**Secure Remote Video Streaming Using Raspberry Pi and Tunneling Services**

Prerequisites:

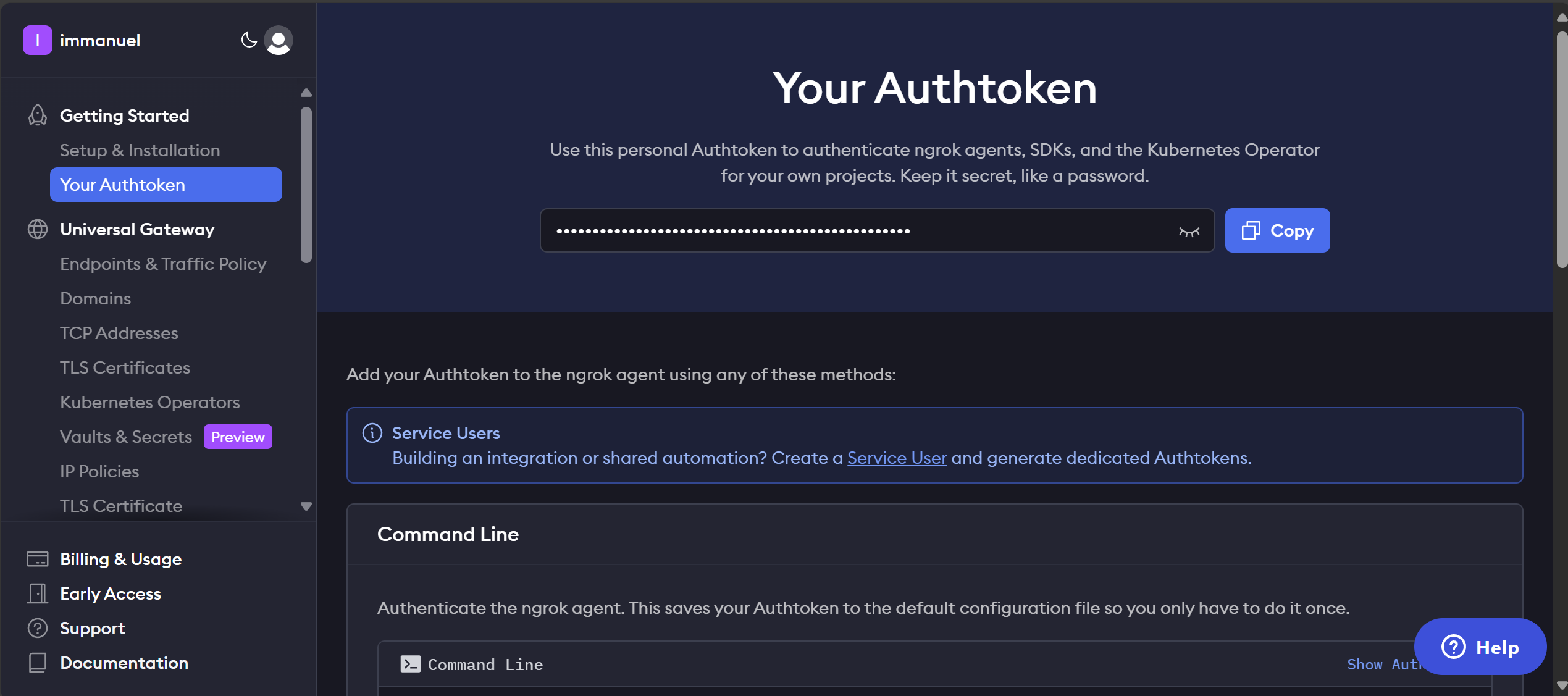
Hardware: 1. Raspberry Pi Zero 2 W with Wifi

2. Camera: IMX378 with ribbon cable

Software: 1. Raspberry pi OS (I used Raspberry Pi lite OS, 64 bit)

2. GUI or SSH access enabled for external display less setup

Account: 1. Create a free ngrok account (sign up at ngrok.com)

2. Copy the Authtoken

#1 Install and Configure Raspberry Pi OS & Camera

1. Flash the SD card (FAT32 formatted) with Raspberry pi OS using Raspberry PI image Software. (I used 16GB SD Card).
2. Power up the PI, connect to WIFI, and update the system

sudo apt update && sudo apt upgrade -y sudo reboot

1. Enable and configure the IMX378 camera ([Refer](https://docs.arducam.com/Raspberry-Pi-Camera/Native-camera/12MP-IMX378/))
2. Edit /boot/firmware/config.txt - sudo nano /boot/firmware/config.txt
3. Modify this line from camera\_auto\_detect=1 to camera\_auto\_detect=0
4. Add this line under [all]: dtoverlay=imx378
5. Save and Reboot

#2 Install Picamera2 and Dependencies

1. Install Picamera2 - sudo apt install -y python3-picamera2
2. Install simplejpeg (JPEG Encoder) - pip3 install simplejpeg

#3 Set Up the Local Streaming Server

1. Create the Python script for the MJPEG web server at your desired location:

nano stream\_server.py

1. Write Code for Simple Streaming

import io

import logging

import socketserver

from http import server

from threading import Condition

from picamera2 import Picamera2

from picamera2.encoders import JpegEncoder

from picamera2.outputs import FileOutput

PAGE = """\

<html>

<head>

<title>Pi Zero 2 W IMX378 Live Stream</title>

</head>

<body>

<h1>Live Video from Raspberry Pi Zero 2 W & IMX378</h1>

<img src="stream.mjpg" width="640" height="480" />

</body>

</html>

"""

class StreamingOutput(io.BufferedIOBase):

def \_\_init\_\_(self):

self.frame = None

self.condition = Condition()

def write(self, buf):

with self.condition:

self.frame = buf

self.condition.notify\_all()

class StreamingHandler(server.BaseHTTPRequestHandler):

def do\_GET(self):

if self.path == '/':

self.send\_response(301)

self.send\_header('Location', '/index.html')

self.end\_headers()

elif self.path == '/index.html':

content = PAGE.encode('utf-8')

self.send\_response(200)

self.send\_header('Content-Type', 'text/html')

self.send\_header('Content-Length', len(content))

self.end\_headers()

self.wfile.write(content)

elif self.path == '/stream.mjpg':

self.send\_response(200)

self.send\_header('Age', 0)

self.send\_header('Cache-Control', 'no-cache, private')

self.send\_header('Pragma', 'no-cache')

self.send\_header('Content-Type', 'multipart/x-mixed-replace; boundary=FRAME')

self.end\_headers()

try:

while True:

with output.condition:

output.condition.wait()

frame = output.frame

self.wfile.write(b'--FRAME\r\n')

self.send\_header('Content-Type', 'image/jpeg')

self.send\_header('Content-Length', len(frame))

self.end\_headers()

self.wfile.write(frame)

self.wfile.write(b'\r\n')

except Exception as e:

logging.warning(

'Removed streaming client %s: %s',

self.client\_address, str(e))

else:

self.send\_error(404)

self.end\_headers()

class StreamingServer(socketserver.ThreadingMixIn, server.HTTPServer):

allow\_reuse\_address = True

daemon\_threads = True

picam2 = Picamera2()

picam2.configure(picam2.create\_video\_configuration(main={"size": (640, 480)}))

output = StreamingOutput()

picam2.start\_recording(JpegEncoder(), FileOutput(output))

try:

address = ('', 7123)

server = StreamingServer(address, StreamingHandler)

server.serve\_forever()

finally:

picam2.stop\_recording()

1. Save and Run the Python file – python3 stream\_server.py
2. On another device on the same network, open a browser to <http://192.168.1.7:7123>

You should see the live stream.

Replace the 192.168.1.7 with your pi address (get Pi's IP with hostname -I)

#4 Stream to the Internet with grok

1. Open New SSH terminal, Download and install ngrok

wget https://bin.equinox.io/c/bNyj1mQVY4c/ngrok-v3-stable-linux-arm64.tgz

sudo tar xvzf ./ngrok-v3-stable-linux-arm64.tgz -C /usr/local/bin

rm ngrok-v3-stable-linux-arm64.tgz

1. Authenticate with your ngrok account

ngrok authtoken <your-authtoken>

1. Start the tunnel:

ngrok http 7123

1. Ngrok will display a public URL like this https://abc123.ngrok-free.app

#5 View Remotely

1. Open a browser and go to the ngrok URL
2. The live IMX378 feed will stream directly

Note:  
  
To change the authtoken:

1. Remove the old authtoken

rm -rf ~/.config/ngrok/ngrok.yml

1. Add you new authtoken

ngrok config add-authtoken <your\_new\_authtoken\_here>

⚠️if you’re using the ngroks free plan 🡪 It expires soon after crossing limited bandwidth