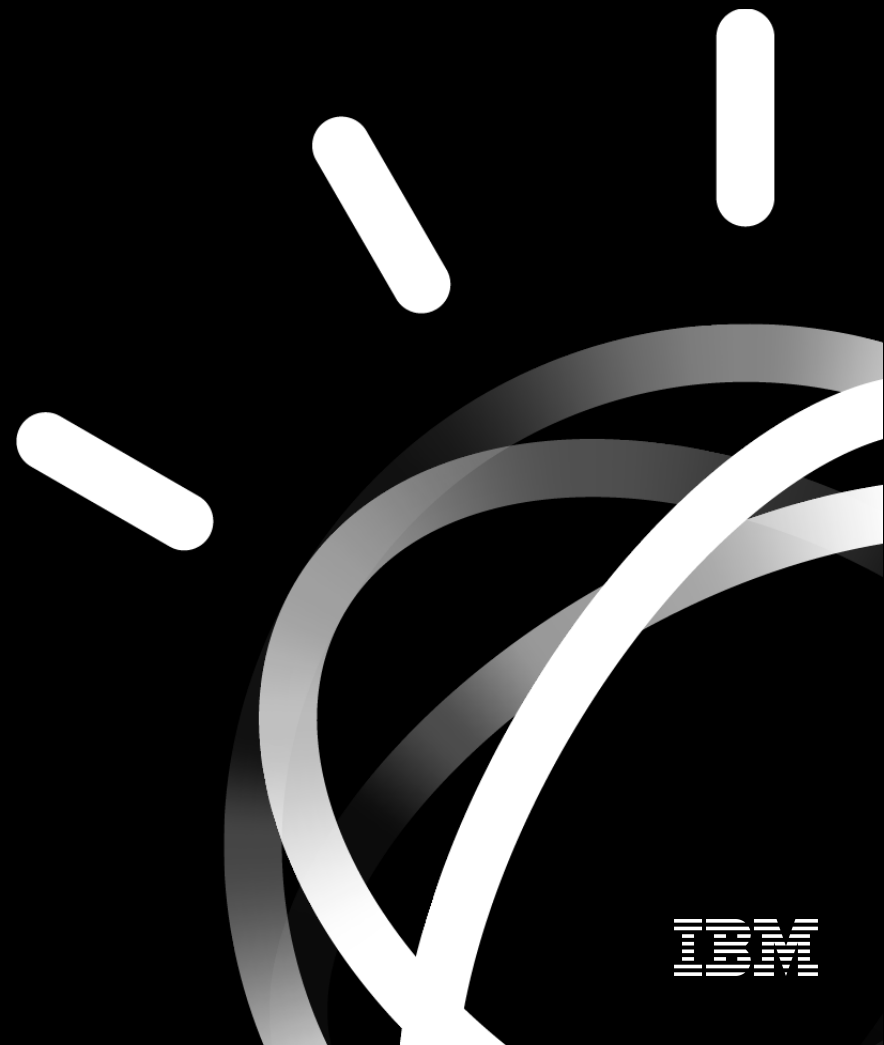
Drone Programming

Aerial Image Object Detection with Watson Visual Recognition

John Walicki
IBM Developer Advocate
CTO IoT / Edge Advocacy
@johnwalicki



IBM **Developer**

IBM

Agenda

- Tello Intro
  - Drone Programming – Python / Node.js / Node-RED / Go / Swift
- Watson Studio – Drone Image Analysis
  - Training a Watson Visual Recognition Custom Object Detection Model
  - Save Lives!
- DroneAID - Object Detection with Tello Camera
  - Demo!
- Control your Tello Drone using Node-RED
  - Demo!
- Program your Drone with Python
  - Demo / Hands On !

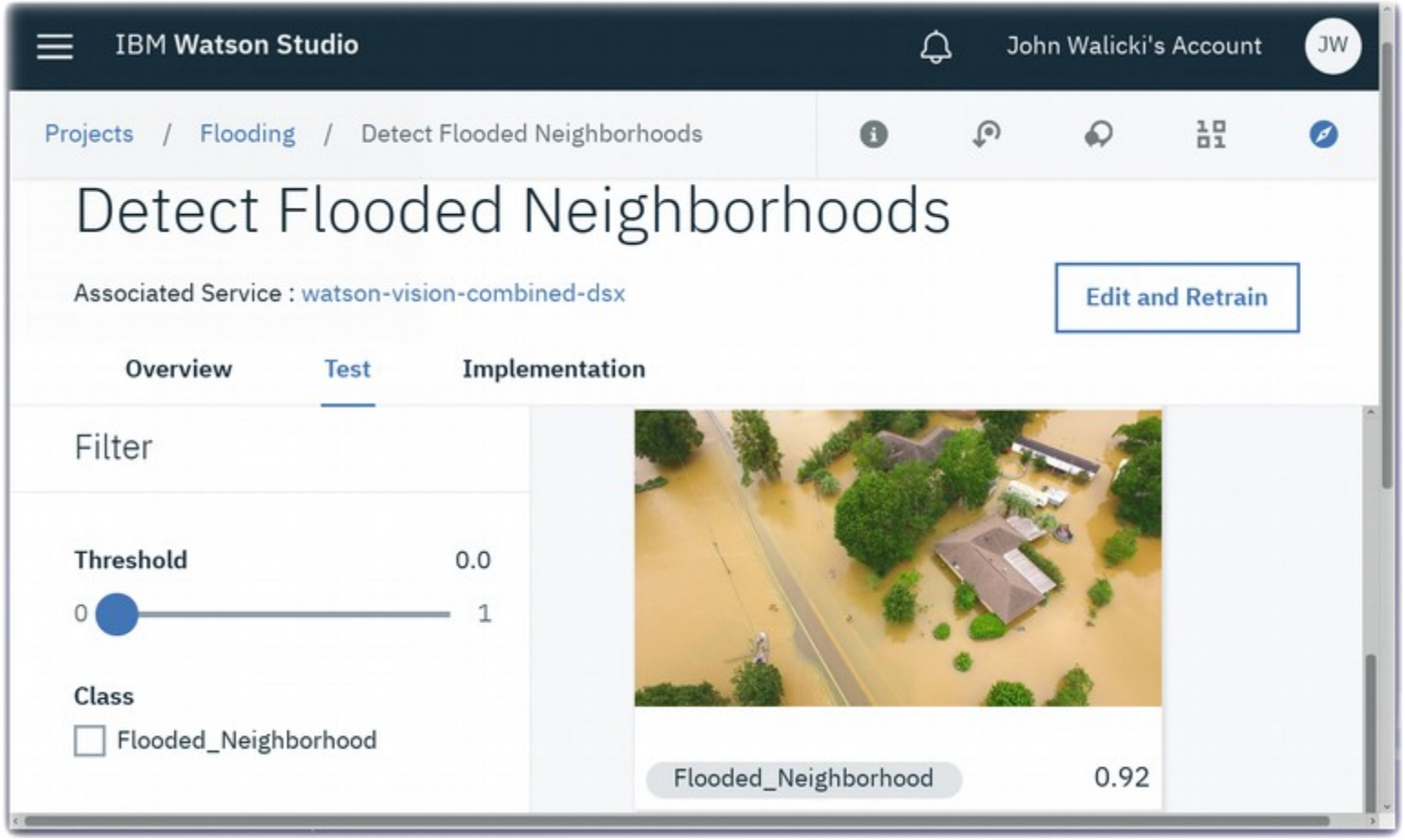
http://ibm.biz/TorontoDroneDrop

Tello Programming

Python
Node.js
Node-RED
Go
Swift

Tello SDK Docs: https://www.ryzerobotics.com/tello/downloads

Drone Aerial Image Detection with Watson Studio Visual Recognition



Get the Code:

https://github.com/IBM/drones-iot-visual-recognition
https://developer.ibm.com/tutorials/use-drones-after-floods-to-help-survivors-watson-visual-recognition/

Detect Survivors on Rooftops

- Use drone aerial images, Watson Studio and Watson Visual Recognition to survey flood damaged neighborhoods, identify homes with survivors on rooftops and detect rescue boats
- A lifesaving scenario might be to use drones to identify flood survivors and direct rescue boats to the victims using GPS and visual recognition.
- Train a model to detect survivors waving their arms
  - Find drone images of people waving their arms for help
  - Zip the images (>10) / Upload to Cloud Object Store
  - Create a new class and retrain your model
- Train another Watson Visual Recognition Model to detect Wildfire Burned Neighborhoods using drone images
  - https://developer.ibm.com/tutorials/detect-wildfire-damaged-homes-using-drone-images-watson-visual-recognition/

rescueboat.jpg

Rescue Boat	0.62
Rooftop Survivors	0.57
Flooded Neighborhood	0.11

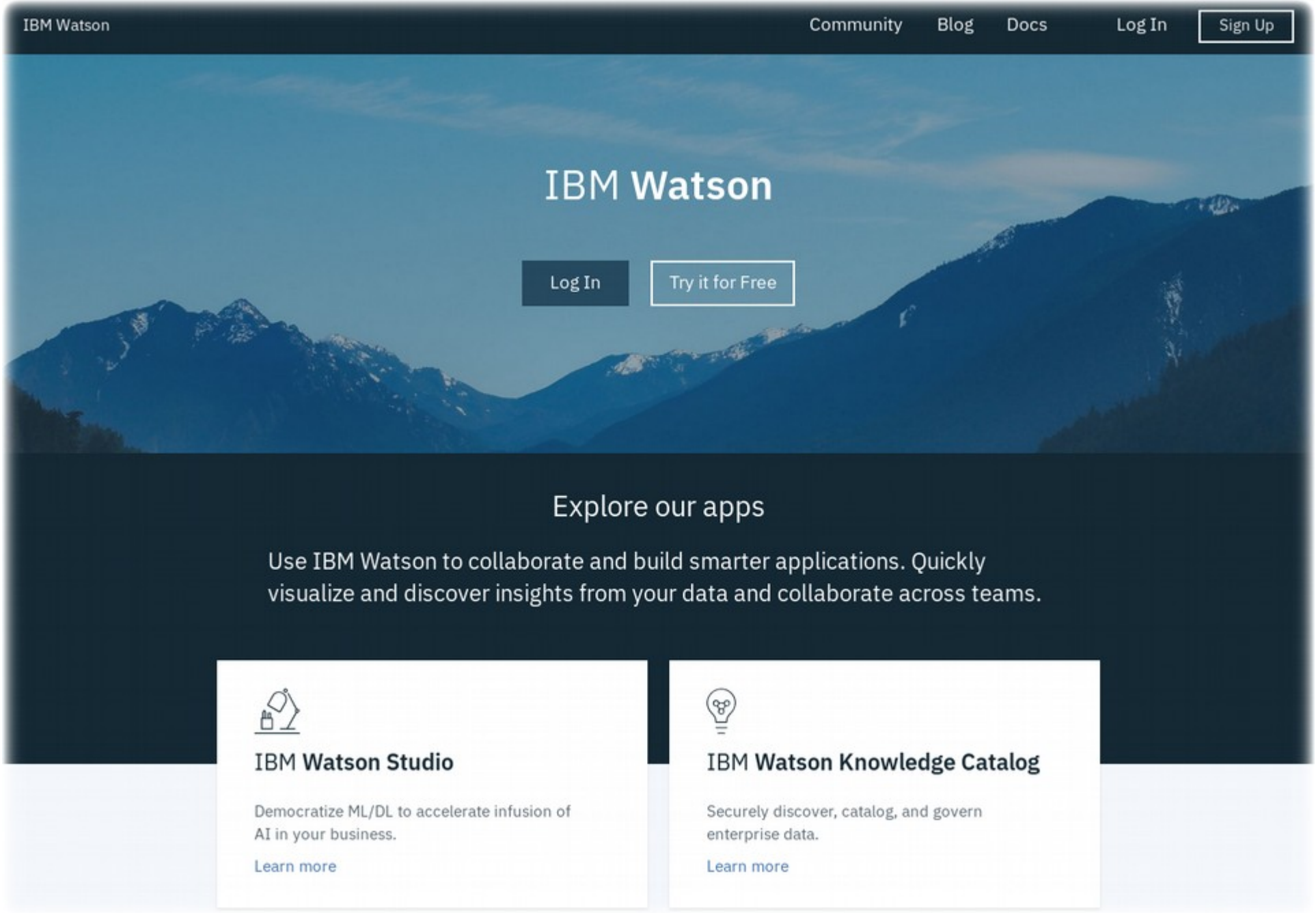
grande.jpg

Flooded Neighborhood	0.89
Rescue Boat	0.02
Rooftop Survivors	0.00

tokyoflood.jpg

Rooftop Survivors	0.91
Rescue Boat	0.00
Flooded Neighborhood	0.00

- Use drone aerial images, Watson Studio and Watson Visual Recognition to survey flood damaged neighborhoods, identify homes with survivors on rooftops and detect rescue boats



IBM Watson Studio

Upgrade

IBM

JW

Projects / Flooding / Flooding

Flooding

Associated Service : watson-vision-combined-dsx

Edit and Retrain

Overview | Test | Implementation

Filter

Threshold 0.0

0 — 1

Class

- [ ] Flooded Neighborhood
- [ ] Rescue Boat
- [ ] Rooftop Survivors

× Clear results

1_s.jpg

Rescue Boat	0.89
Rooftop Survivors	0.06
Flooded Neighborhood	0.02

rescueboat.jpg

Rescue Boat	0.62
Rooftop Survivors	0.57
Flooded Neighborhood	0.11

grande.jpg

Flooded Neighborhood	0.89
Rescue Boat	0.02
Rooftop Survivors	0.00

screen-shot-2016-04-20-at-105252.jpg

Flooded Neighborhood	0.89
Rescue Boat	0.08
Rooftop Survivors	0.00

tokyoflood.jpg

Rooftop Survivors	0.91
Rescue Boat	0.00
Flooded Neighborhood	0.00

sjm-floodalert-0226-04.jpg

Rescue Boat	0.83
Flooded Neighborhood	0.30
Rooftop Survivors	0.00

Homework - Learning Objectives

After completing this workshop you will be able to:

- Create a Visual Recognition model in Watson Studio running in IBM Cloud
- Capture images from a drone and zip them into a class (provided)
- Train a model to identify objects in the images
- Score the identified objects

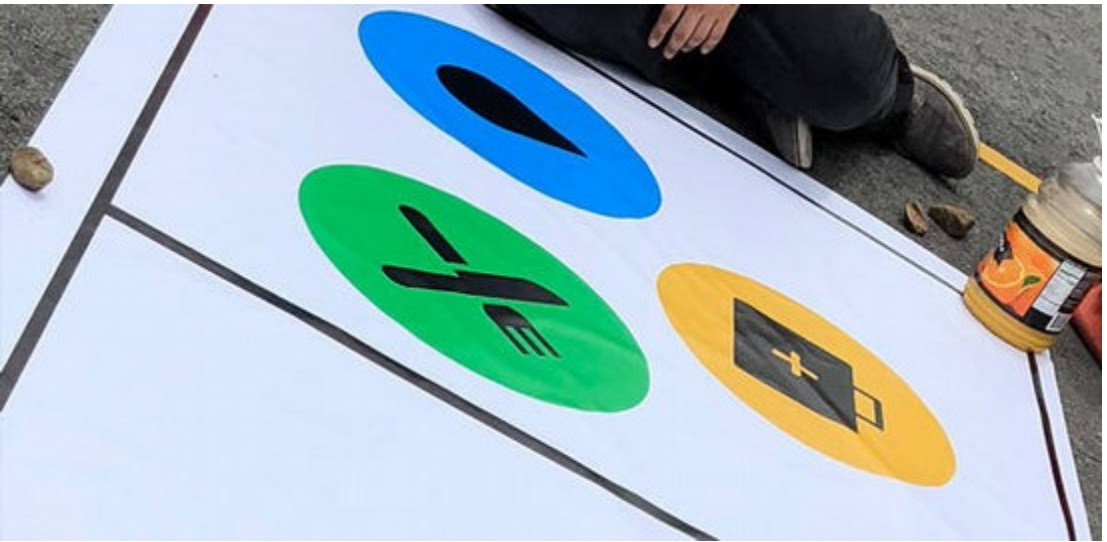
http://ibm.biz/TorontoDroneDrop

DroneAid

Pedro Cruz

IBM Developer Advocate

Founder of DroneAID




Pedro lives here

No Power

No Cell Signal

No Food or Water

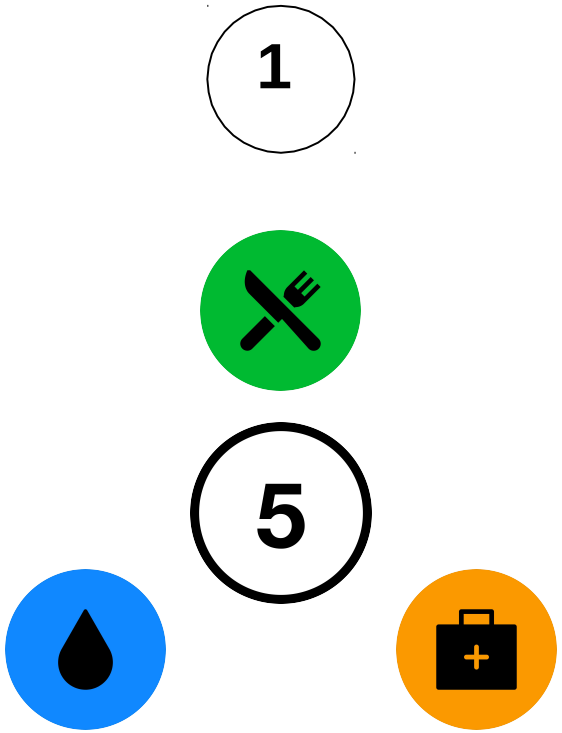

S.O.S
AGUA
COMIDA
#MARIANA2
SE LEVANTA


HELP!

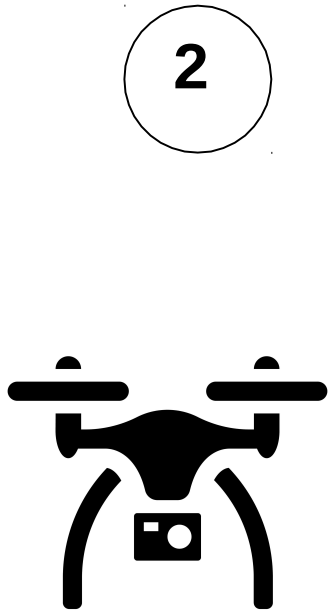
Aerial Scout for First Responders

How it works

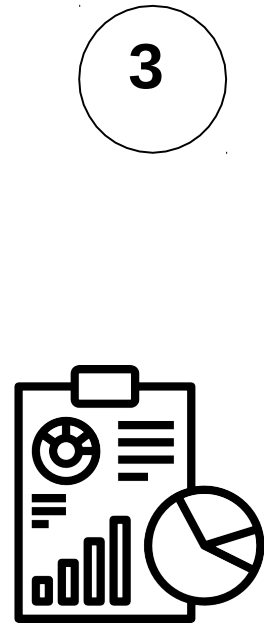
DroneAid



Needs



Fly



Report

United Nations Office for the Coordination of Humanitarian Affairs

Symbol	Meaning	Symbol	Meaning
	Immediate Help Needed		Shelter Needed
	No Help Needed		First Aid Kit Needed
	Water Needed		Area with Children in Need
	Food Needed		Area with Elderly in Need

DroneAid

Map Satellite

Detected: “SOS”, “AGUA”, “COMIDA”

GPS Coordinates: 18.129938, -65.858775 (Humacao, PR)

Timestamp: 11:24 AM, October 20, 2017

Cancel

Confirm

DroneAid

Legend

- SOS
- OK
- First Aid
- Water
- Food

Escuela Santiago Veve Calzada
Iglesia de Dios Pentecostal, MI Tejas 1
Escuela Manuel Surillo
Supermercado Kennedy
Danny's Agrocentro
Iglesia Jesucristo Senor Nuestro
Capilla Catolica Corpus Cristi
Head Start- Jagüeyes
Parque Felix Herrera
LAS PIEDRAS
YABUCOA
HUMACAO
Mariana
Río Humacao
Sector Puesta Del Sol
Sector Sumidero Aguas Buenas
Sector Diamante
Cam Cesar Maldonado
Cam Los Maldonado
Cam Los Sanabria
Cam Blas Cuadrado
Cam Julio De Léon
Cam Collazo
Cam Diana Maldonado
Cam Epifanio Beltrán
Cam Soto Moreno
PR-908
9921
9908
9904
9903
905
909

Google

DroneAid **Tello Demo** | DroneAid model inferencing on video stream from a Tello drone

Prediction

On

Confidence

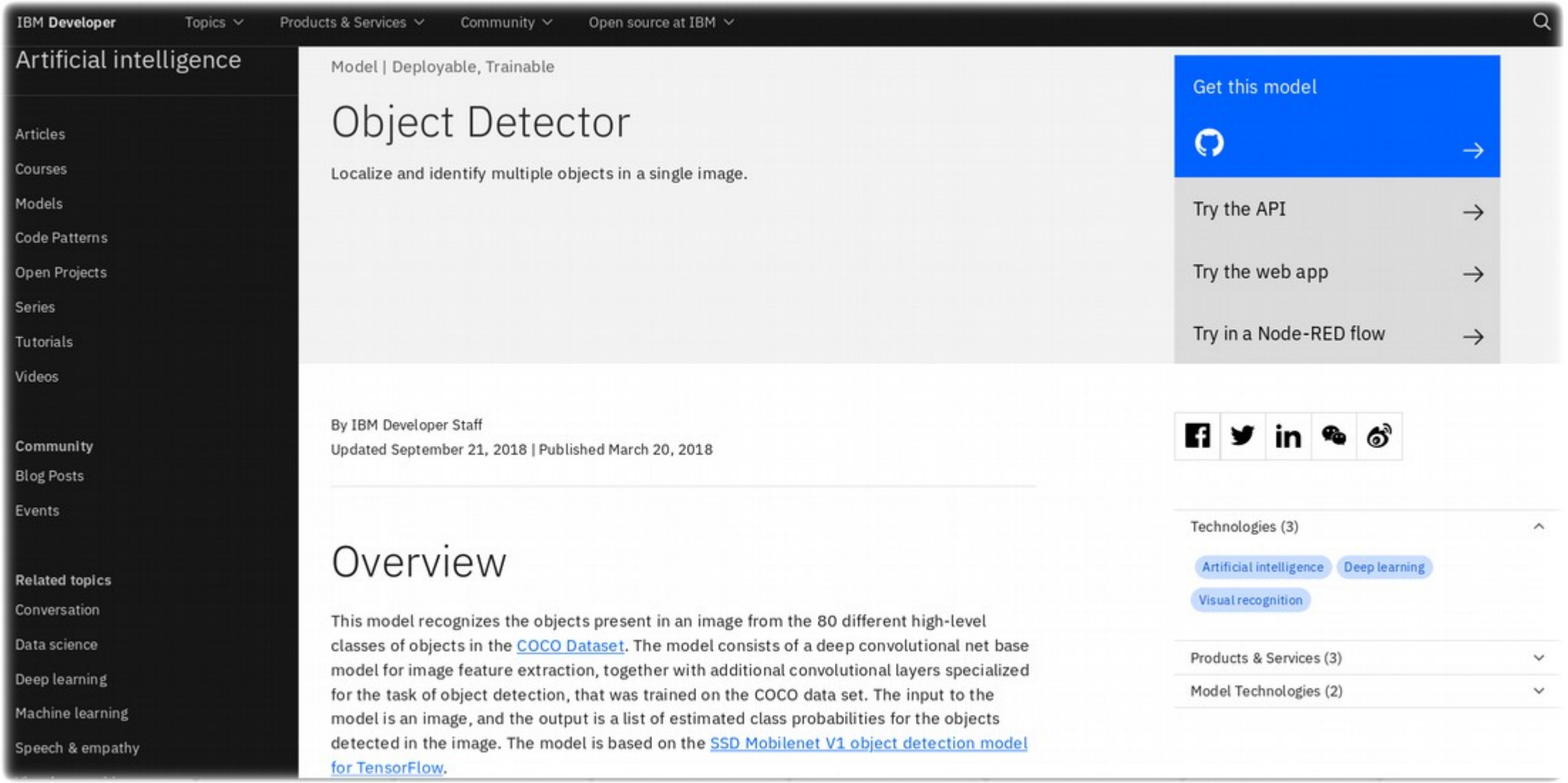
0.4

Stop stream

children	0
elderly	0
firstaid	1
food	0
ok	0
shelter	0
sos	3
water	0

Get the Code:

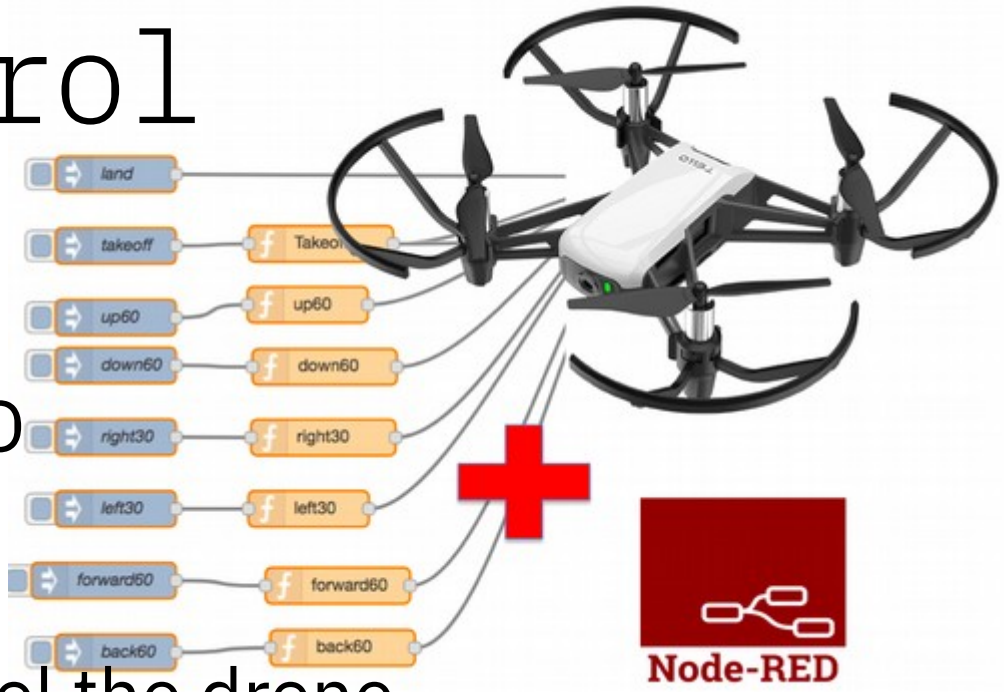
https://developer.ibm.com/exchanges/models/all/max-object-detector/



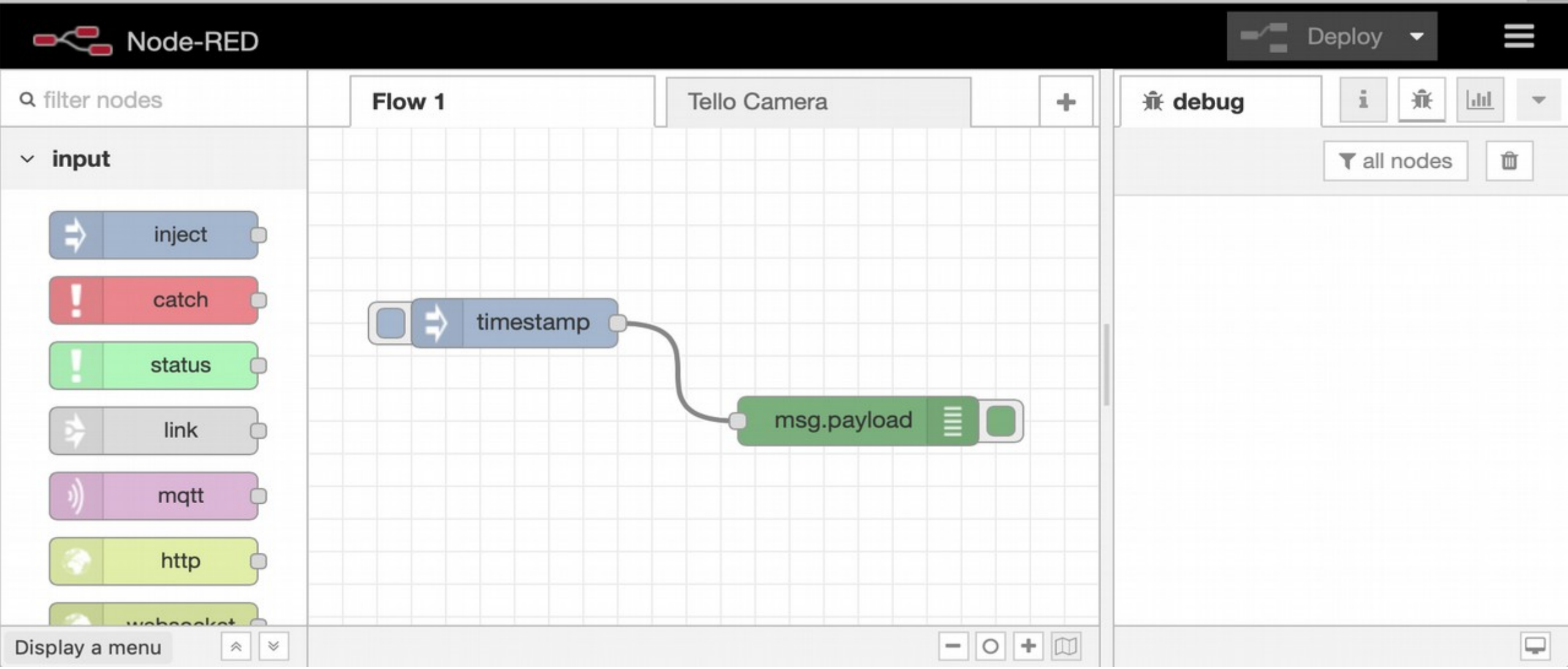
Get the Code:

https://github.com/Code-and-Response/DroneAID

Node-RED Tello Control

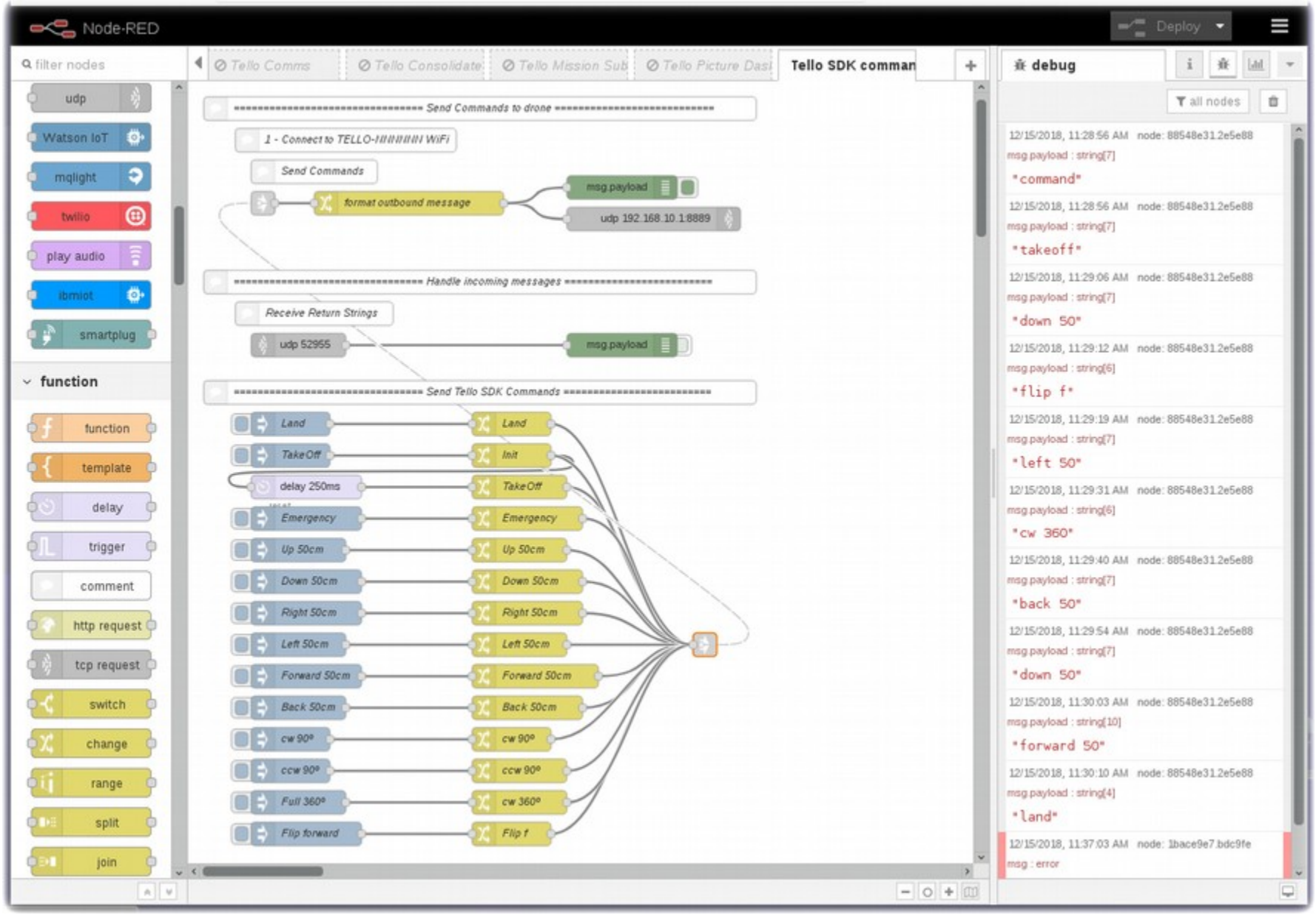


- Control your Tello Drone using Node-RED
  - Learn about Node-RED
  - Send SDK commands to the drone
  - Build a Node-RED Dashboard to control the drone
  - Receive telemetry data from the drone
  - Create missions for your drone
  - Take pictures using the drone camera
  - Watson Visual Recognition and Drone image analysis
  - Train a Visual Recognition Custom Classifier for drone images

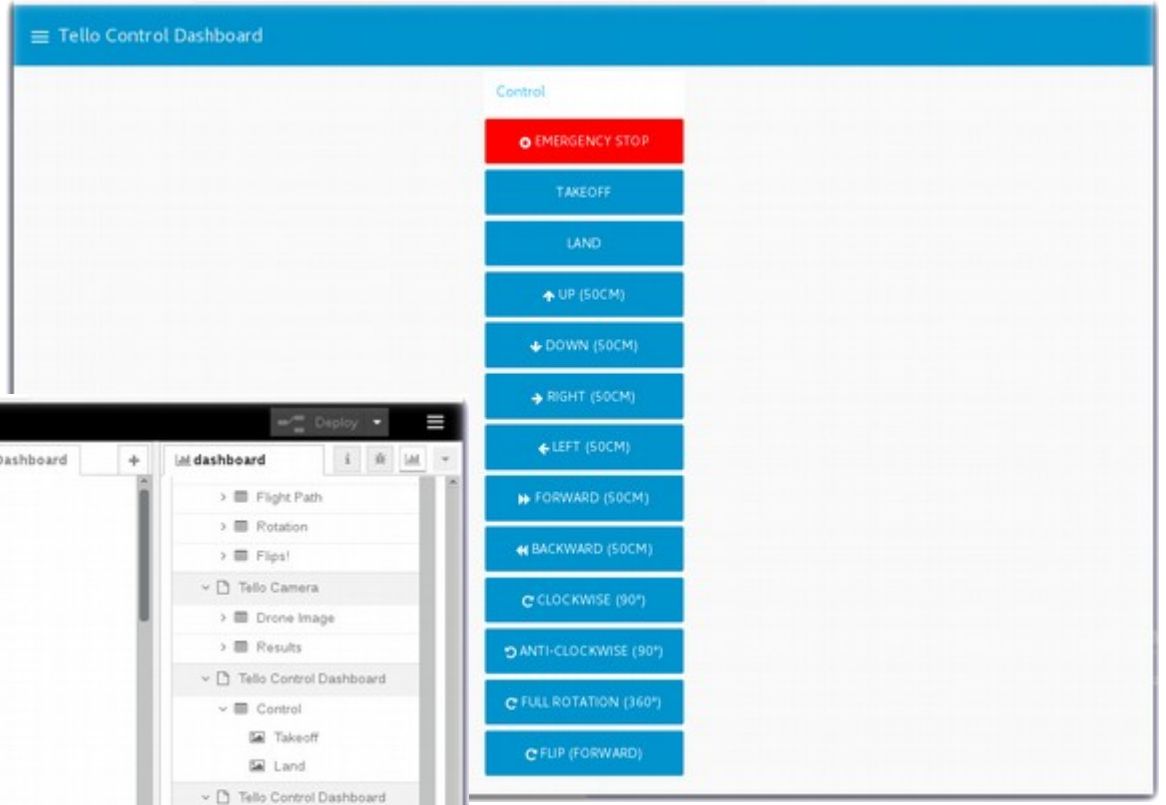


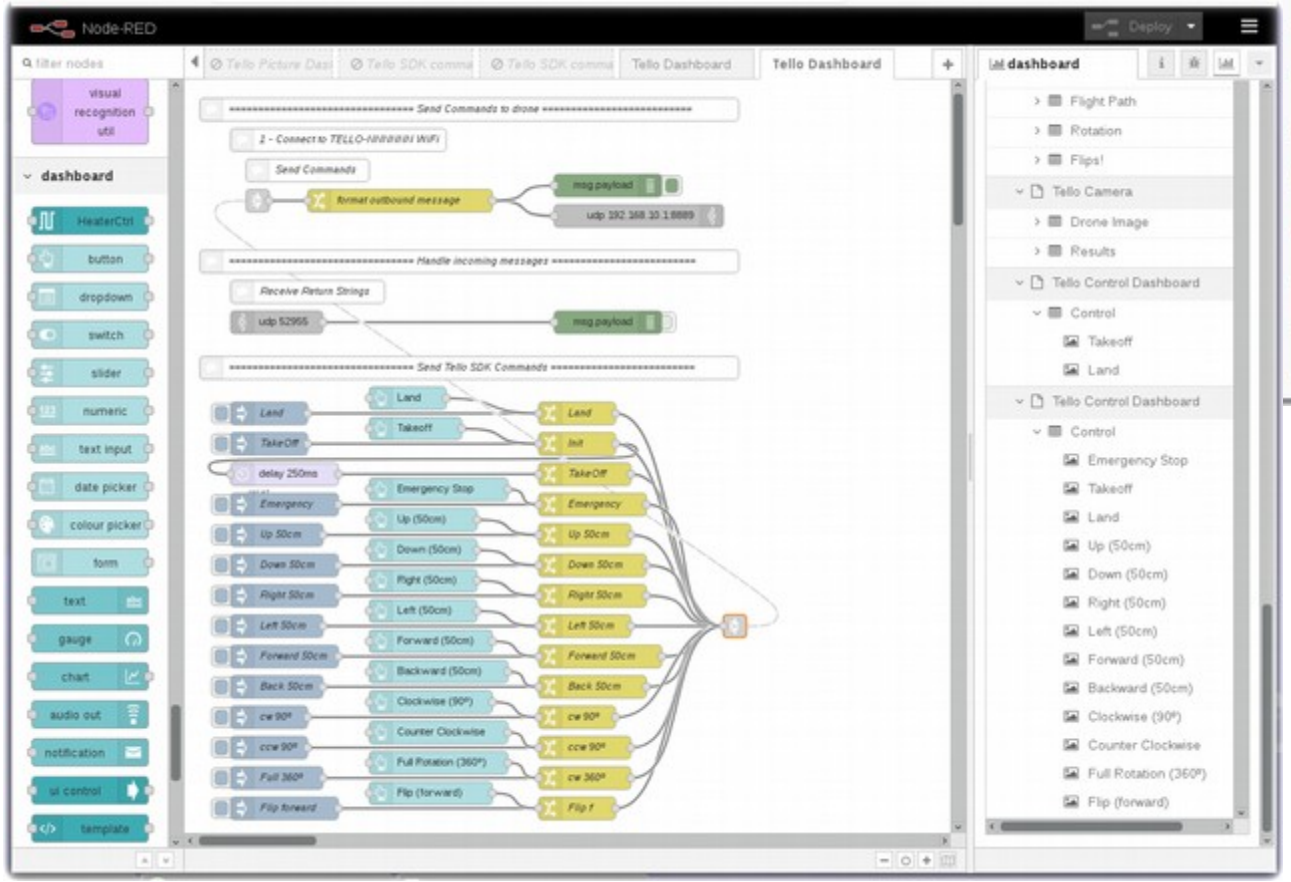
Node-RED is a open source project and flow-based programming environment from the JS Foundation. It provides a palette of nodes that allow users to very quickly wire up IoT applications that can combine streams of both physical and digital events. The Node.JS runtime is easy to install on both devices and the cloud, and provides a framework for extending its capabilities.

Send Commands to the Drone using Node-RED

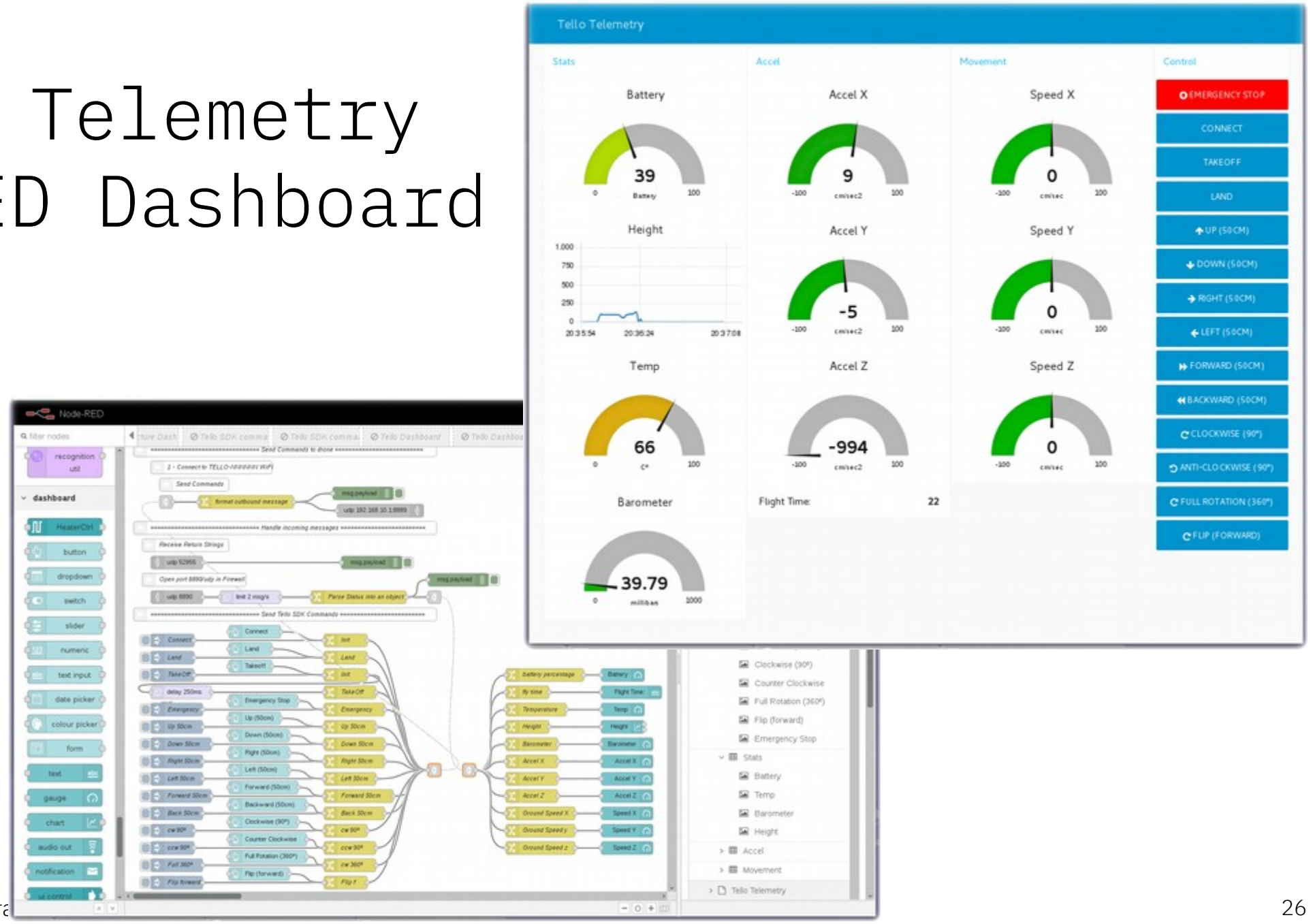


Tello Control Center Node-RED Dashboard

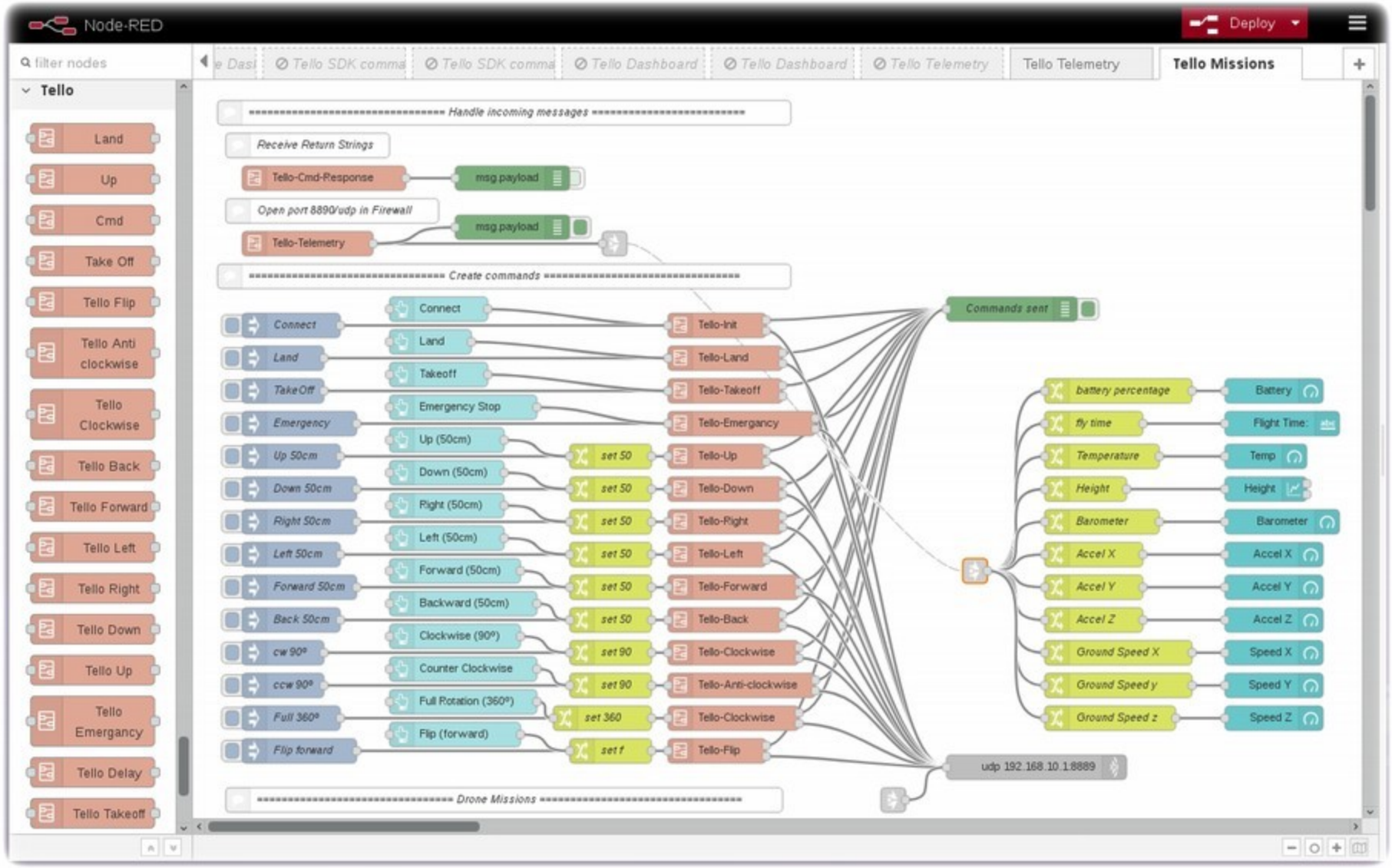




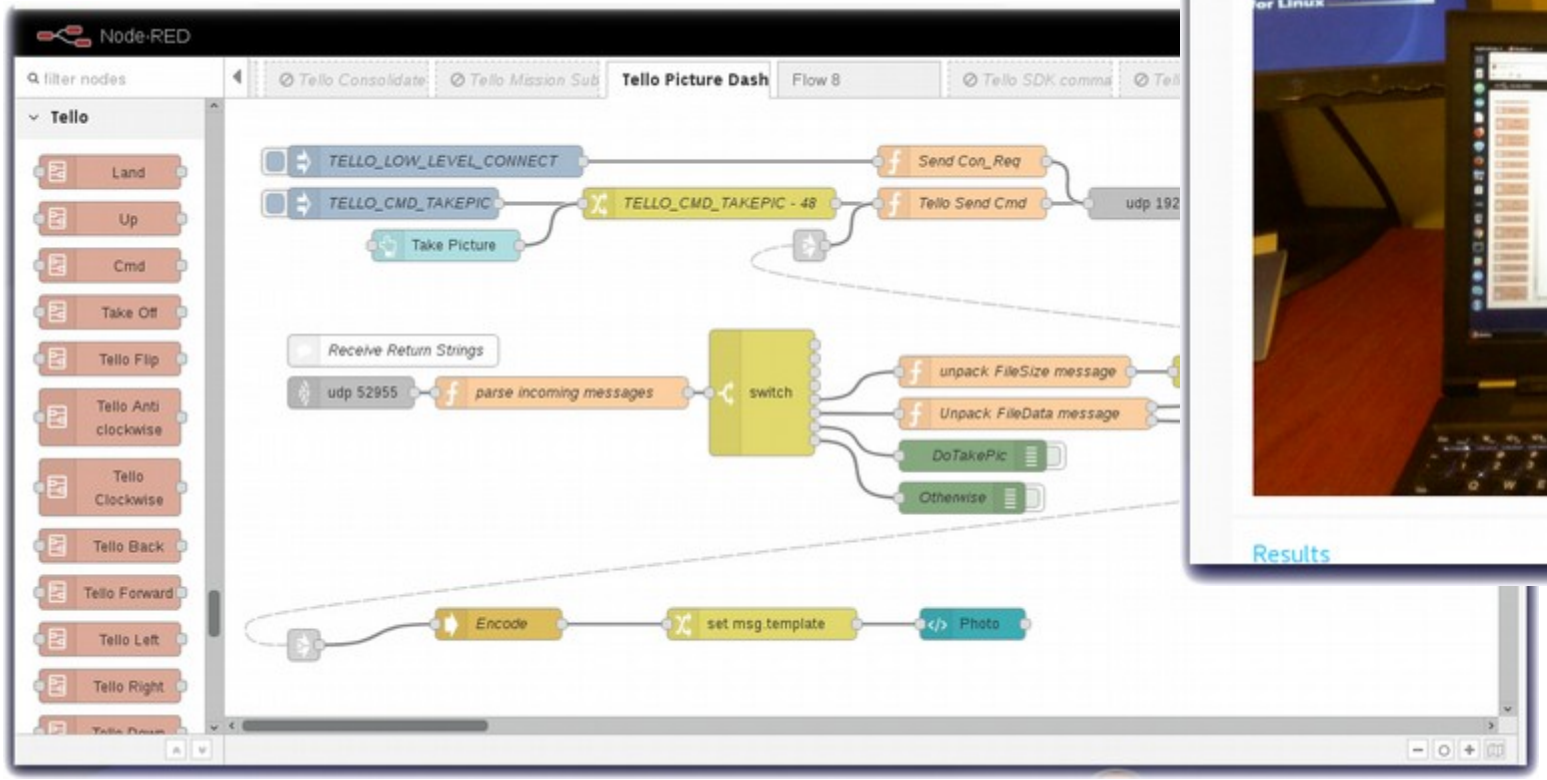
Tello Telemetry Node-RED Dashboard

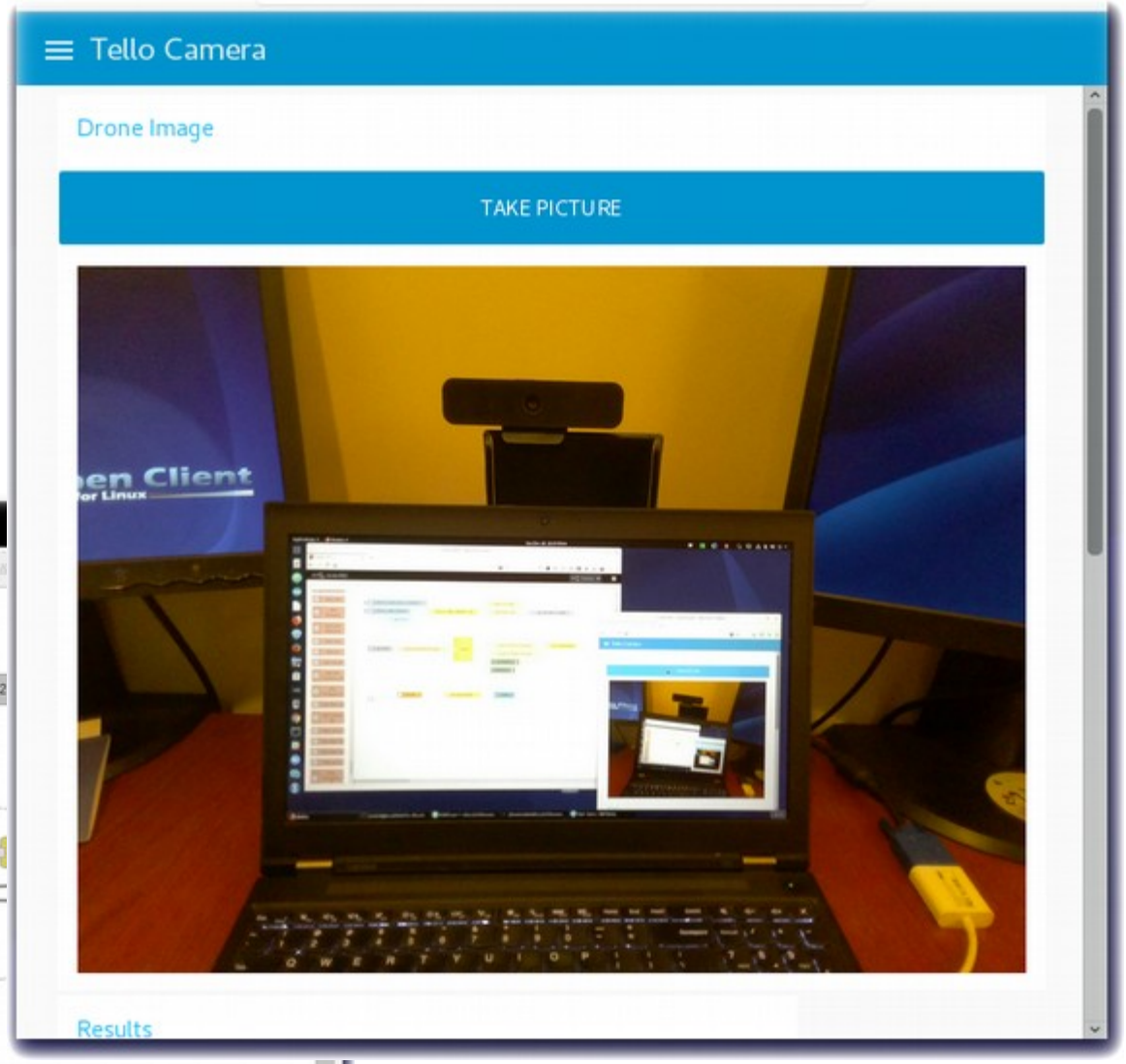


Send Tello Drone on Autonomous Missions

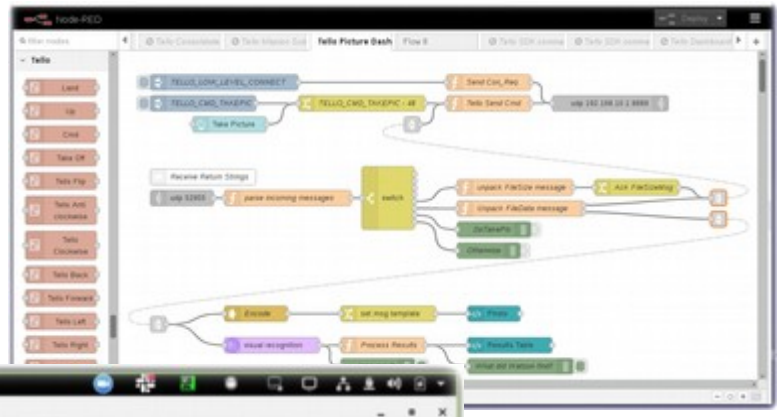


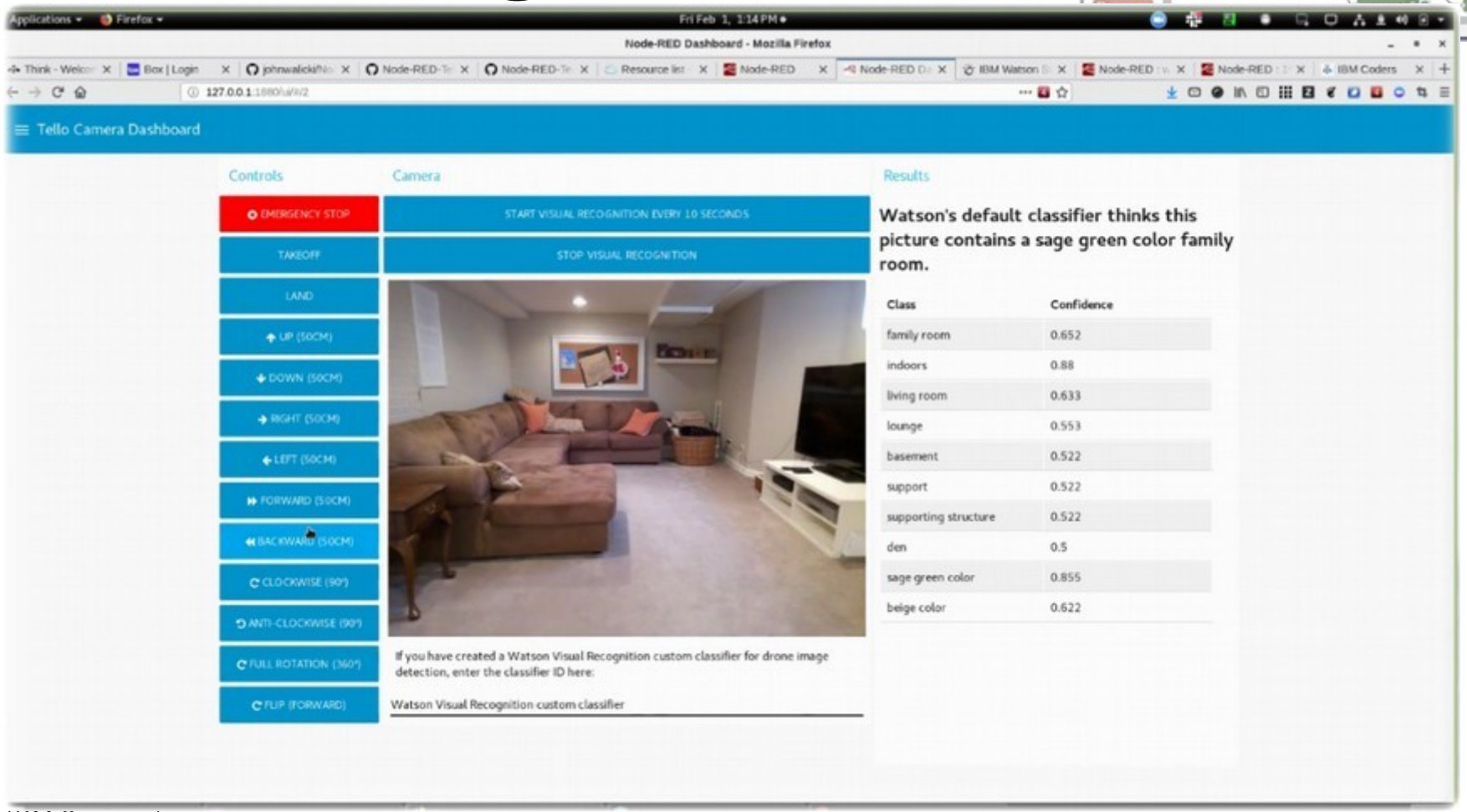
Tello Camera



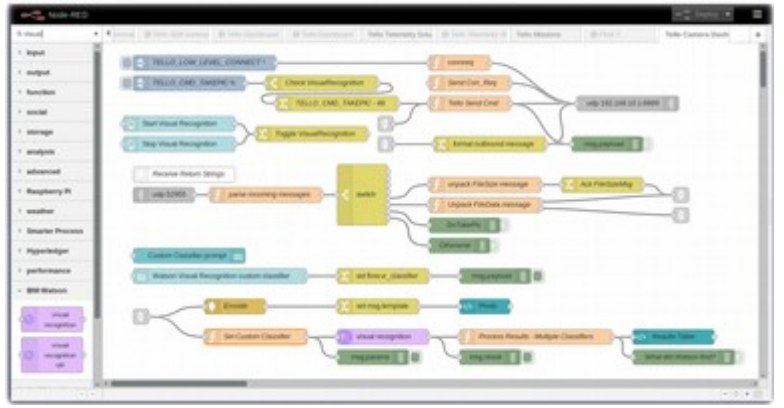


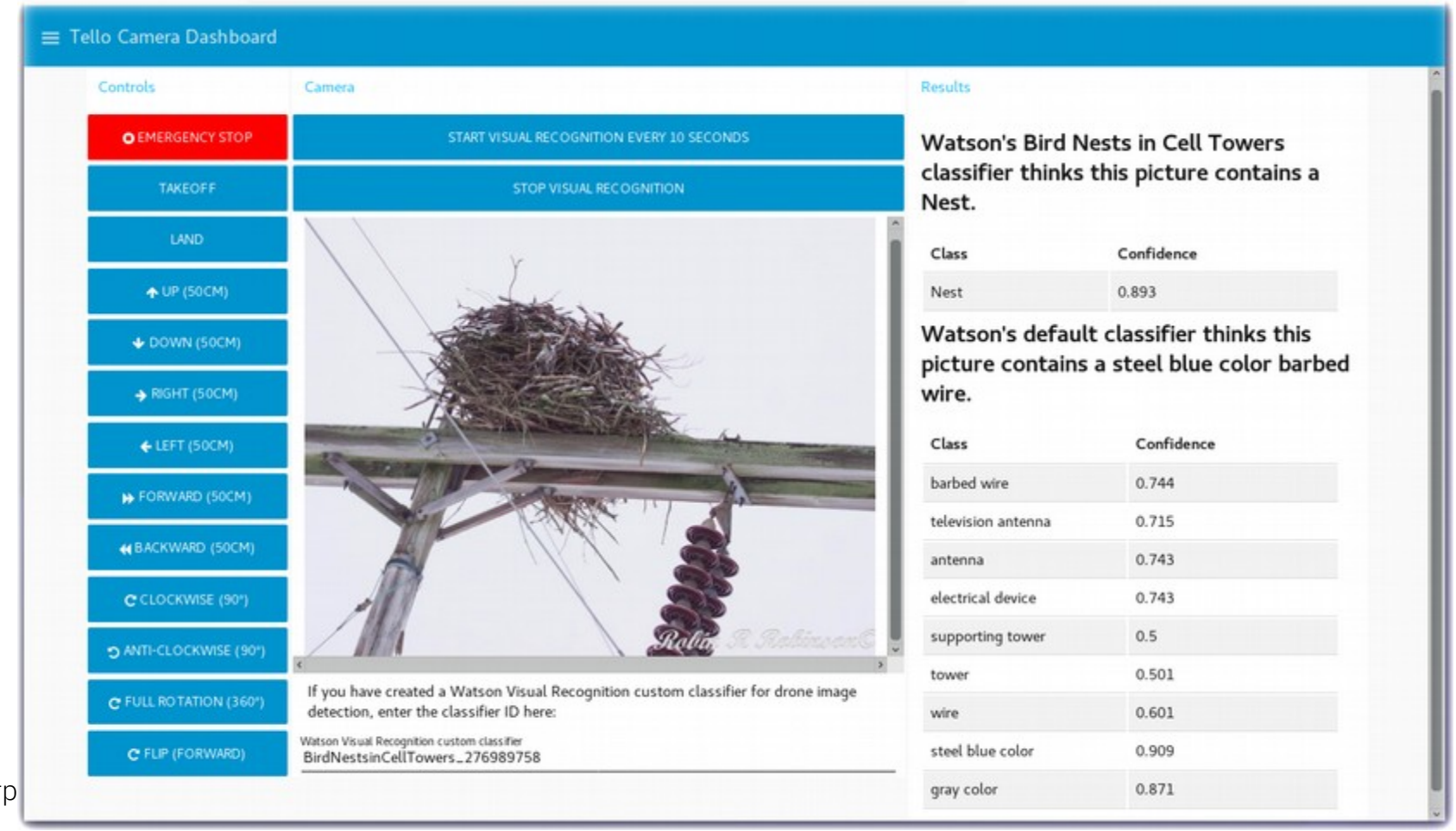
Tello Camera with Watson Visual Recognition





Tello Camera with Watson Visual Recognition Custom Classifier





Get the Code:

https://github.com/johnwalicki/Node-RED-Tello-Control

TelloPy

https://github.com/hanyazou/TelloPy

```
$ pip install tellopy
```

What does the code look like?

```
from time import sleep
import tellopy


def handler(event, sender, data, **args):
    drone = sender
    if event is drone.EVENT_FLIGHT_DATA:
        print(data)


def test():
    drone = tellopy.Tello()
    try:
        drone.subscribe(drone.EVENT_FLIGHT_DATA, handler)

        drone.connect()
        drone.wait_for_connection(60.0)
        drone.takeoff()
        sleep(5)
        drone.down(50)
        sleep(5)
        drone.land()
        sleep(5)
    except Exception as ex:
        print(ex)
    finally:
        drone.quit()

if __name__ == '__main__':
    test()
```

Standard setup – we need to access the DJI drone library and the standard time library from python

from time import sleep
import tellopy

What does the code look like?

```
from time import sleep
import tellopy


def handler(event, sender, data, **args):
    drone = sender
    if event is drone.EVENT_FLIGHT_DATA:
        print(data)


def test():
    drone = tellopy.Tello()
    try:
        drone.subscribe(drone.EVENT_FLIGHT_DATA, handler)

        drone.connect()
        drone.wait_for_connection(60.0)
        drone.takeoff()
        sleep(5)
        drone.down(50)
        sleep(5)
        drone.land()
        sleep(5)
    except Exception as ex:
        print(ex)
    finally:
        drone.quit()

if __name__ == '__main__':
    test()
```

Simple Event Handler routine, we'll see it used later, but when the drone raises an event, this is called

Simple "handler" function to print out diagnostic information from your "in-flight" drone. More information can be found in your red folder

What does the code look like?

```
from time import sleep
import tellopy


def handler(event, sender, data, **args):
    drone = sender
    if event is drone.EVENT_FLIGHT_DATA:
        print(data)


def test():
    drone = tellopy.Tello()
    try:
        drone.subscribe(drone.EVENT_FLIGHT_DATA, handler)

        drone.connect()
        drone.wait_for_connection(60.0)
        drone.takeoff()
        sleep(5)
        drone.down(50)
        sleep(5)
        drone.land()
        sleep(5)
    except Exception as ex:
        print(ex)
    finally:
        drone.quit()

if __name__ == '__main__':
    test()
```

drone = tellopy.Tello()

.

.

drone.subscribe()

What does the code look like?

```
from time import sleep
import tellopy


def handler(event, sender, data, **args):
    drone = sender
    if event is drone.EVENT_FLIGHT_DATA:
        print(data)


def test():
    drone = tellopy.Tello()
    try:
        drone.subscribe(drone.EVENT_FLIGHT_DATA, handler)

        drone.connect()
        drone.wait_for_connection(60.0)
        drone.takeoff()
        sleep(5)
        drone.down(50)
        sleep(5)
        drone.land()
        sleep(5)
    except Exception as ex:
        print(ex)
    finally:
        drone.quit()

if __name__ == '__main__':
    test()
```


Standard stuff. We need to have your laptop connect to the drone – be sure your laptop is on the same SSID (wireless network as your drone)

drone.connect()
drone.wait_for_connection()
drone.takeoff()
sleep(5)

What does the code look like?

```
from time import sleep
import tellopy


def handler(event, sender, data, **args):
    drone = sender
    if event is drone.EVENT_FLIGHT_DATA:
        print(data)


def test():
    drone = tellopy.Tello()
    try:
        drone.subscribe(drone.EVENT_FLIGHT_DATA, handler)

        drone.connect()
        drone.wait_for_connection(60.0)
        drone.takeoff()
        sleep(5)
        drone.down(50)
        sleep(5)
        drone.land()
        sleep(5)
    except Exception as ex:
        print(ex)
    finally:
        drone.quit()

if __name__ == '__main__':
    test()
```

Always sleep() between commands

Stay in the try/except block !!

If not the drone.quit() is never called
the drone runs “forever”

Drone control statements:

takeoff() – the drone starts up and hovers at about 2’
down(x) – moves down a speed of ‘x’ (0-100)
land() – drone immediately lands

This is the typical flow: takeoff()
set in motion
sleep (while it moves)

What does the code look like?

```
from time import sleep
import tellopy


def handler(event, sender, data, **args):
    drone = sender
    if event is drone.EVENT_FLIGHT_DATA:
        print(data)


def test():
    drone = tellopy.Tello()
    try:
        drone.subscribe(drone.EVENT_FLIGHT_DATA, handler)

        drone.connect()
        drone.wait_for_connection(60.0)
        drone.takeoff()
        sleep(5)
        drone.down(50)
        sleep(5)
        drone.land()
        sleep(5)
    except Exception as ex:
        print(ex)
    finally:
        drone.quit()

if __name__ == '__main__':
    test()
```

One last thing

If you issue a:

- drone.up(50) and nothing else

The drone will rise up (50 cm/sec) forever (or until 20 feet or so)

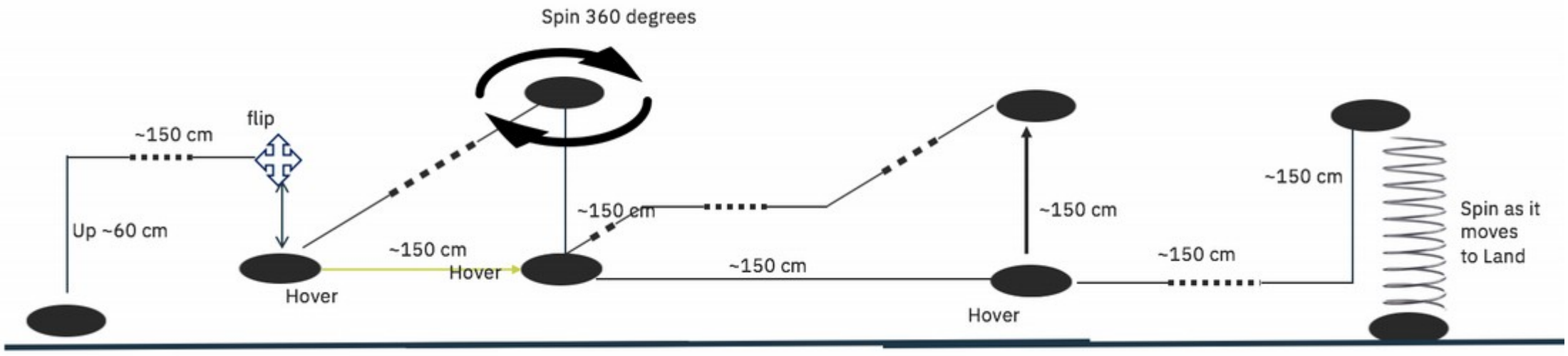
If you issue:

- Drone.up(50)
- Drone.forward(50)

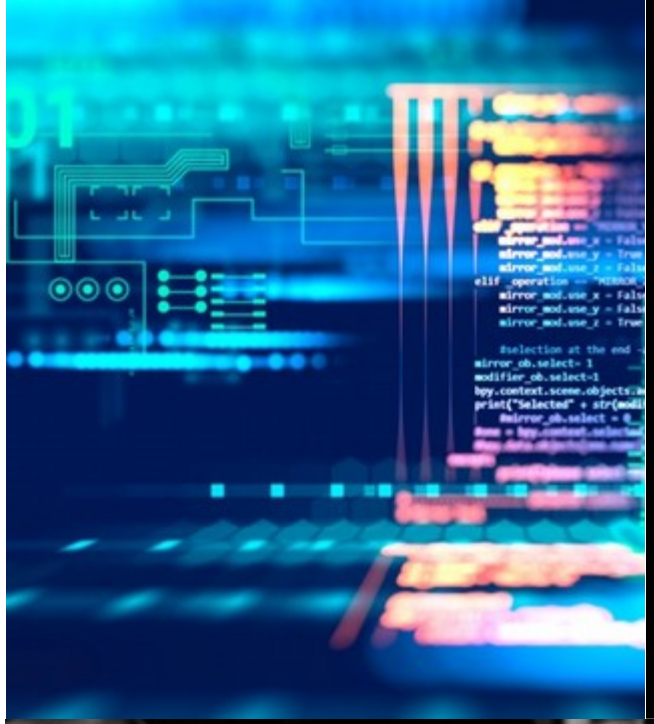
It will go BOTH up and forward at 50 cm/sec – forever

To stop the motion, do a **drone.up(0**) (go up a zero speed – ie, hover) and **drone.forward(0)**

Create a Tello Flight Plan



Thank you



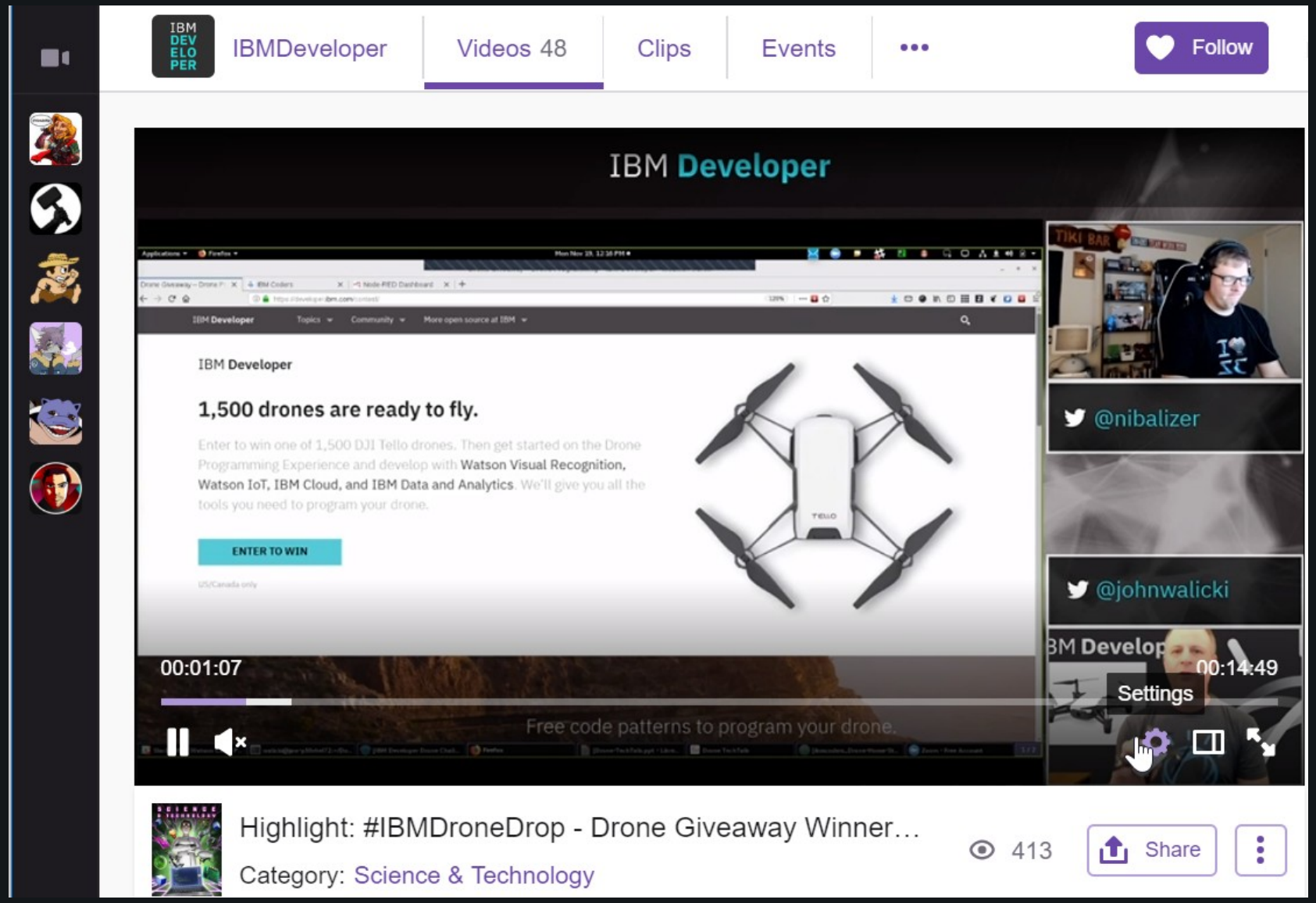
John Walicki

CTO, IoT / Edge Developer Advocacy

IBM Academy of Technology Leadership Team

- twitter.com/johnwalicki
- github.com/johnwalicki
- developer.ibm.com/profiles/walicki
- linkedin.com/in/jowalicki

Twitch TV | IBM Developer Channel



Please note

IBM’s statements regarding its plans, directions, and intent are subject to change or withdrawal without notice and at IBM’s sole discretion.

Information regarding potential future products is intended to outline our general product direction and it should not be relied on in making a purchasing decision.

The information mentioned regarding potential future products is not a commitment, promise, or legal obligation to deliver any material, code or functionality. Information about potential future products may not be incorporated into any contract.

The development, release, and timing of any future features or functionality described for our products remains at our sole discretion.

Performance is based on measurements and projections using standard IBM benchmarks in a controlled environment. The actual throughput or performance that any user will experience will vary depending upon many factors, including considerations such as the amount of multiprogramming in the user’s job stream, the I/O configuration, the storage configuration, and the workload processed. Therefore, no assurance can be given that an individual user will achieve results similar to those stated here.

Notices and disclaimers

U.S. Government Users Restricted Rights — use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM.

Information in these presentations (including information relating to products that have not yet been announced by IBM) has been reviewed for accuracy as of the date of initial publication and could include unintentional technical or typographical errors. IBM shall have no responsibility to update this information. **This document is distributed "as is" without any warranty, either express or implied. In no event, shall IBM be liable for any damage arising from the use of this information, including but not limited to, loss of data, business interruption, loss of profit or loss of opportunity.** IBM products and services are warranted per the terms and conditions of the agreements under which they are provided.

IBM products are manufactured from new parts or new and used parts.
In some cases, a product may not be new and may have been previously installed. Regardless, our warranty terms apply."

Any statements regarding IBM's future direction, intent or product plans are subject to change or withdrawal without notice.

Performance data contained herein was generally obtained in a controlled, isolated environments. Customer examples are presented as illustrations of how those customers have used IBM products and the results they may have achieved. Actual performance, cost, savings or other results in other operating environments may vary.

References in this document to IBM products, programs, or services does not imply that IBM intends to make such products, programs or services available in all countries in which IBM operates or does business.

Workshops, sessions and associated materials may have been prepared by independent session speakers, and do not necessarily reflect the views of IBM. All materials and discussions are provided for informational purposes only, and are neither intended to, nor shall constitute legal or other guidance or advice to any individual participant or their specific situation.

It is the customer's responsibility to insure its own compliance with legal requirements and to obtain advice of competent legal counsel as to the identification and interpretation of any relevant laws and regulatory requirements that may affect the customer's business and any actions the customer may need to take to comply with such laws. IBM does not provide legal advice or represent or warrant that its services or products will ensure that the customer follows any law.

Notices and disclaimers continued

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products about this publication and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products.
Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products. IBM does not warrant the quality of any third-party products, or the ability of any such third-party products to interoperate with IBM's products. **IBM expressly disclaims all warranties, expressed or implied, including but not limited to, the implied warranties of merchantability and fitness for a purpose.**

The provision of the information contained herein is not intended to, and does not, grant any right or license under any IBM patents, copyrights, trademarks or other intellectual property right.

IBM, the IBM logo, ibm.com and [names of other referenced IBM products and services used in the presentation] are trademarks of International Business Machines Corporation, registered in many jurisdictions worldwide. Other product and service names might be trademarks of IBM or other companies. A current list of IBM trademarks is available on the Web at "Copyright and trademark information" at: www.ibm.com/legal/copytrade.shtml.