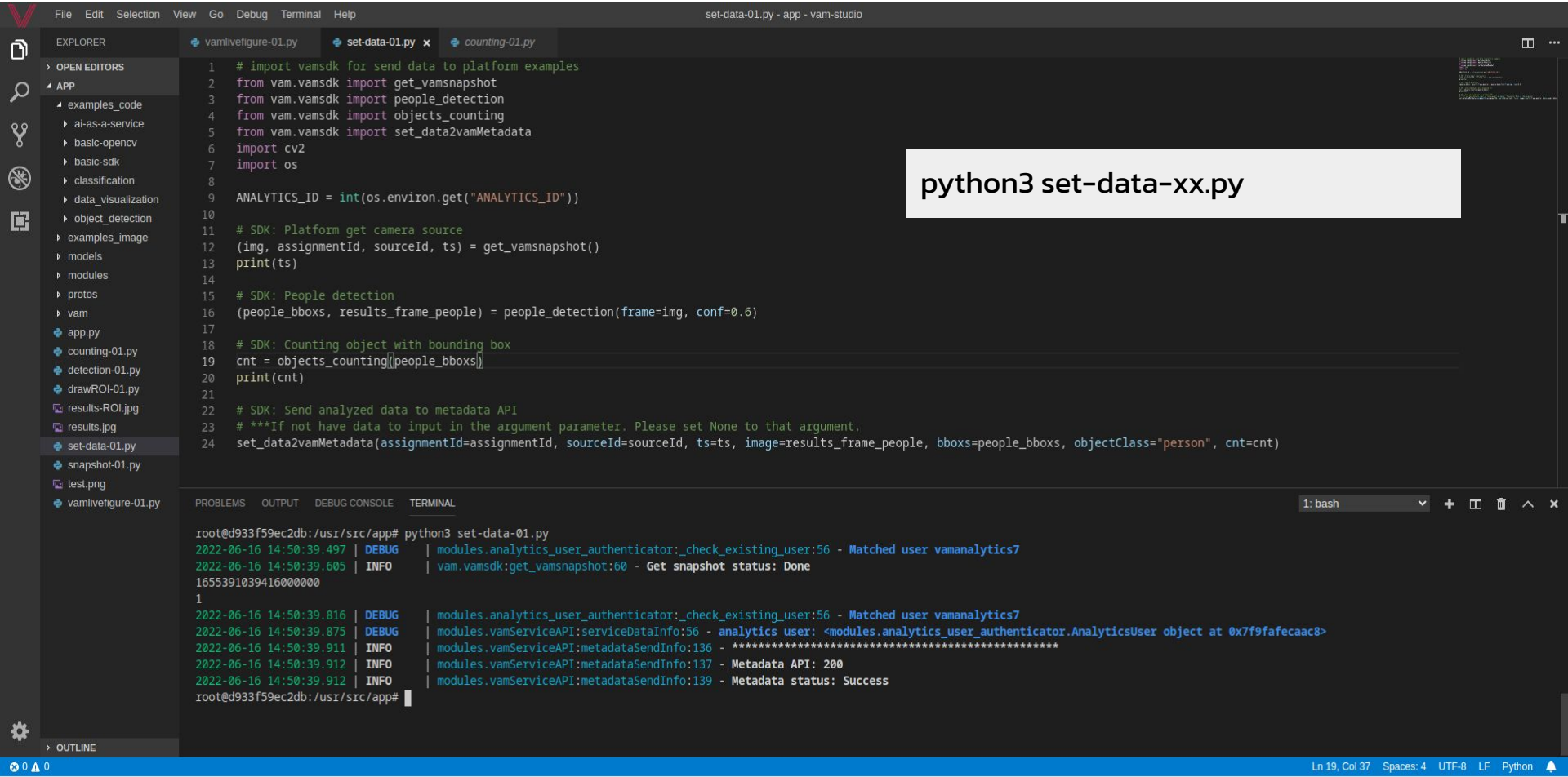


VAM SDK - set_data2vamMetadata

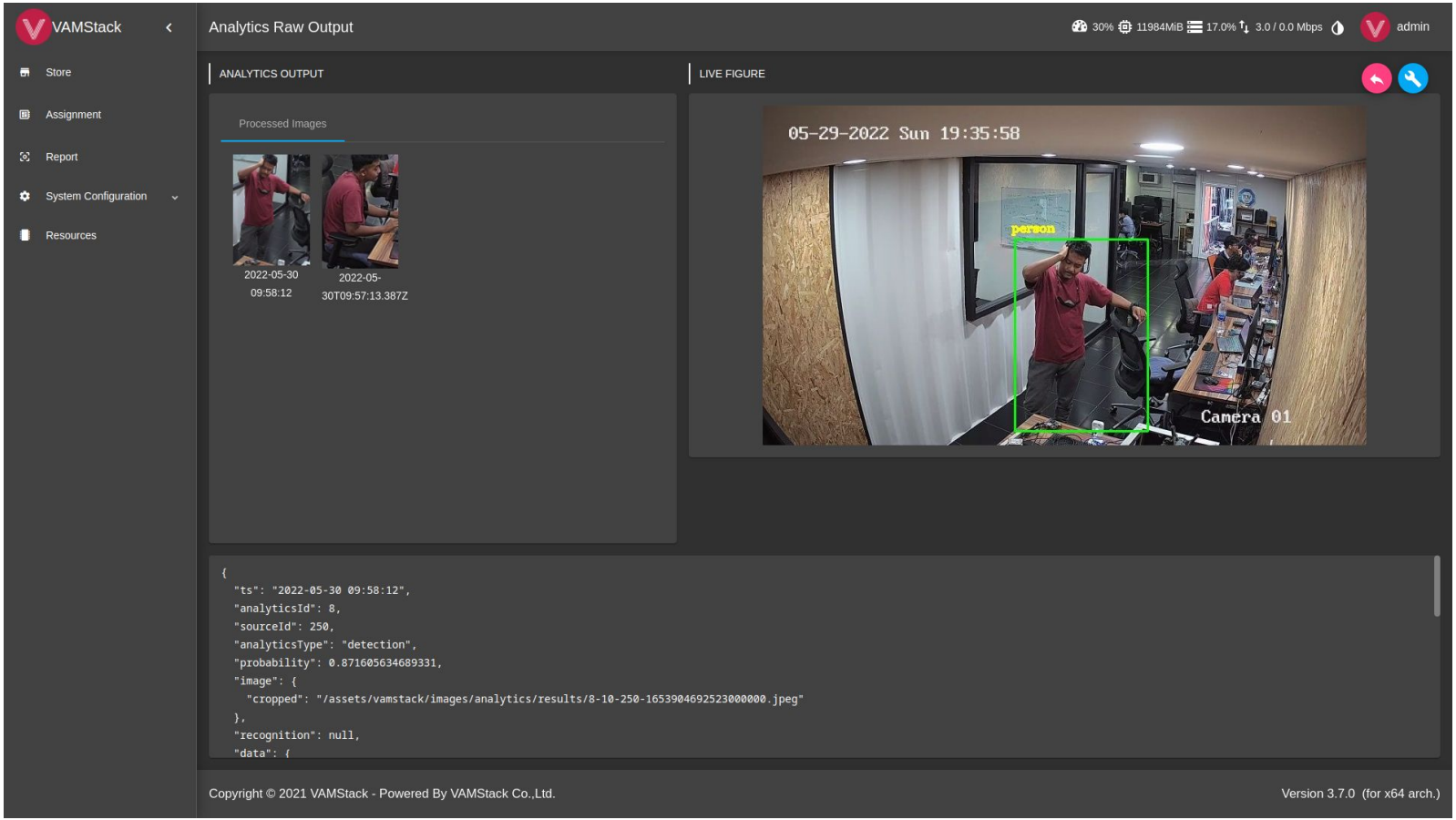
1. สร้างไฟล์แล้วตั้งชื่อเป็น set-data-xx.py โดยให้แทน xx ด้วยหมายเลขกลุ่ม เช่น set-data-01.py
2. คัดลอกชุดโค้ดของโปรแกรมดังต่อไปนี้ลงในไฟล์ที่สร้างไว้

```
# import vamsdk for send data to platform examples
from vam.vamsdk import get_vamsnapshot
from vam.vamsdk import people_detection
from vam.vamsdk import set_data2vamMetadata
import cv2
import os

ANALYTICS_ID = int(os.environ.get("ANALYTICS_ID"))

# SDK: Platform get camera source
(img, assignmentId, sourceId, ts) = get_vamsnapshot()
print(ts)

# SDK: People detection
(people_bboxs, results_frame_people) = people_detection(frame=img, conf=0.6)

# SDK: Counting object with bounding box
cnt = objects_counting(people_bboxs)
print(cnt)

# SDK: Send analyzed data to metadata API
# ***If not have data to input in the argument parameter. Please set None to that argument.
set_data2vamMetadata(assignmentId=assignmentId, sourceId=sourceId, ts=ts, image=results_frame_people, bboxs=people_bboxs,
objectClass="person", cnt=cnt)
```

set_data2vamMetadata



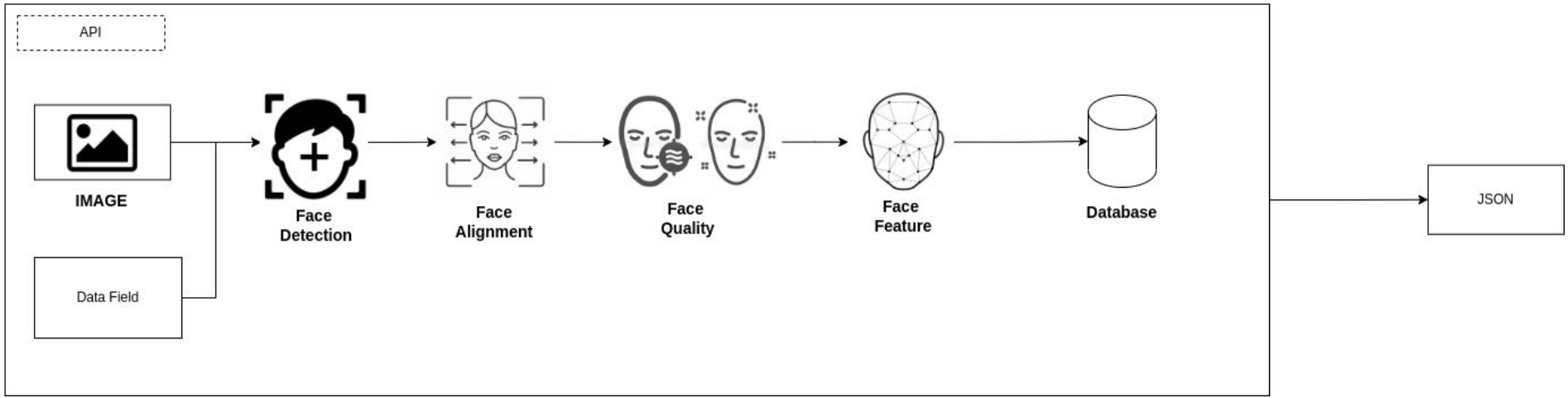
VAM SDK - set_data2vamMetadata



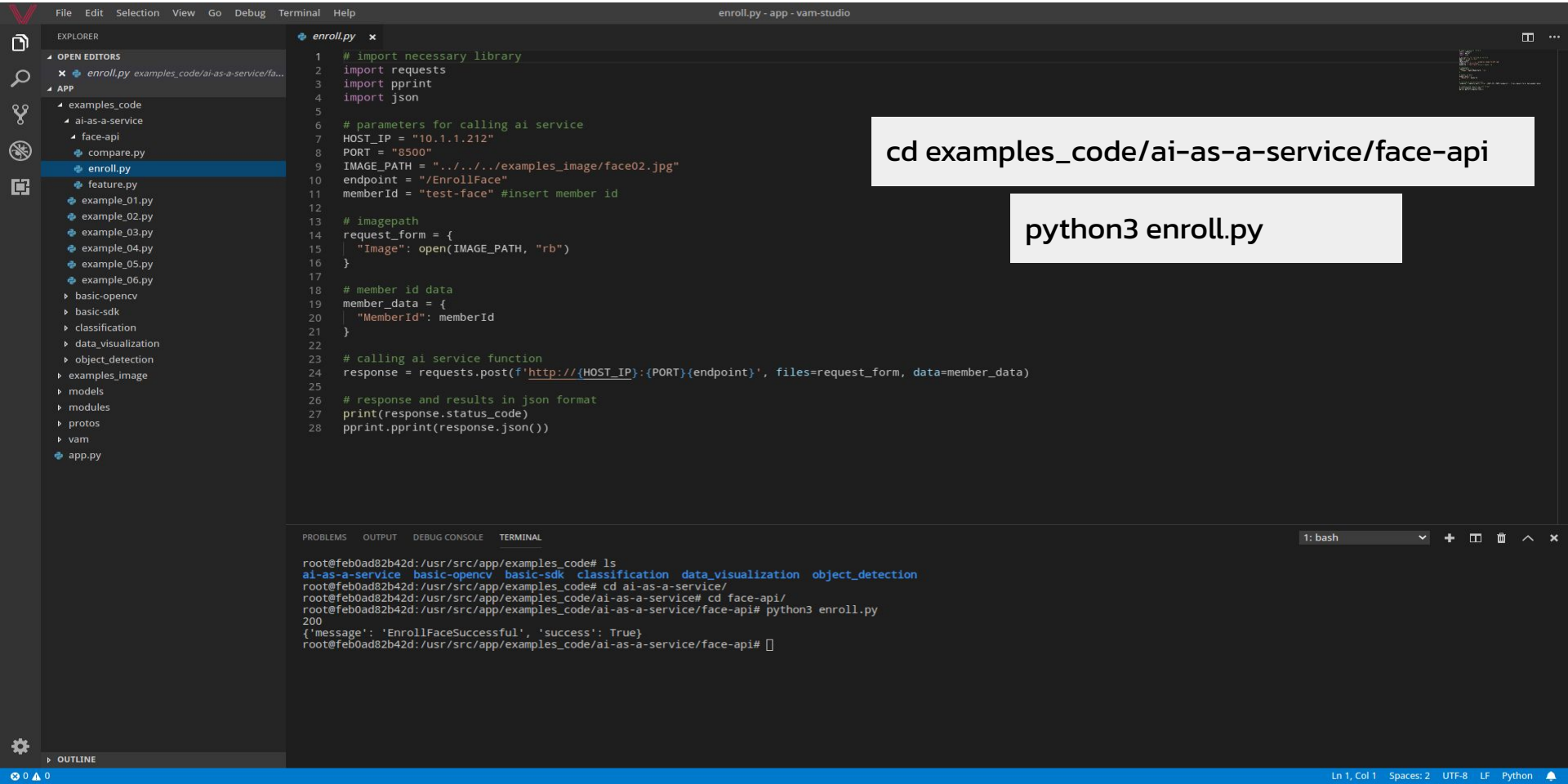
1. Face Registration
    1.1. EnrollFace
2. Face Recognition
    2.1. FaceFeature
3. Face Detection and Quality Verification
    3.1. CompareFace

VAM AIaaS - Face Registration

ENROLL



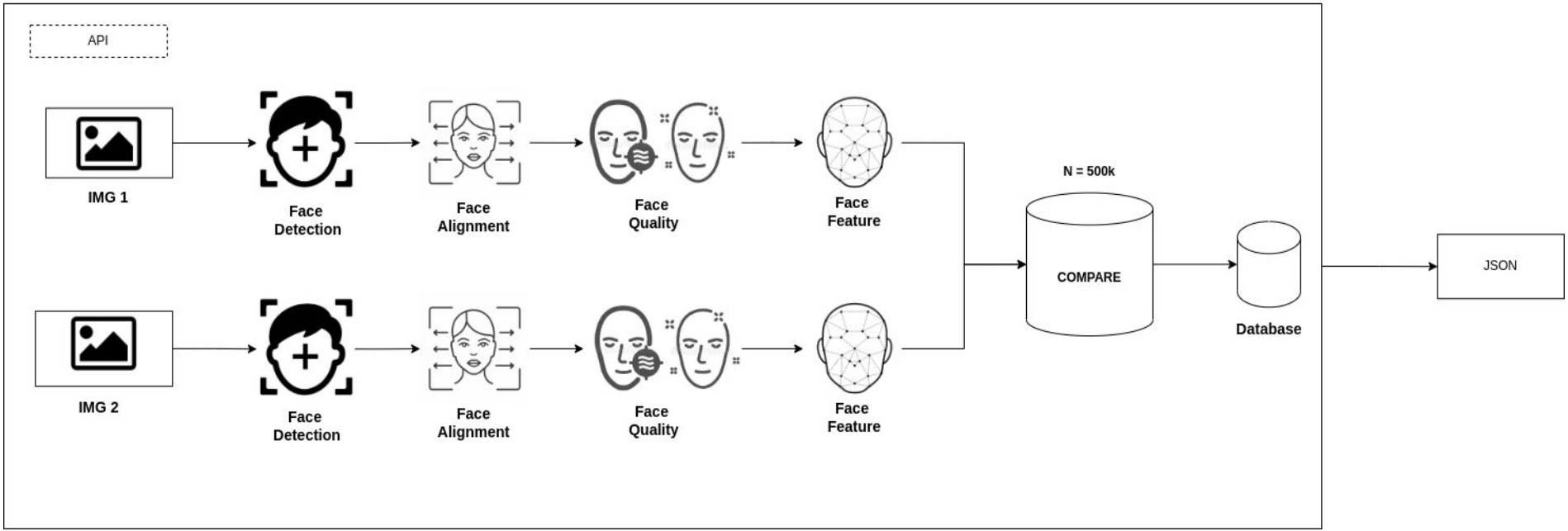
EnrollFace



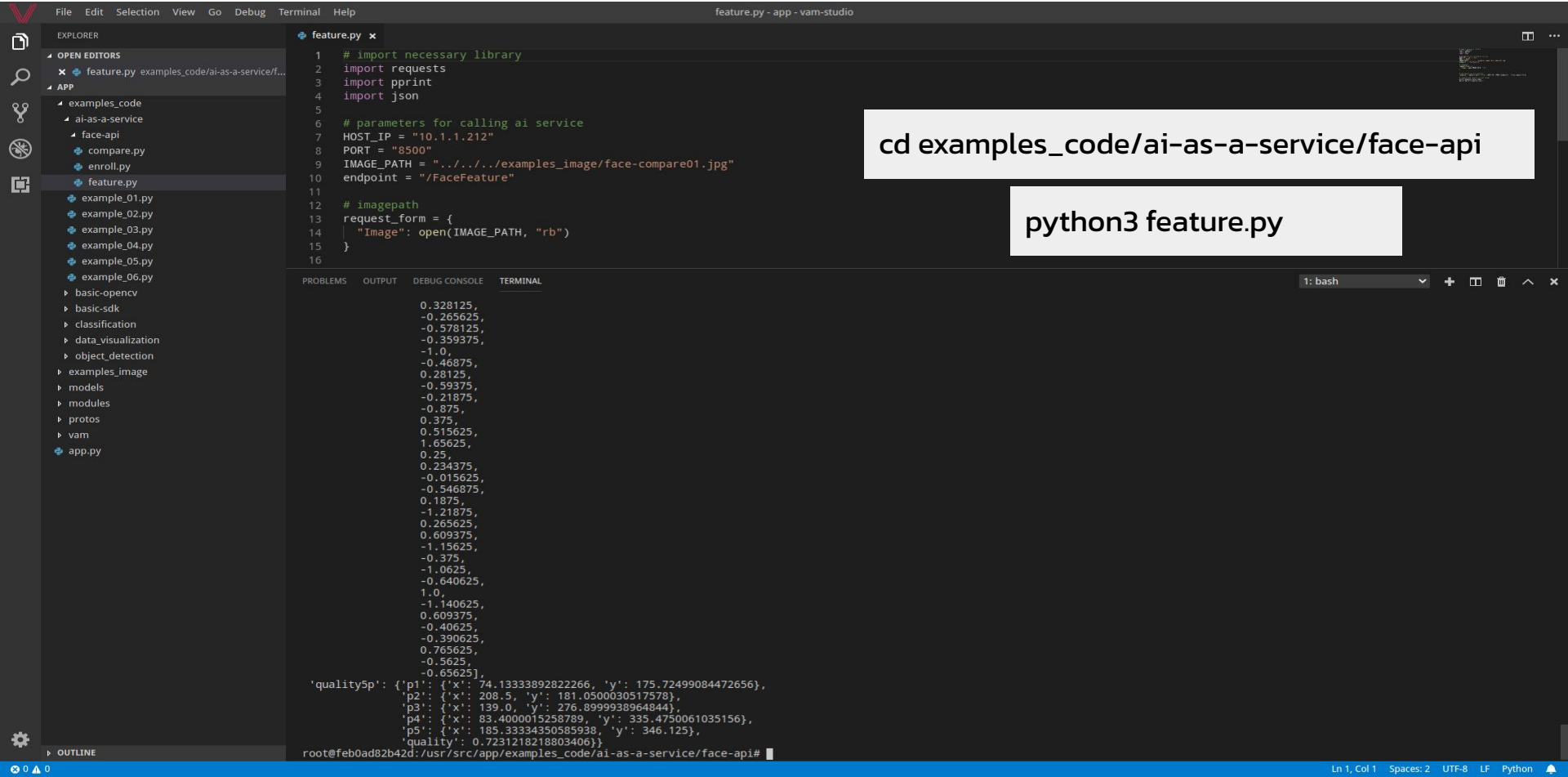
cd examples_code/ai-as-a-service/face-api

python3 enroll.py

RECOGNITION



FaceFeature

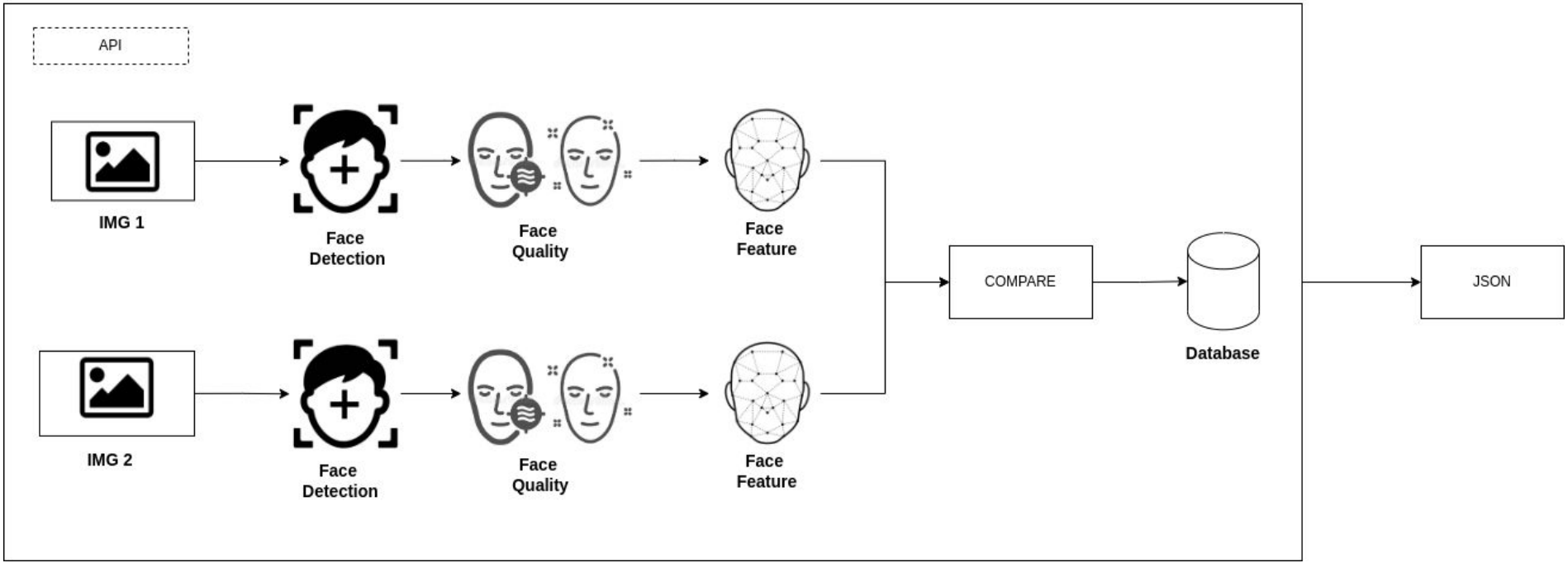



```
cd examples_code/ai-as-a-service/face-api
```

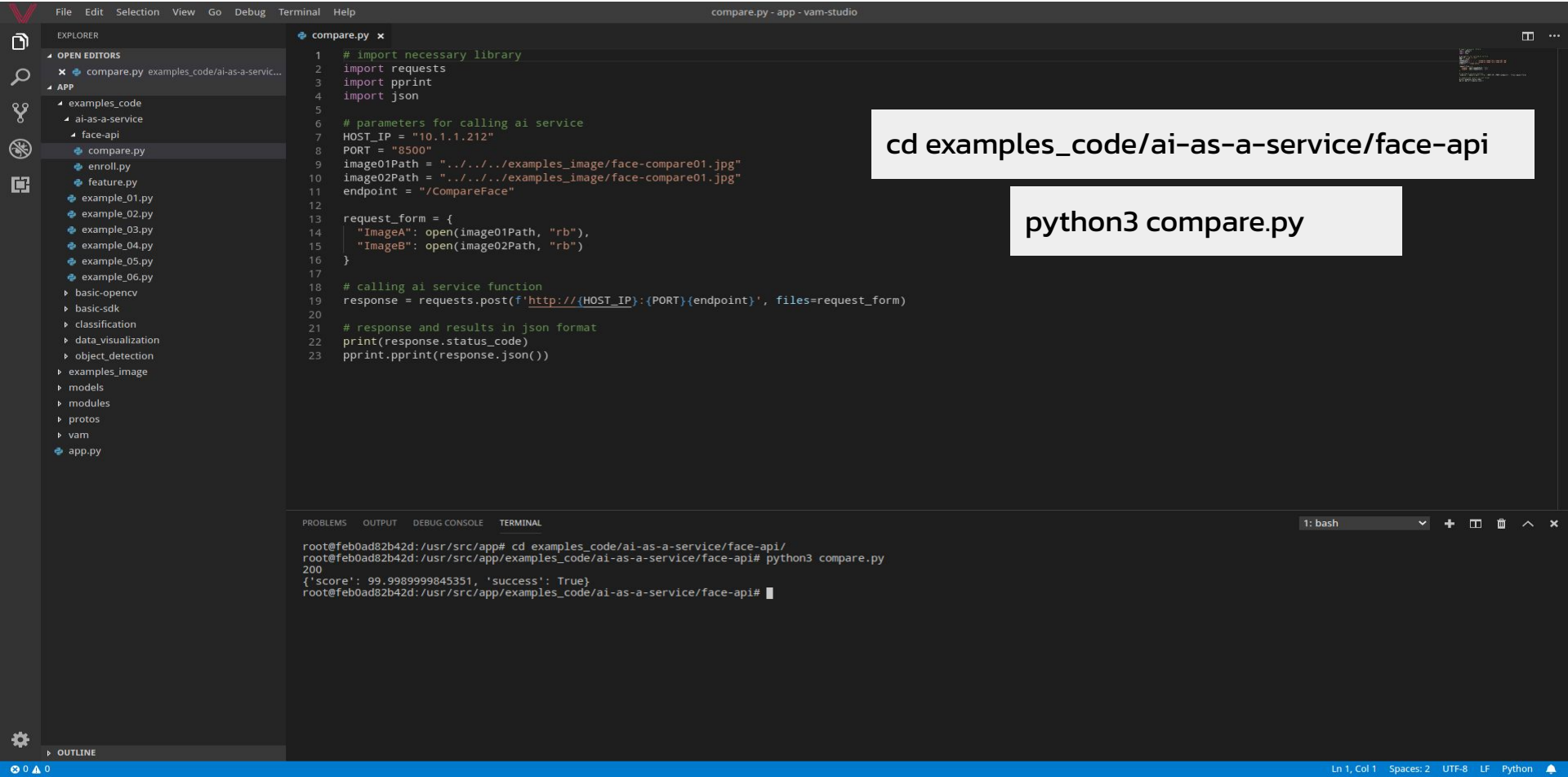
```
python3 feature.py
```

VAM AIaaS - Face Detection and Quality Verification

COMPARE

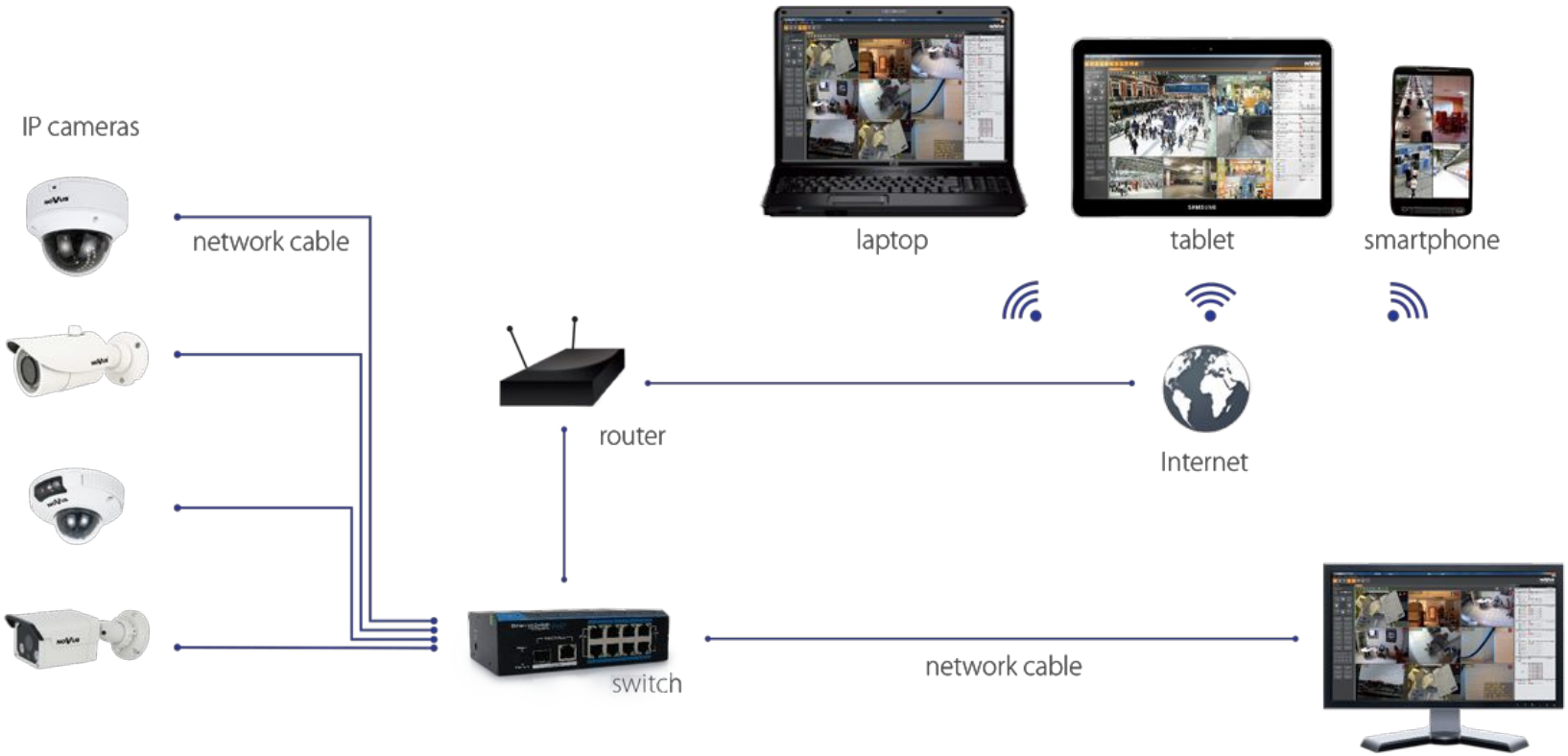


CompareFace



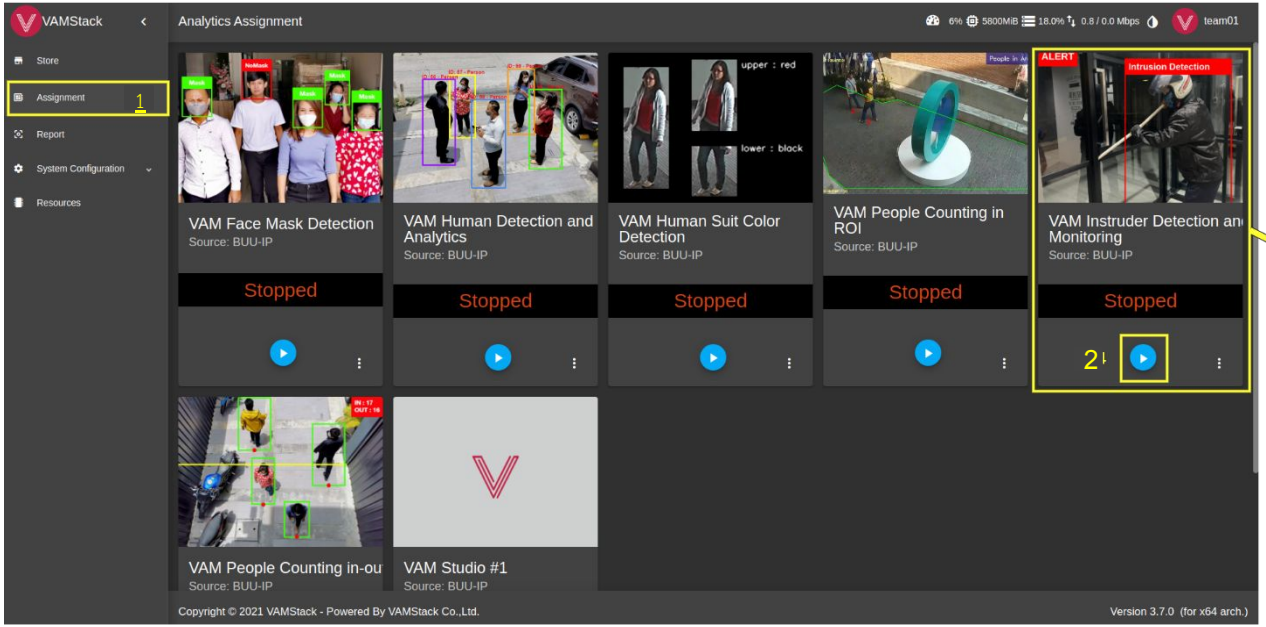
cd examples_code/ai-as-a-service/face-api

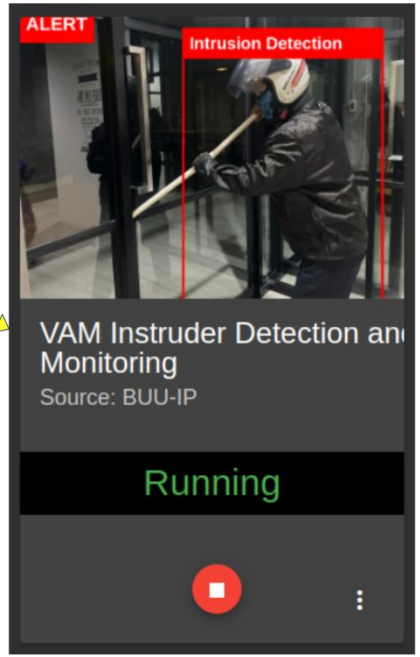
python3 compare.py


IP cameras
network cable
laptop
tablet
smartphone
router
Internet
switch
network cable

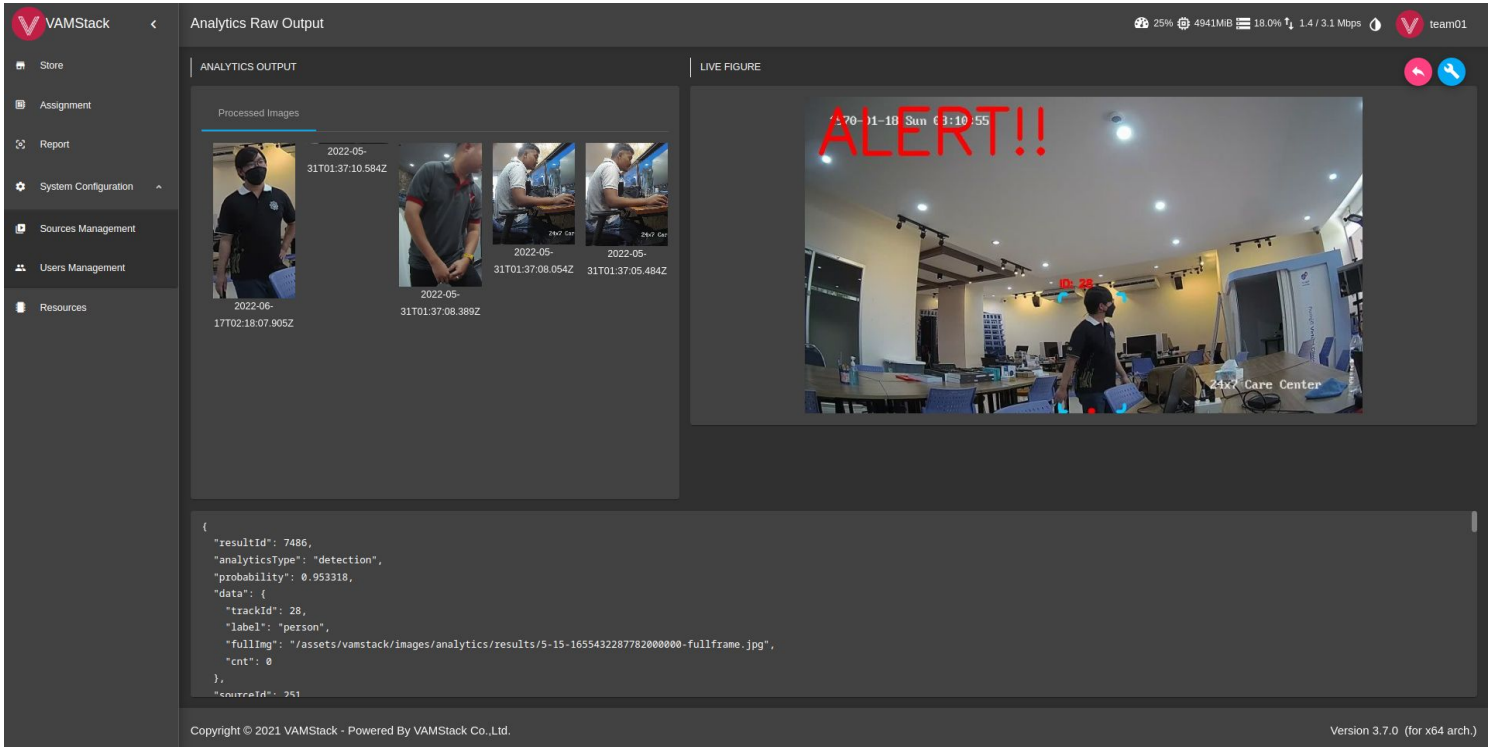
VAM Security & Surveillance - Intrusion Detection

1. คลิก “Assignment” ที่แถบเมนูด้านซ้ายของหน้า VAM Platform
2. คลิกปุ่มเปลี่ยนสถานะของ “VAM Intrusion Detection” ให้เป็น “Running”

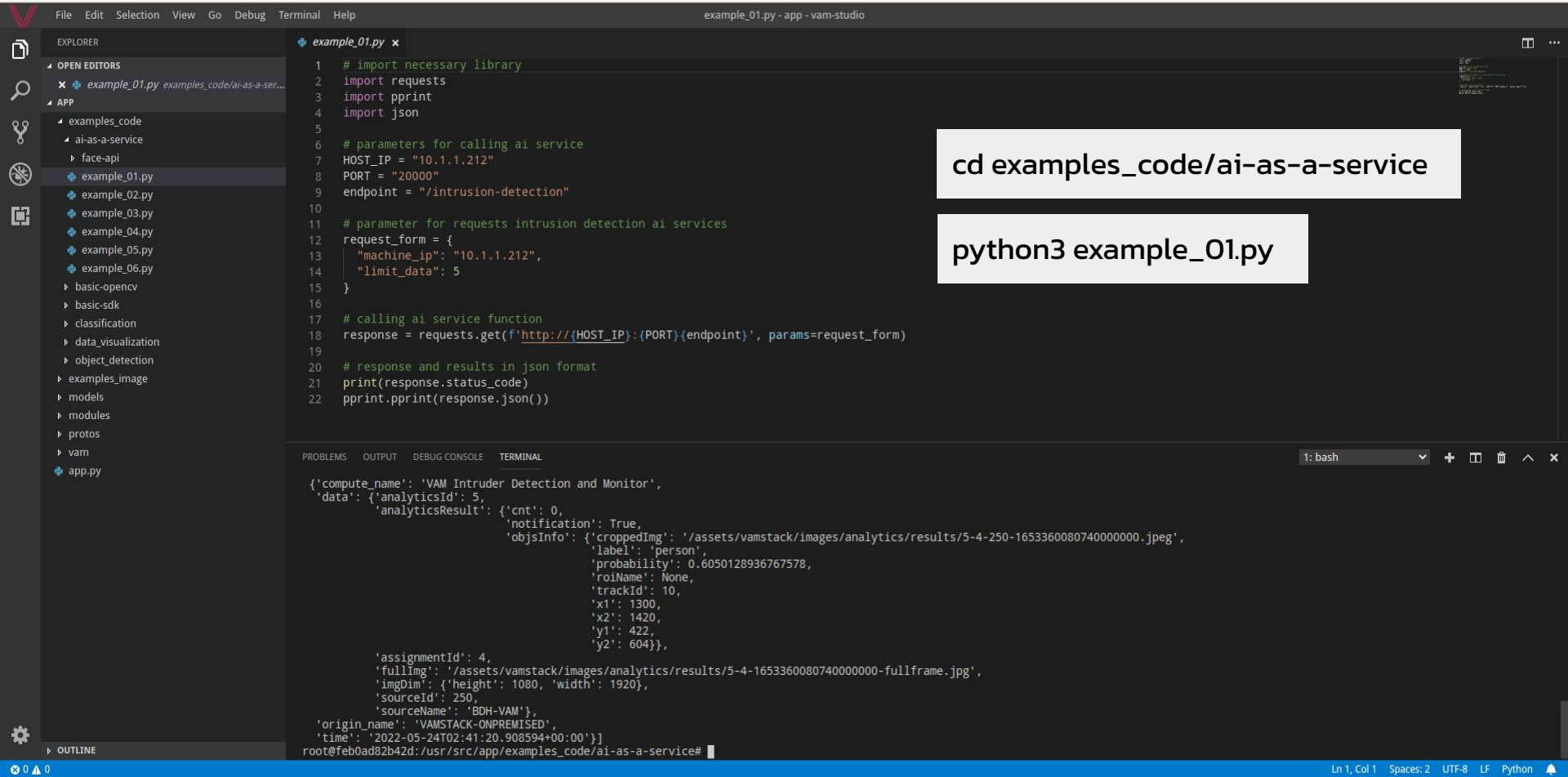




3. คลิกการ์ด “VAM Intrusion Detection” ที่มีสถานะเริ่มทำงาน (Running) เพื่อดูการประมวลผลและวิเคราะห์ข้อมูล



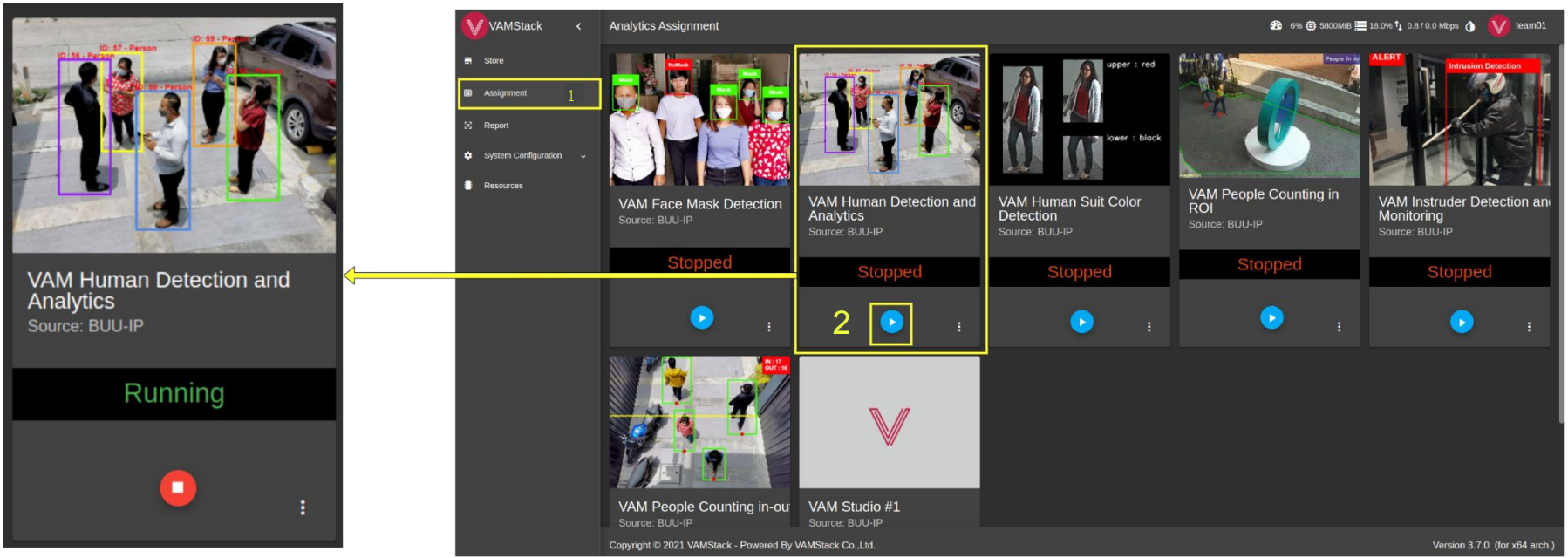
Intrusion Detection



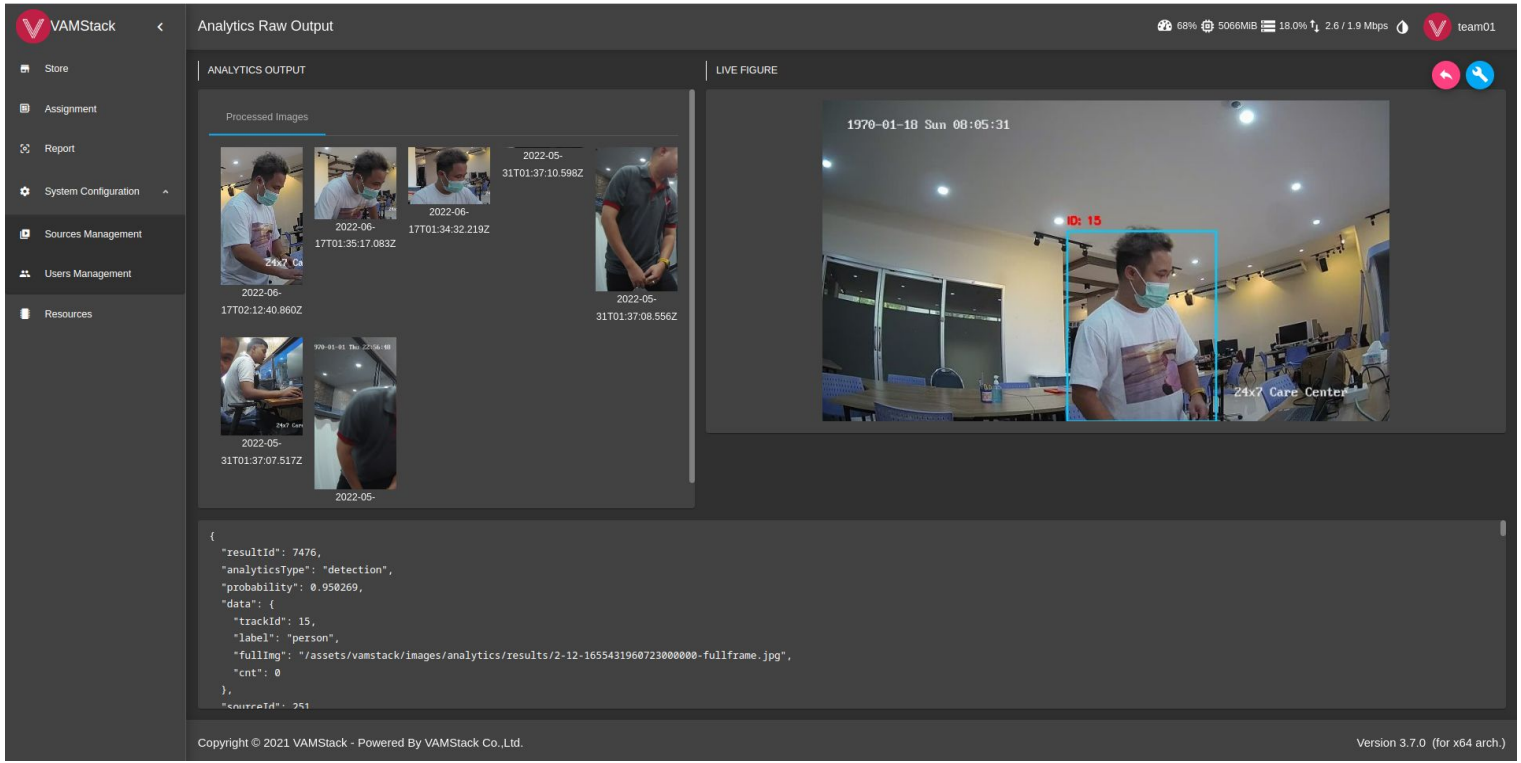
cd examples_code/ai-as-a-service

python3 example_01.py

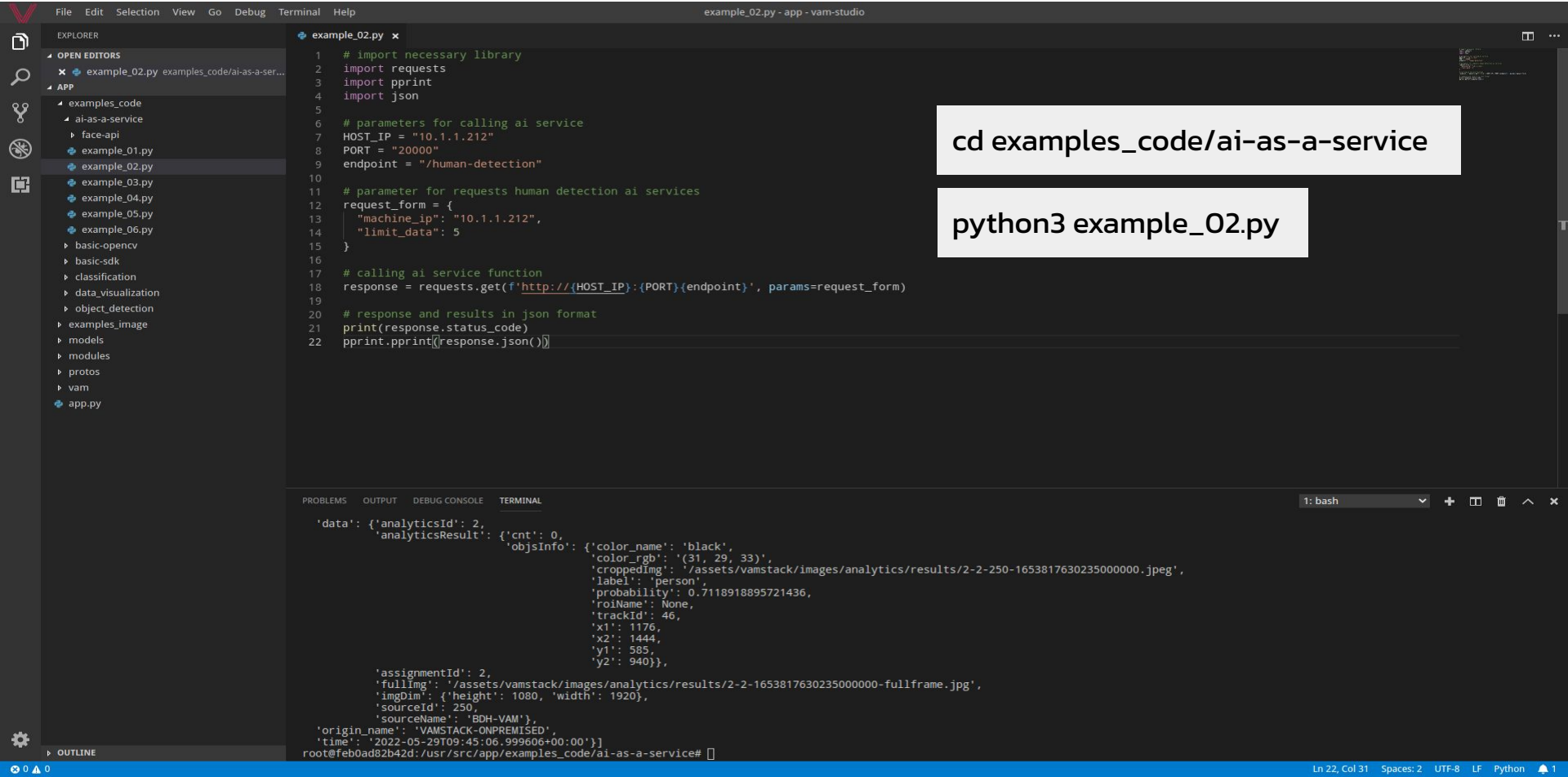
1. คลิก “Assignment” ที่แถบเมนูด้านซ้ายของหน้า VAM Platform
2. คลิกปุ่มเปลี่ยนสถานะของ “VAM Human Detection” ให้เป็น “Running”



3. คลิกที่การ์ด “VAM Human Detection” เพื่อดูการประมวลผลและวิเคราะห์ข้อมูล

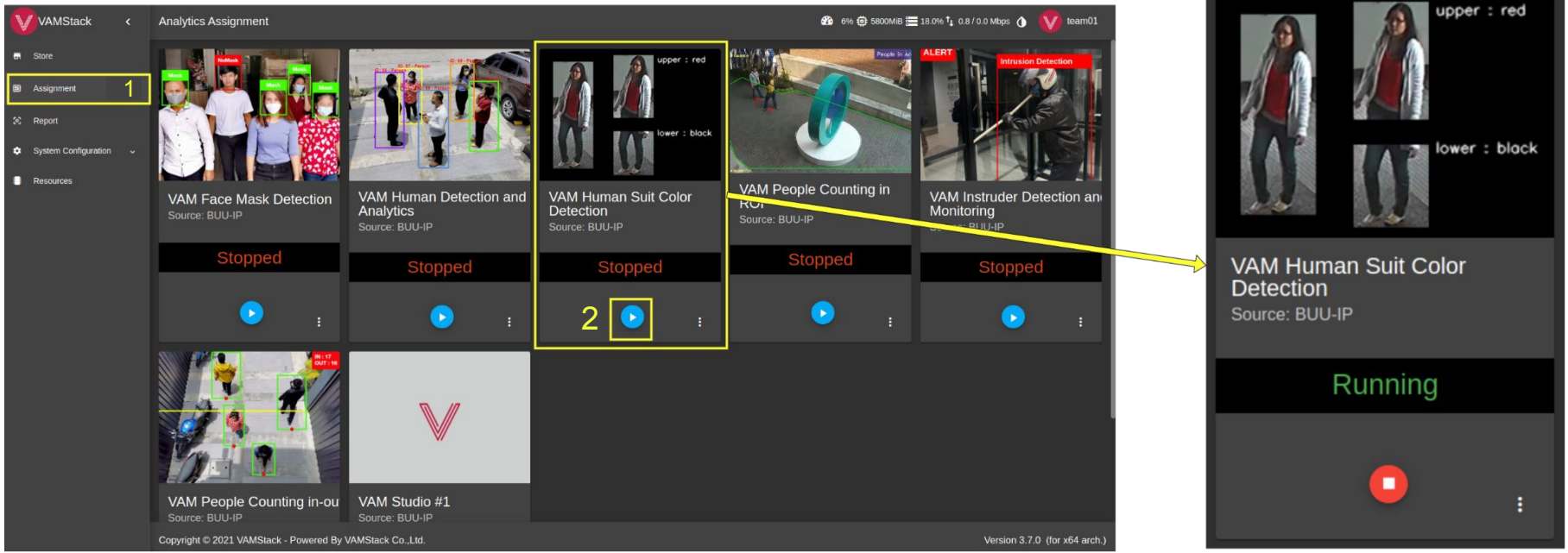


Human Detection

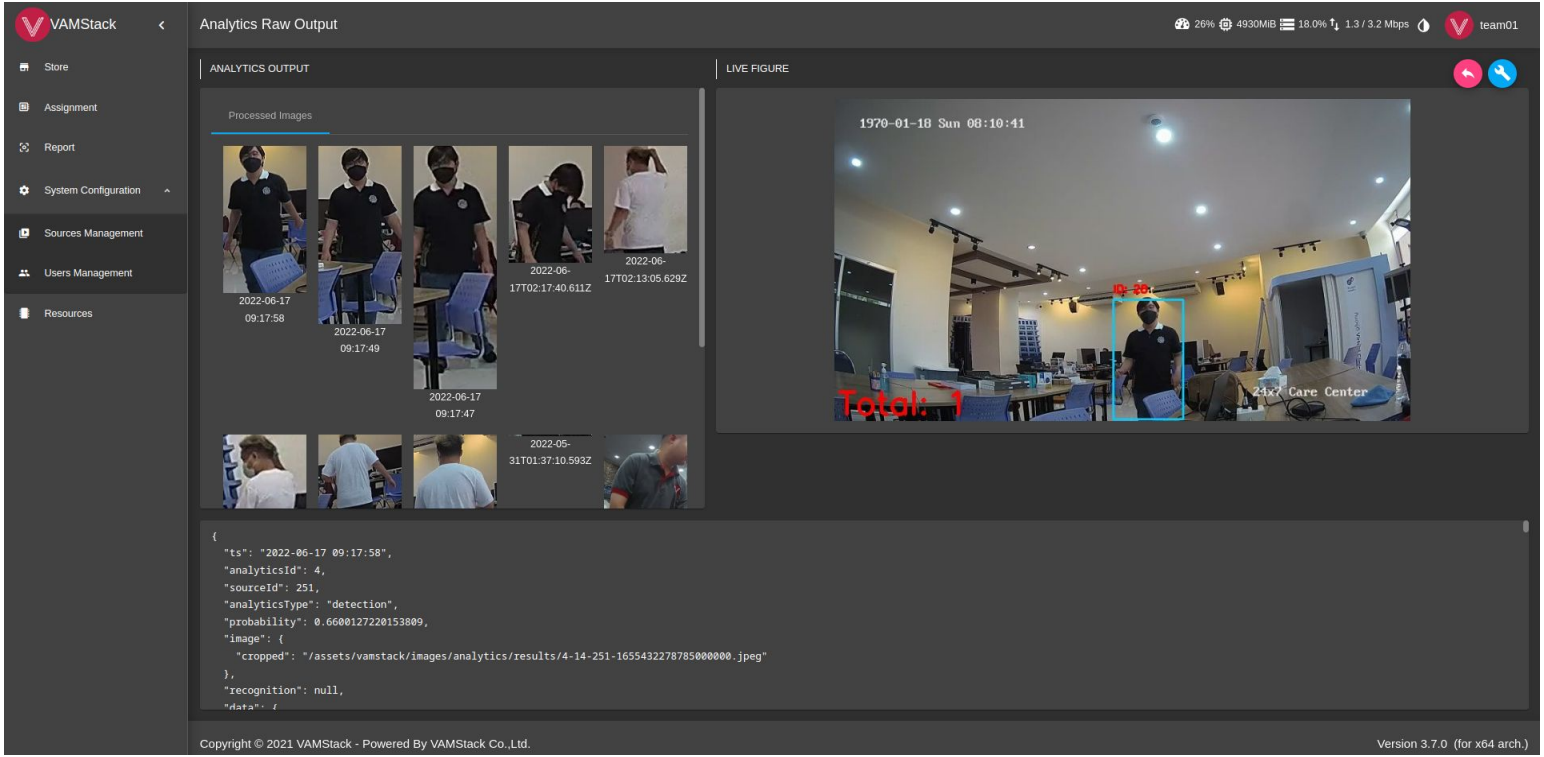


VAM Security & Surveillance - Human Suit Color Detection

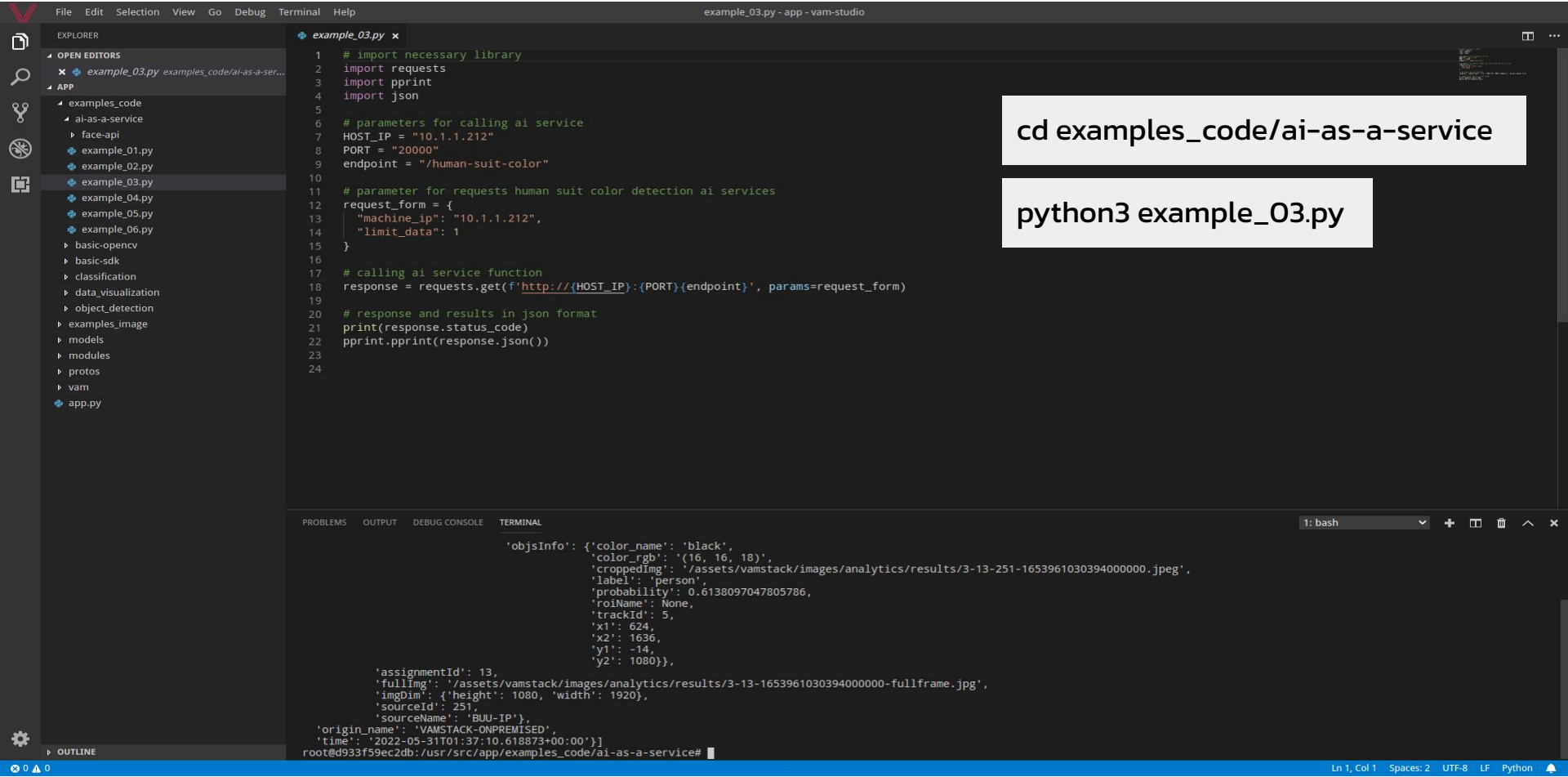
1. คลิก “Assignment” ที่แถบเมนูด้านซ้ายของหน้า VAM Platform
2. คลิกปุ่มเปลี่ยนสถานะของ “VAM Human Suit Color Detection” ให้เป็น “Running”



3. คลิกที่การ์ด “VAM Human Suit Color Detection” เพื่อดูการประมวลผลและวิเคราะห์ข้อมูล



Human Suit Color Detection

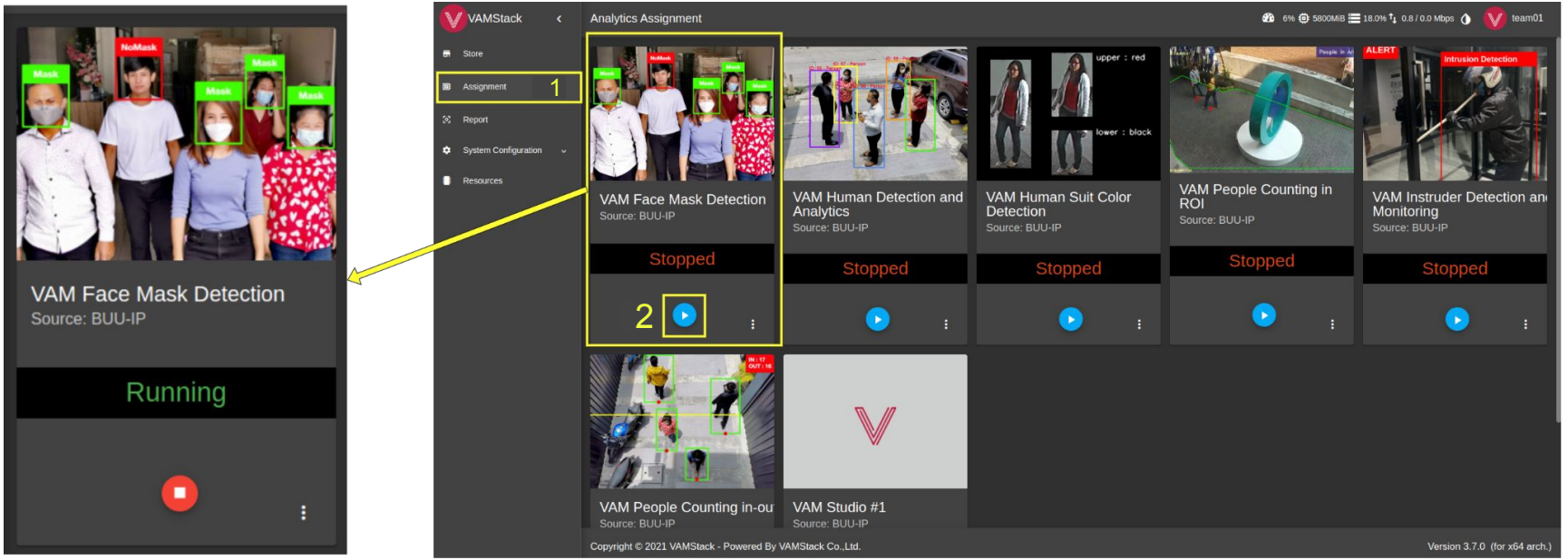



```
cd examples_code/ai-as-a-service
```

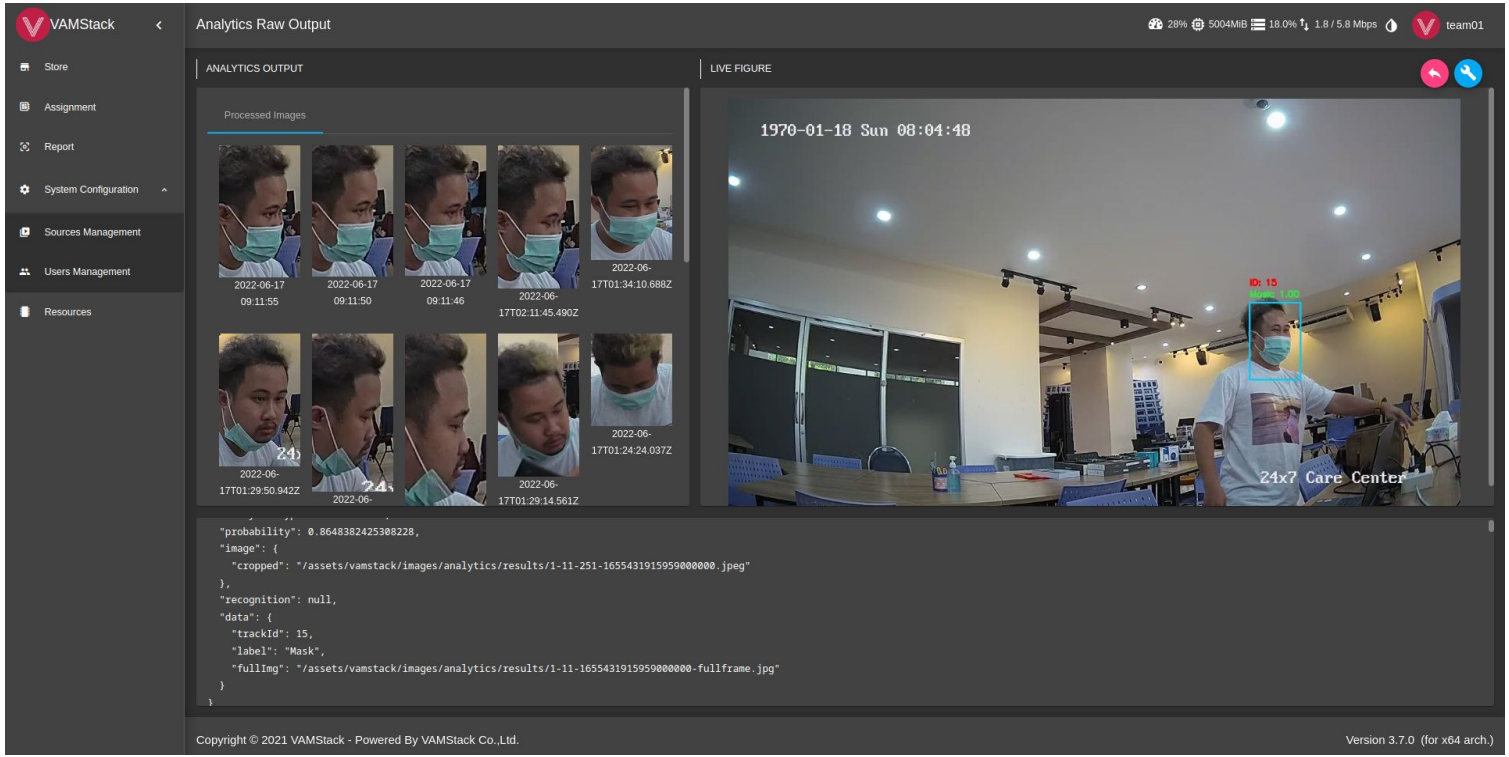
```
python3 example_03.py
```

VAM Security & Surveillance - Face Mask Detection

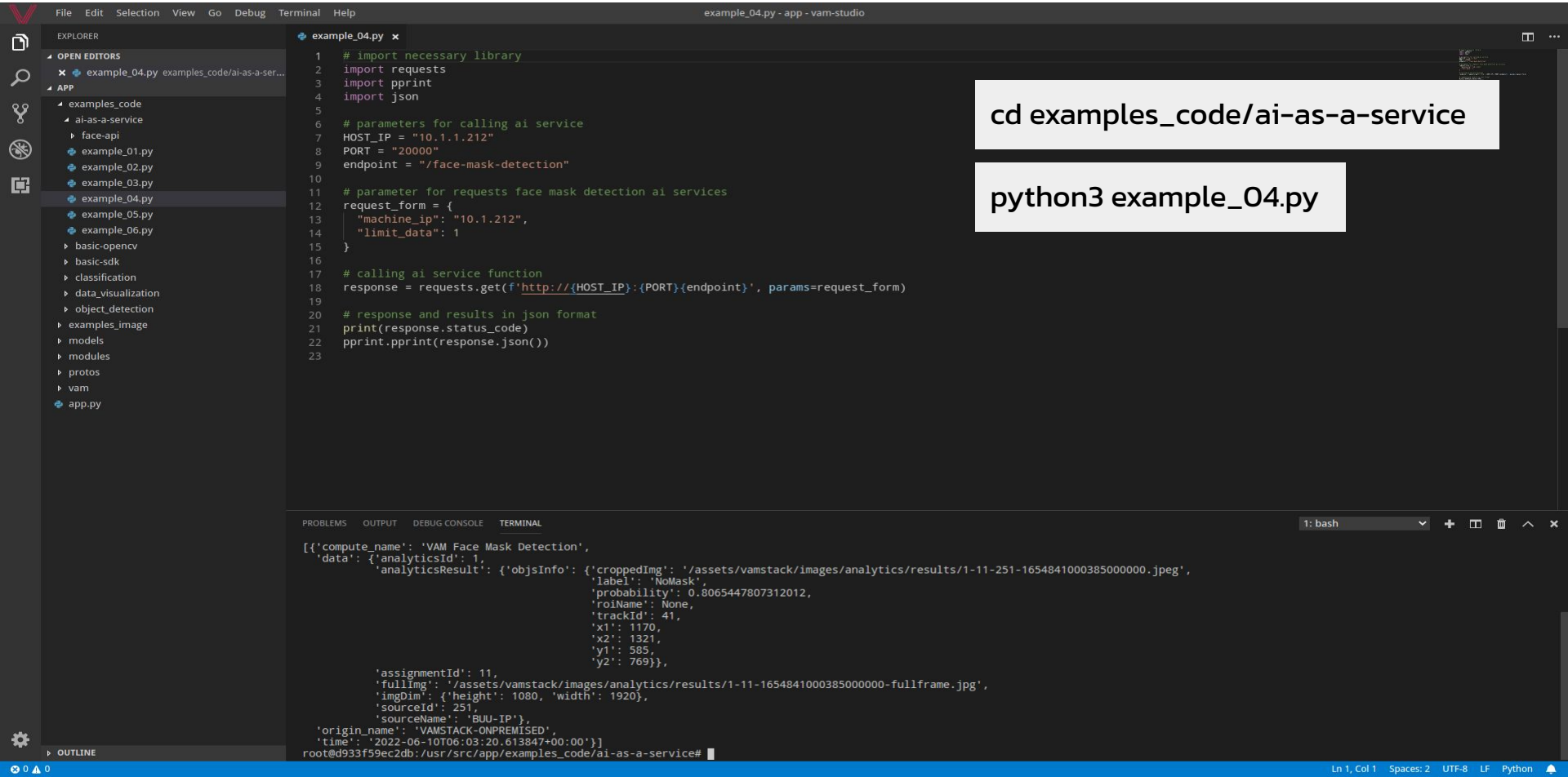
1. คลิก “Assignment” ที่แถบเมนูด้านซ้ายของหน้า VAM Platform
2. คลิกปุ่มเปลี่ยนสถานะของ “VAM Face Mask Detection” ให้เป็น “Running”



3. คลิกที่การ์ด “VAM Face Mask Detection” เพื่อดูการประมวลผลและวิเคราะห์ข้อมูล

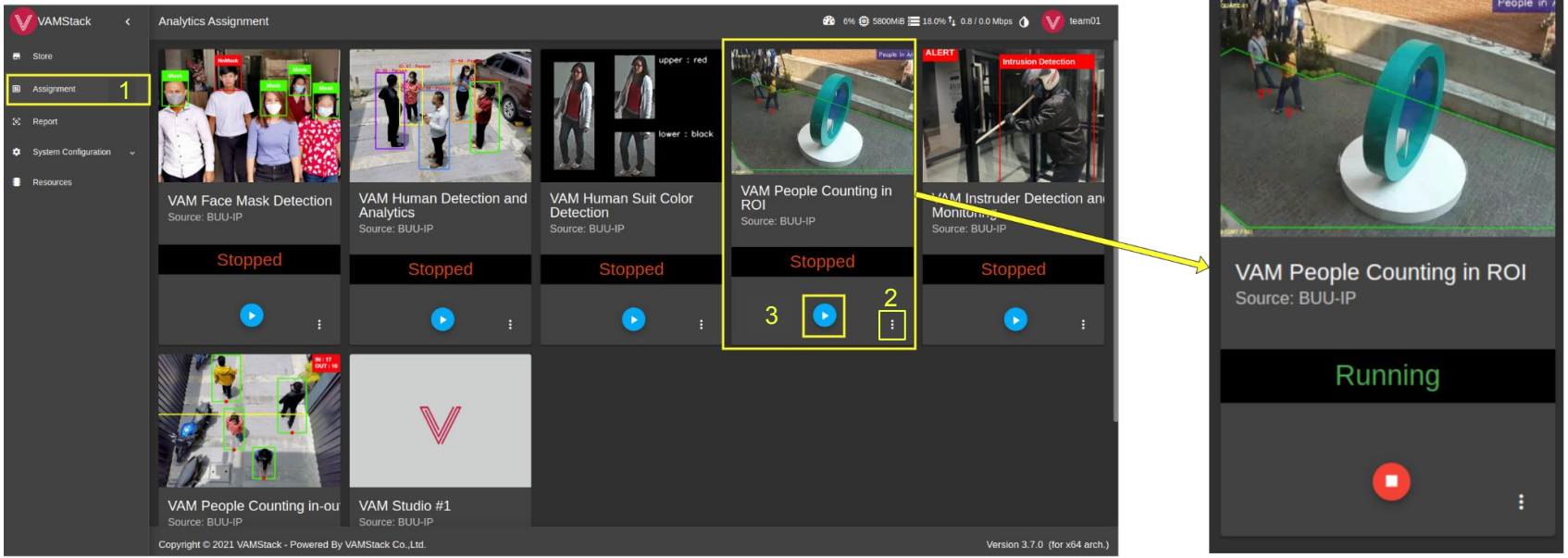


Face Mask Detection

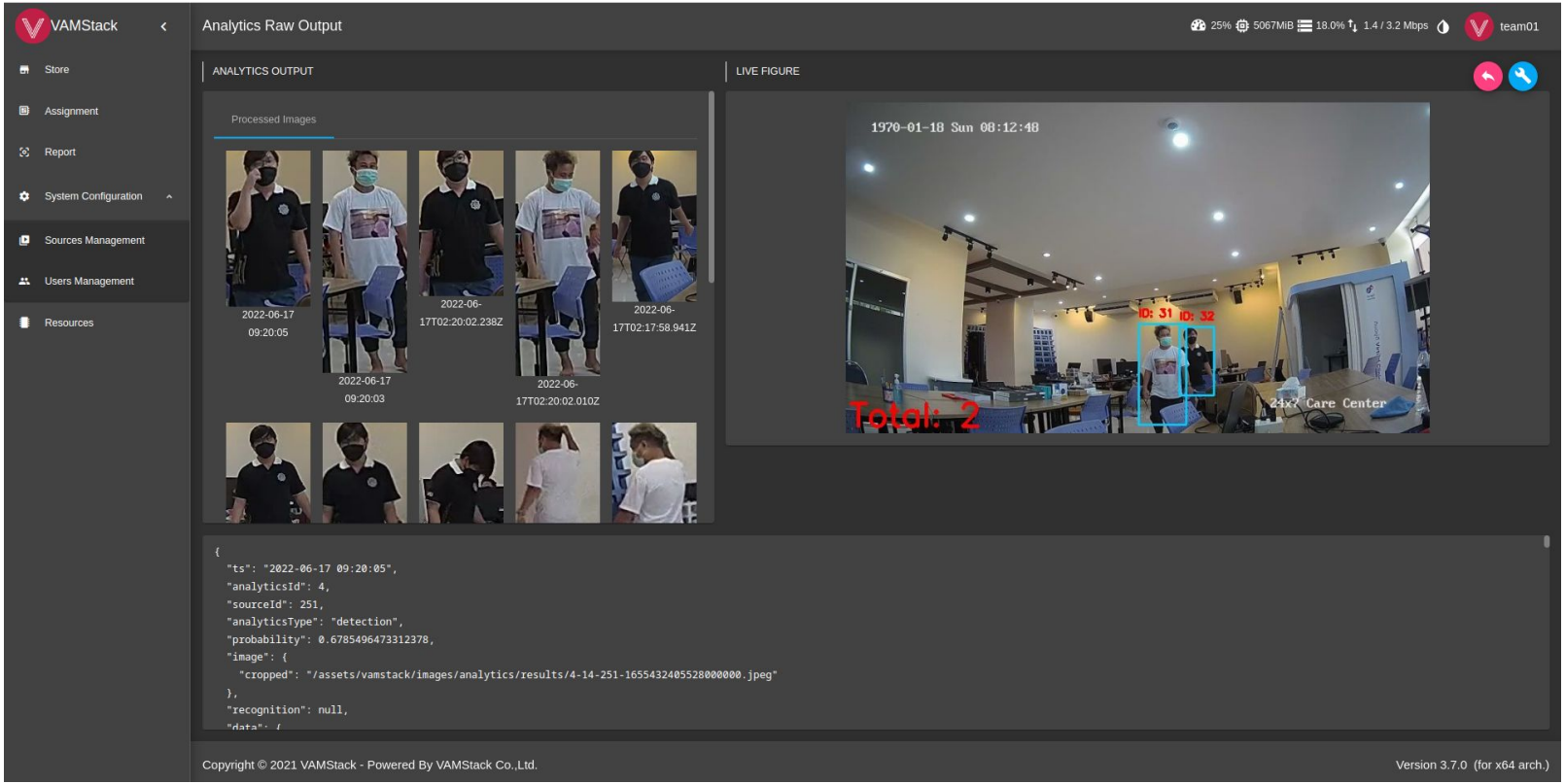


VAM Security & Surveillance - People Counting

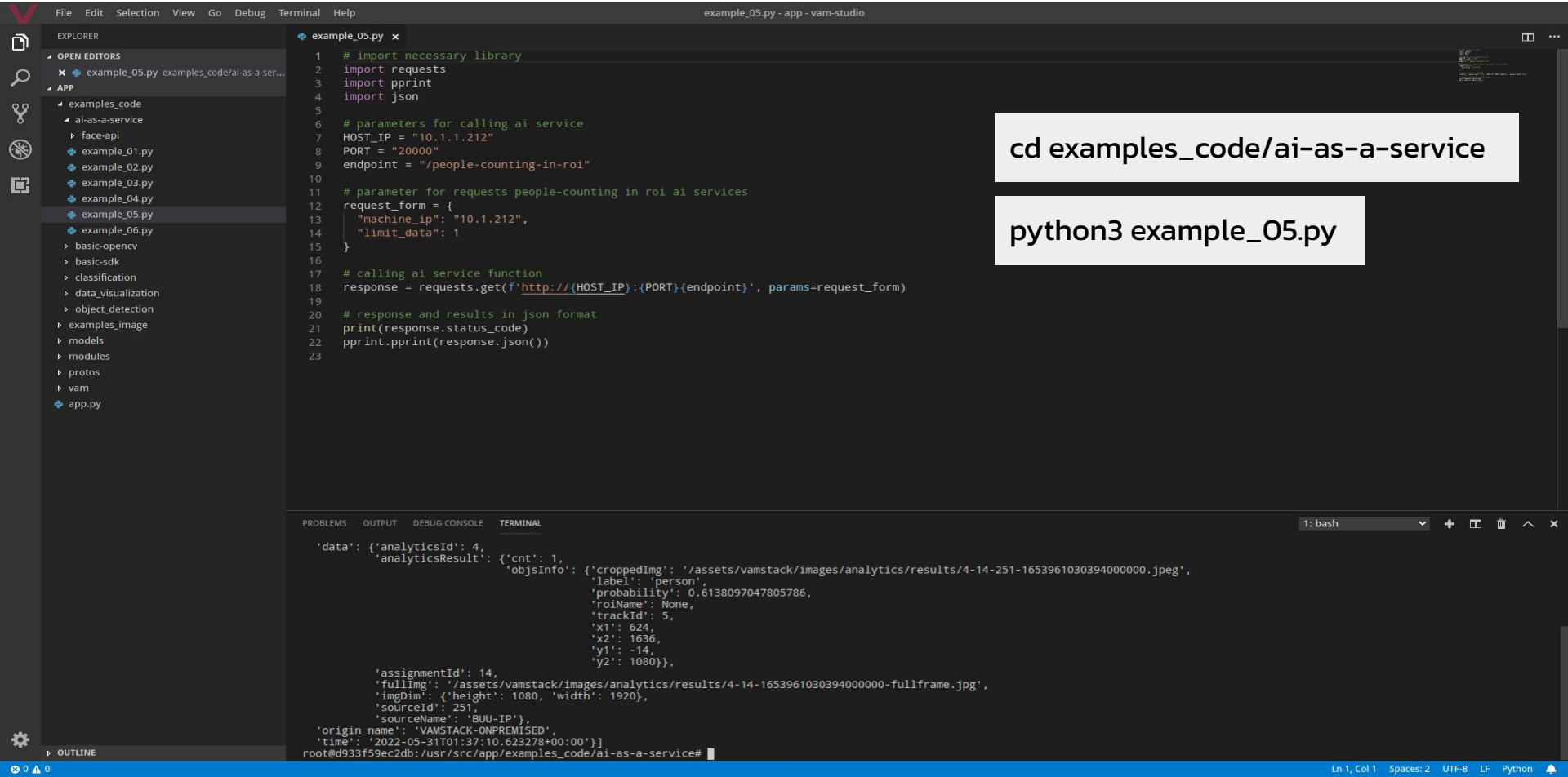
1. คลิก “Assignment” ที่แถบเมนูด้านซ้ายของหน้า VAM Platform
2. ตั้งค่าพื้นที่ที่สนใจของ “VAM People Counting in ROI”
3. คลิกปุ่มเปลี่ยนสถานะของ “VAM People Counting in ROI” ให้เป็น “Running”



3. คลิกที่การ์ด “VAM People Counting in ROI” เพื่อดูการประมวลผลและวิเคราะห์ข้อมูล



People Counting

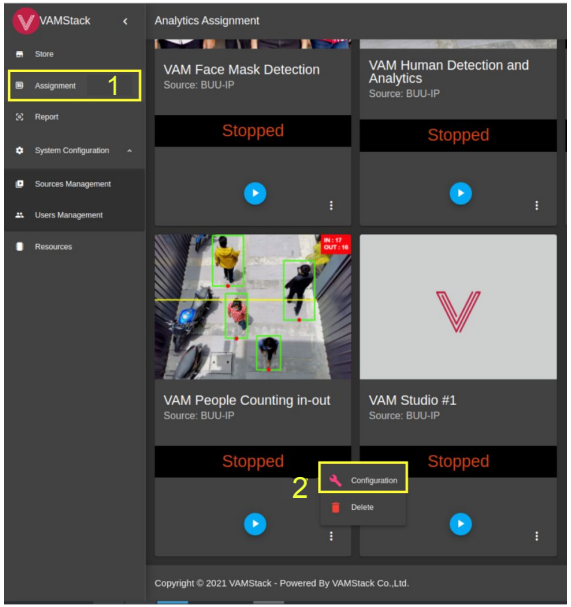


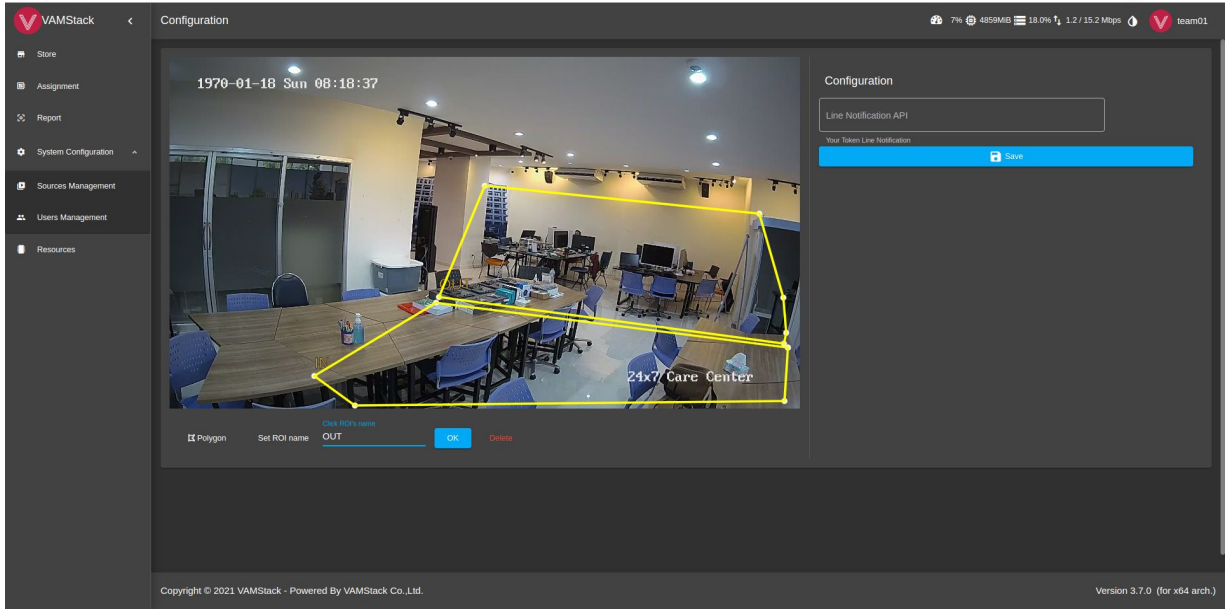
cd examples_code/ai-as-a-service

python3 example_05.py

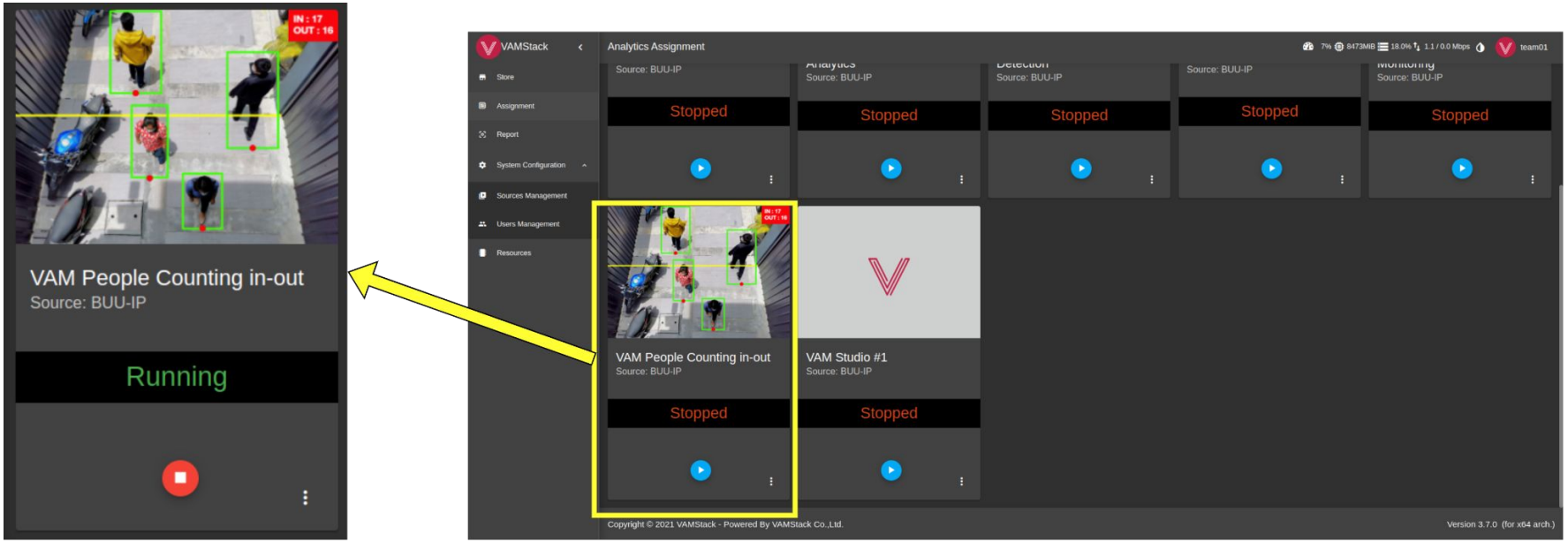
VAM Security & Surveillance - People Counting in-out

1. คลิก “Assignment” ที่แถบเมนูด้านซ้ายของหน้า VAM Platform
2. ตั้งค่าพื้นที่ที่สนใจออกเป็น 2 ส่วนคือ พื้นที่ขาเข้า (IN) และพื้นที่ขาออก (OUT)

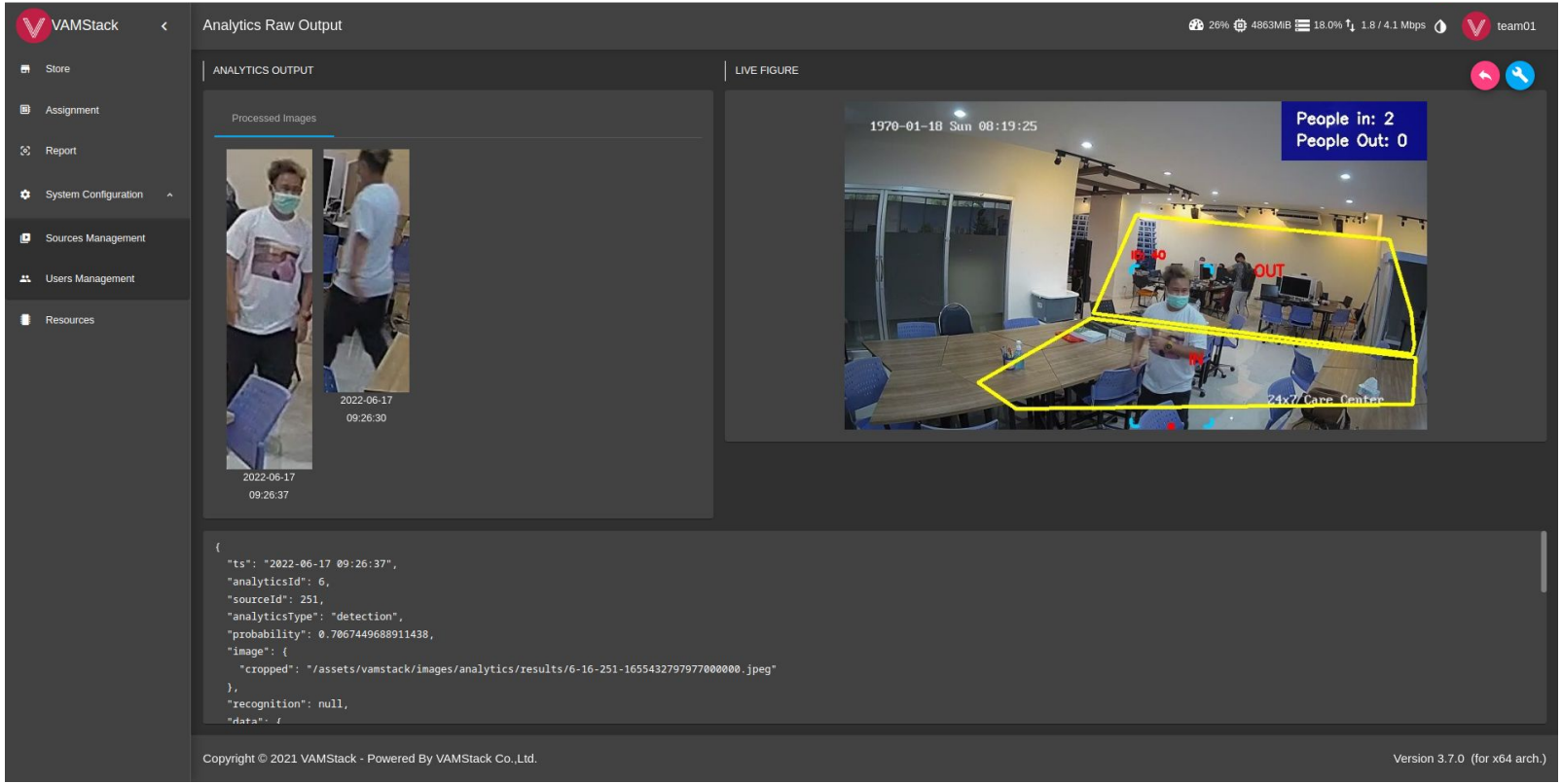




3. คลิกปุ่มเปลี่ยนสถานะของ “VAM People Counting in-out” ให้เป็น “Running”



3. คลิกที่การ์ด “VAM People Counting in-out” เพื่อดูการประมวลผลและวิเคราะห์ข้อมูล



People Counting in-out