**AWS(AMAZON WEB SERVICES) EC2 BUILD**

1) Create new AWS account from:

<https://signin.aws.amazon.com/>

2) Select services/EC2/Launch Instance

3) Choose an Amazon Machine Image (AMI). I choose **Ubuntu Server 18.04 LTS (HVM), SSD Volume Type.**

4) Choose an Instance Type. I choose **t2.micro Free tier eligible.**

5) Configure Instance Details. You can tune a lot of configuration but if you want to connect EC2 you have to tune **Auto-assign Public IP=Enable.**

6)Add Storage i use default setting

7) Add Tags

8) Configure Security Group i use default setting and appear this warning

“Rules with source of 0.0.0.0/0 allow all IP addresses to access your instance. We recommend setting security group rules to allow access from known IP addresses only.”

İf you want change source ip.

9) Review Instance if everything look like good, you can select launch.

10)Create or select key pair.

11) To access your instance:

Open an SSH client. (find out how to connect using PuTTY)

Locate your private key file (awskey.pem). The wizard automatically detects the key you used to launch the instance.

Your key must not be publicly viewable for SSH to work. Use this command if needed:

“chmod 400 awskey.pem”

Connect to your instance using its Public DNS:

elasticip.compute-1.amazonaws.com

Example:

“ssh -i "awskey.pem" [ubuntu@ec2-eleaticip.compute-1.amazonaws.com](mailto:ubuntu@ec2-eleaticip.compute-1.amazonaws.com)”

Installing AWS Mavlink Router

1)Set up the AWS firewall

Open port 22 only to incoming/outgoing traffic from anywhere

Open port 8080 to traffic only from AWSIPADRESS, i.e. its self

2)Get the script to install and configure the AWS instance:

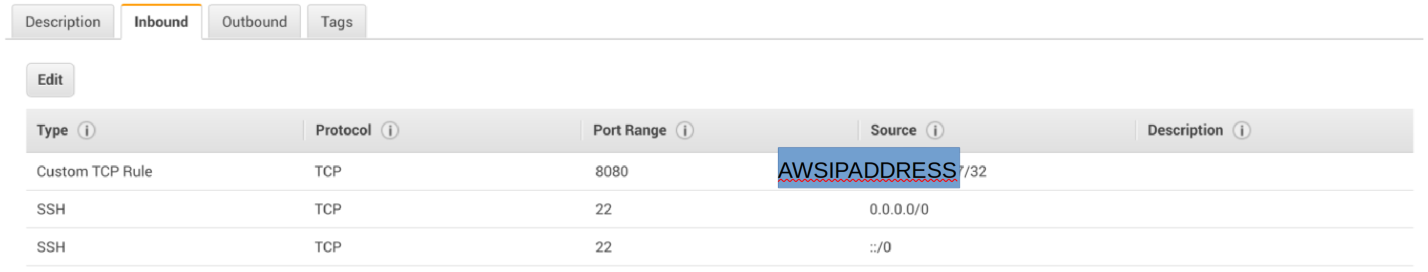
“wget https://github.com/swarmnect/FOToD/raw/master/AWSinstance/AWSbuild.sh”

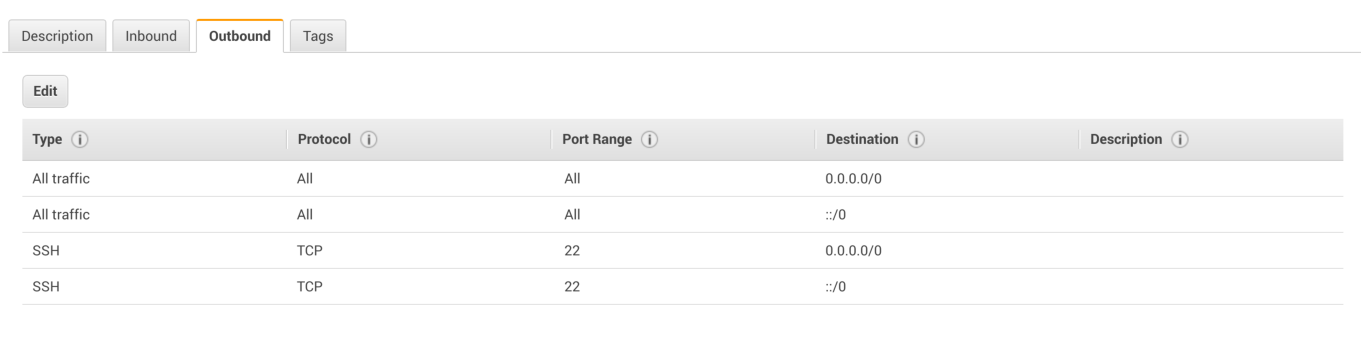
Run script:

sudo chmod 777 ~/AWSbuild.sh;

sudo ~/AWSbuild.sh | tee buildlog.txt

Done!





We need aws ip adress i use alastic ip

<https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/elastic-ip-addresses-eip.html>

**MAVLINK ROUTER BUILD**

Target hardware/prerequisites

* SD Card
* Raspberry Pi
* Omnibus F4 flight controller
* UART cable from flight controller to Pi

Installing Pi

1) Download and unzip the Raspberry Pi OS image from:

<https://www.raspberrypi.org/downloads/raspbian/>

2) Follow the instructions here to copy the image to an SD card and boot Raspberry Pi.

<https://www.raspberrypi.org/documentation/installation/installing-images/>

3) Configure Raspberry Pi.

Using this code “sudo raspi-config”

* Enable ssh log in
* Setup locale
* Setup keyboard
* Setup auto-log in
* Set serial options: serial login NO; serial enable YES
* Enable webcam

4) Check /etc/config.txt that this line appears: “enable\_uart=1” if it doesn’t appears add this line.

5) Reboot Pi.

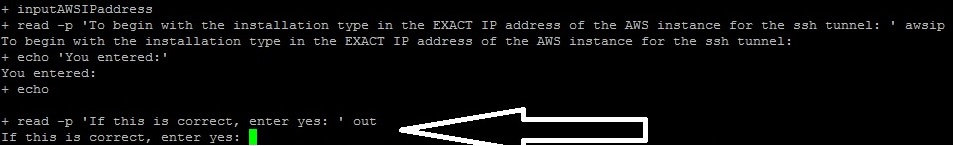
“sudo reboot”

6) Get the script to install and configure the Pi.

“wget <https://gitlab.com/pjbca/4guav/raw/master/MavlinkRouterBuild/MavlinkRouterBuild.sh>”

“sudo chmod 777 ~/MavlinkRouterBuild.sh”

“sudo ~/MavlinkRouterBuild.sh 2>&1 | tee MavlinkRouterBuildlog.txt”



End of the installation we have to write AWS ip.

I add my ip" elasticip.compute-1.amazonaws.com"

7) [Create a key pair](https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/ec2-key-pairs.html)

You need to set its permission carefully for AWS to accept it.

“sudo chmod 400 /home/pi/awskey.pem” Reboot Pi again. And Done.

**WEBCAM BUILD**

**Target hardware/prerequisites**

* SD Card
* Raspberry Pi
* Raspberry Pi camera

1) Get the script to install and configure the Pi:

“wget <https://github.com/swarmnect/FOToD/raw/master/webcam/installwebcam.sh>”

2)Run script to set up the files so webcam starts at boot and AWS ssh tunnel configuration:

sudo chmod 777 ~/installwebcam.sh;

sudo ~/installwebcam.sh 2>&1 | tee installwebcambuildlog.txt

End of the installation you have to write AWS ip.

3) Next, run the webcam installation process (use default settings unless you want something different):

sudo chmod 777 /home/pi/RPi\_Cam\_Web\_Interface/install.sh;

sudo /home/pi/RPi\_Cam\_Web\_Interface/install.sh 2>&1 | tee RPi\_Cam\_Web\_Interfacebuildlog.txt

4) You need to set its permission for pem file but I did it before.

“sudo chmod 400 /home/pi/awskey.pem”

Reboot Pi:

“sudo reboot”

5) Run it once manually and hit yes making sure it connects, then reboot again.

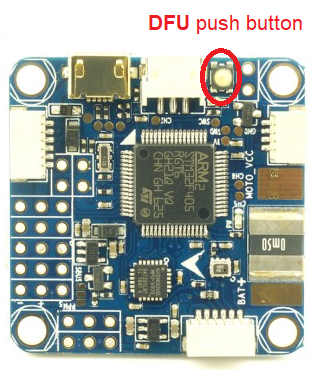
“sudo /home/pi/startupscripts/start\_sshtoAWSforWebCam.sh”

Once you enter yes and confirm it connects, just ctl-c out.

Reboot Pi again: “sudo reboot” Done.

**OMNIBUS F4 BUILD**

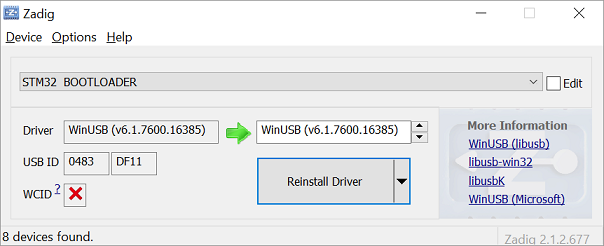
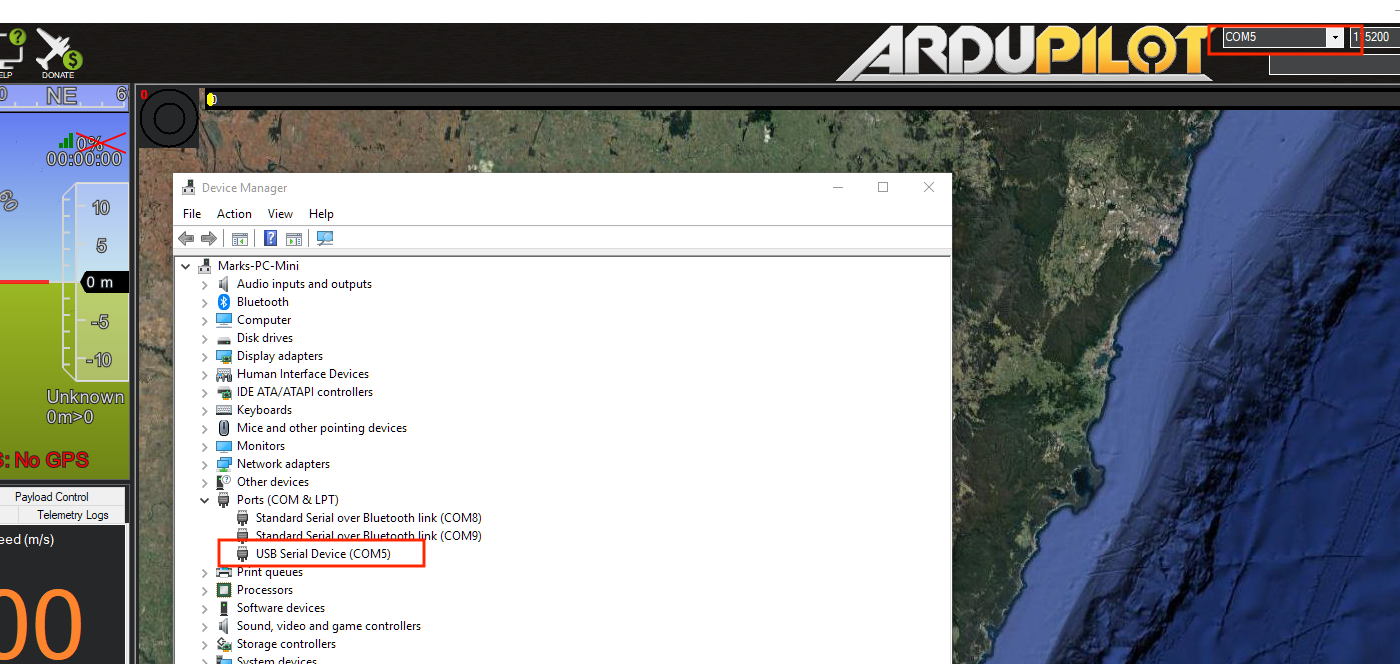
Start DFU mode To do this, you’ll need to locate the DFU jumper on your board. On most board flavours this is a little push button thas needs to be pressed while connecting your board to your PC via USB:



If you board is not recognize as a com port under windows, you have to download and install zadig : <http://zadig.akeo.ie/>

**DFU flashing under Windows - USB DFU:**

Make sure you have zadig if you're using Windows to enable the DFU driver. Instructions:

1. Download Zadig: <http://zadig.akeo.ie/>
2. Put device in DFU mode. If this is the first time to put Betaflight on you need to short the BL or BOOT pads (or press and hold the BOOT tactile button) while plugging the USB into the board.
3. Open Zadig.
4. Options > List All Devices
5. Click on the drop bown box and click the device listed STM32 BOOTLOADER 
6. In the box to the right of the green arrow, select WinUSB (v6.1.7600.16385)
7. Click Install Driver
8. After the install completes, restart your computer (you can cheat and ensure no browser is running - but it is not guaranteed to work). The board should stay in DFU mode - IF - usb power remains during the reboot. If not, execute step 2 again.
9. Open up the Betaflight configurator.
10. Go to firmware flasher, select "No reboot sequence"
11. On F4 targets disable "Full Chip Erase". Use the config reset in Configurator later. ([#200](https://github.com/betaflight/betaflight-configurator/issues/200) reports the issue.)
12. Load Firmware [Local]
13. Browse to and select the proper hex file. For me [omnibusf4v6\_bl.hex](https://firmware.ardupilot.org/Tools/Bootloaders/omnibusf4v6_bl.hex) https://firmware.ardupilot.org/Tools/Bootloaders/
14. Click flash firmware.
15. Open Mission Planner and go to the Initial Setup tab. Verify that the COM port in the top right is the same as in Device Manager.
16. Choose “Load Custom Firmware” and browse to the respective .apj file. After the flash is complete, power cycle the device. For me <https://firmware.ardupilot.org/Rover/>

<https://firmware.ardupilot.org/Rover/stable-4.0.0/omnibusf4v6/ardurover.apj>

