

AI Pyramid - Frigate NVR

Frigate は、リアルタイム AI 物体検出技術を用いて構築されたオープンソースのネットワークビデオレコーダー（NVR）システムです。すべてのビデオ処理は自身のハードウェア上でローカルに実行され、カメラのビデオストリームがネットワーク外に出ることはないため、データのプライバシーとセキュリティが確保されます。

1. リソースファイルの取得

以下のいずれかの方法で Frigate のデプロイリソースを取得してください。

方法 1：手動ダウンロード

Frigate [リソースリポジトリ](#) にアクセスしてダウンロードし、AI Pyramid デバイスにアップロードします。

方法 2：コマンドラインによるクローン

依存関係のチェック

システムに `git lfs` がインストールされていない場合は、[git lfs インストールガイド](#) を参考にインストールしてください。

```
git clone -b v0.17-ax650 https://huggingface.co/AXERA-TECH/frigate-resource
```

1.1 リソースファイルの説明

クローン完了後のディレクトリ構造は以下の通りです。

```
root@m5stack-AI-Pyramid:~/rsp/frigate-resource# ls -lh
total 4.2G
-rw-rw-r-- 1 axera axera 736 Jan 28 14:27 docker-compose.yml
-rw-rw-r-- 1 axera axera 4.2G Jan 28 14:33 frigate-ax650-990aee8.tar
-rw-rw-r-- 1 axera axera 3.6K Jan 28 14:27 README.md
```

2. Docker イメージのデプロイ

ステップ 1：イメージのインポート

```
docker load -i frigate-resource/frigate-ax650-990aee8.tar
```

注意

イメージファイルはバージョン更新により変更される可能性があるため、実際のディレクトリ内のファイル名に従ってインポートしてください。

ステップ 2：作業ディレクトリの準備

```
mkdir -p ~/frigate-runtime/{config,storage}
cp frigate-resource/docker-compose.yml ~/frigate-runtime/
```

中国大陸のユーザー

モデルファイルを正常に自動ダウンロードするために、docker-compose.yml 内の HuggingFace 環境変数の設定をコメントアウト解除する必要があります。

```
services:
  frigate:
    container_name: frigate
    privileged: true
    restart: unless-stopped
    stop_grace_period: 30s
    image: frigate:ax650-990aee8
    shm_size: "1024mb"
    volumes:
      - /soc/lib:/soc/lib
      - /usr/local/lib/libexif.so:/soc/lib/libexif.so
      - /etc/localtime:/etc/localtime:ro
      - ./config:/config
      - ./storage:/media/frigate
      - type: tmpfs
        target: /tmp/cache
        tmpfs:
          size: 1000000000
    ports:
      - "8971:8971"
      # - "5000:5000"
      - "8554:8554"
      - "8555:8555/tcp"
      - "8555:8555/udp"
    environment:
      TZ: "Asia/Shanghai"
      LD_LIBRARY_PATH: "/usr/local/lib:/usr/lib:/opt/lib:/soc/lib"
      HF_ENDPOINT: https://hf-mirror.com
```

"docker-compose.yml" 28L, 736B

28,7

All

ステップ 3：コンテナの起動

メモリ設定

4GB メモリ版の AI Pyramid については、プログラムを正常に動作させるために、起動前に [仮想メモリの追加チュートリアル](#) を参照してください。

```
cd ~/frigate-runtime/
docker compose up -d
```

ステップ 4：初期ログイン情報の取得

起動完了後、ログを確認して自動生成された初期ユーザー名とパスワードを取得します。

```
docker logs frigate
```

出力例:

```
root@m5stack-AI-Pyramid:~/frigate-runtime# docker logs frigate
2026-01-28 14:55:53.546981309 [2026-01-28 14:55:53] frigate.app INFO : *****
2026-01-28 14:55:53.548374976 [2026-01-28 14:55:53] frigate.app INFO : *****
2026-01-28 14:55:53.549670976 [2026-01-28 14:55:53] frigate.app INFO : *** Auth is
2026-01-28 14:55:53.550969226 [2026-01-28 14:55:53] frigate.app INFO : *** Created
2026-01-28 14:55:53.574785726 [2026-01-28 14:55:53] frigate.app INFO : *** User: a
2026-01-28 14:55:53.664755976 [2026-01-28 14:55:53] frigate.app INFO : *** Passwor
2026-01-28 14:55:53.734733393 [2026-01-28 14:55:53] frigate.app INFO : *****
2026-01-28 14:55:53.814732809 [2026-01-28 14:55:53] frigate.app INFO : *****
2026-01-28 14:55:53.904771809 [2026-01-28 14:55:53] frigate.app INFO : Starting FastA
```

3. 初期設定

3. 初期設定

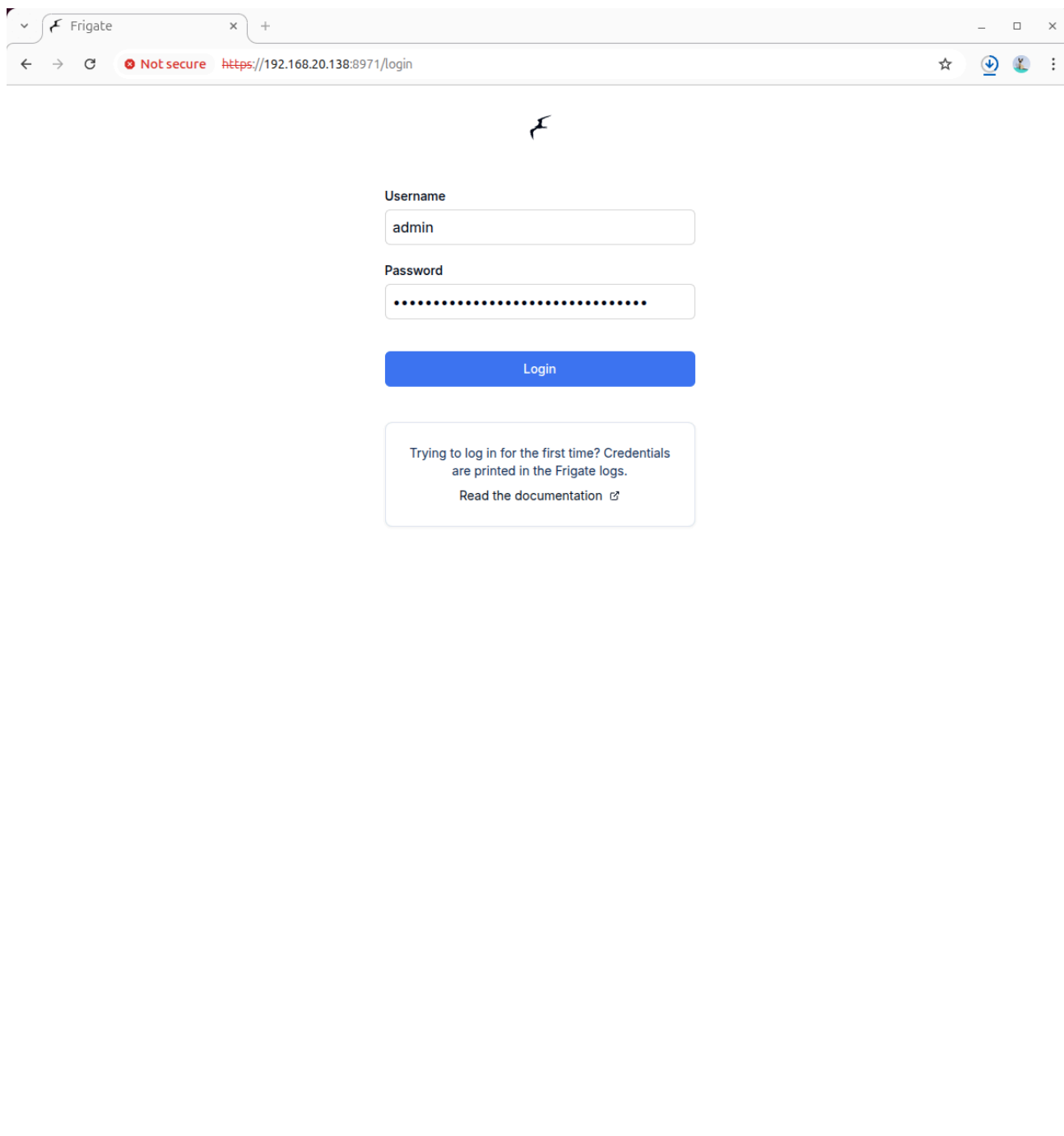
3.1 Web 管理画面へのアクセス

ブラウザを使用して Frigate 管理画面にアクセスします。

```
https://server_ip:8971
```

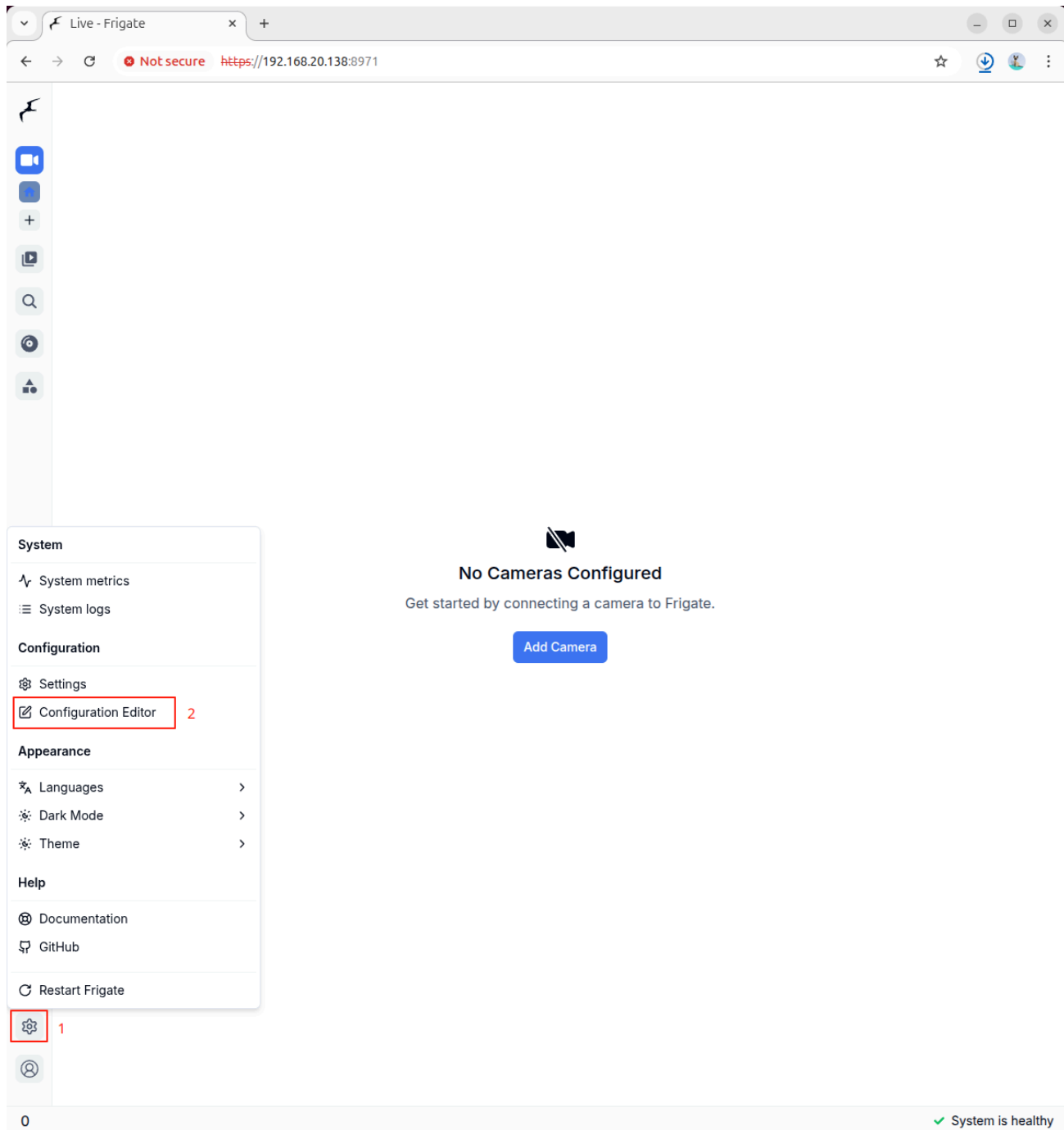
初回アクセス

前のステップで取得した `admin` ユーザー名とランダムに生成されたパスワードを使用してログインしてください。

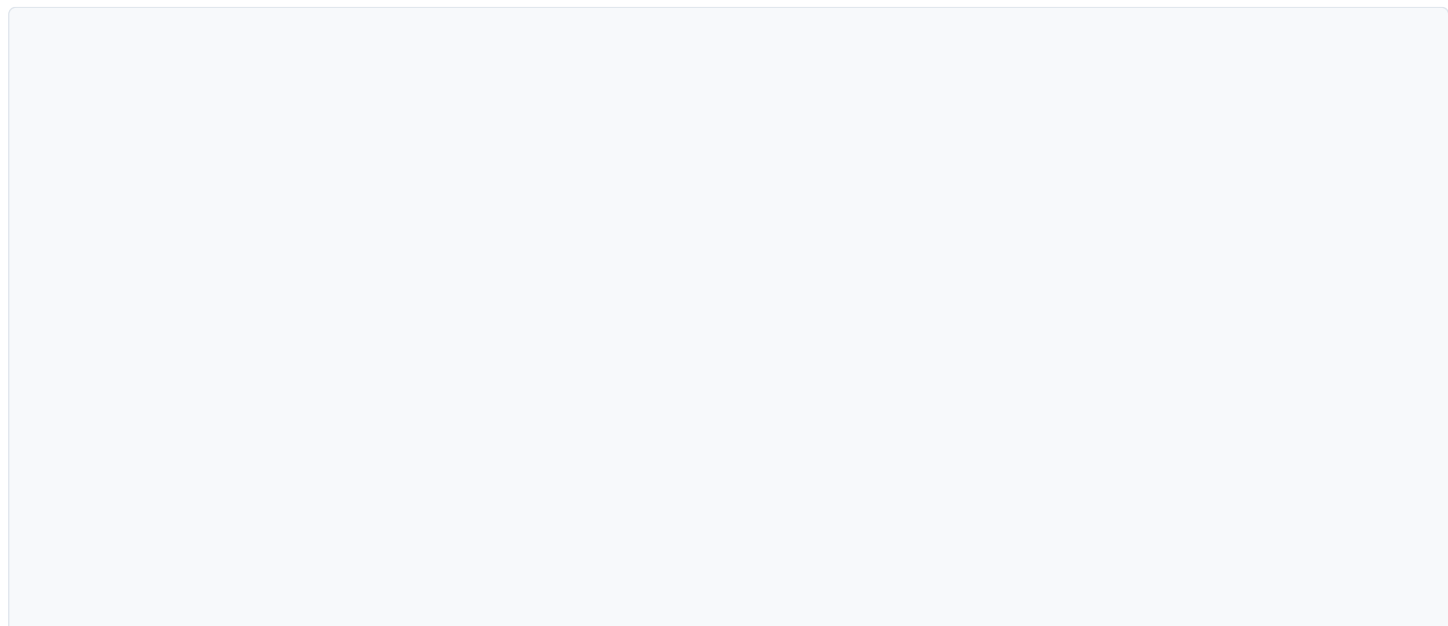


3.2 カメラパラメータの設定

画面左側の「設定」をクリックして設定画面に入り、設定ファイルを編集します。`go2rtc` セクションを変更し、ご自身の IP カメラのアドレスに置き換えて、以下の設定例を入力して保存してください。



設定例



```

mqtt:
  enabled: false
go2rtc:
  streams:
    # Main stream
    road1:
      - rtsp://192.168.20.57:8554/road1.264
    # Sub stream
    road1_sub:
      - rtsp://192.168.20.57:8554/road1_sub.264
cameras:
  road1:
    enabled: true
    ffmpeg:
      inputs:
        # Path of the recording stream, using the main stream configured in go2rtc
        # During debugging, a local media file can be used here
        - path: rtsp://127.0.0.1:8554/road1
          roles:
            - record
        # Path of the detection stream, using the sub stream configured in go2rtc
        # During debugging, a local media file can be used here
        - path: rtsp://127.0.0.1:8554/road1_sub
          roles:
            - detect
        # preset-axera-h264 is used to decode H.264 streams
        # preset-axera-h265 is used to decode H.265 streams
        # These two decoder presets are preferred
        hwaccel_args: preset-axera-h264
        # If the resolution of the detection sub stream is lower than the detection resolution,
        # you can reduce the detection resolution to match the stream
        # or use preset-axera-h264-compatible / preset-axera-h265-compatible as decoder presets

record:
  enabled: true

# Enable detection
# If detection width and height are not set, the native resolution of the detection stream is used
detect:
  enabled: true
  width: 576
  height: 320
  fps: 5

# Configure the detection engine to use axengine
detectors:
  axengine:
    type: axengine

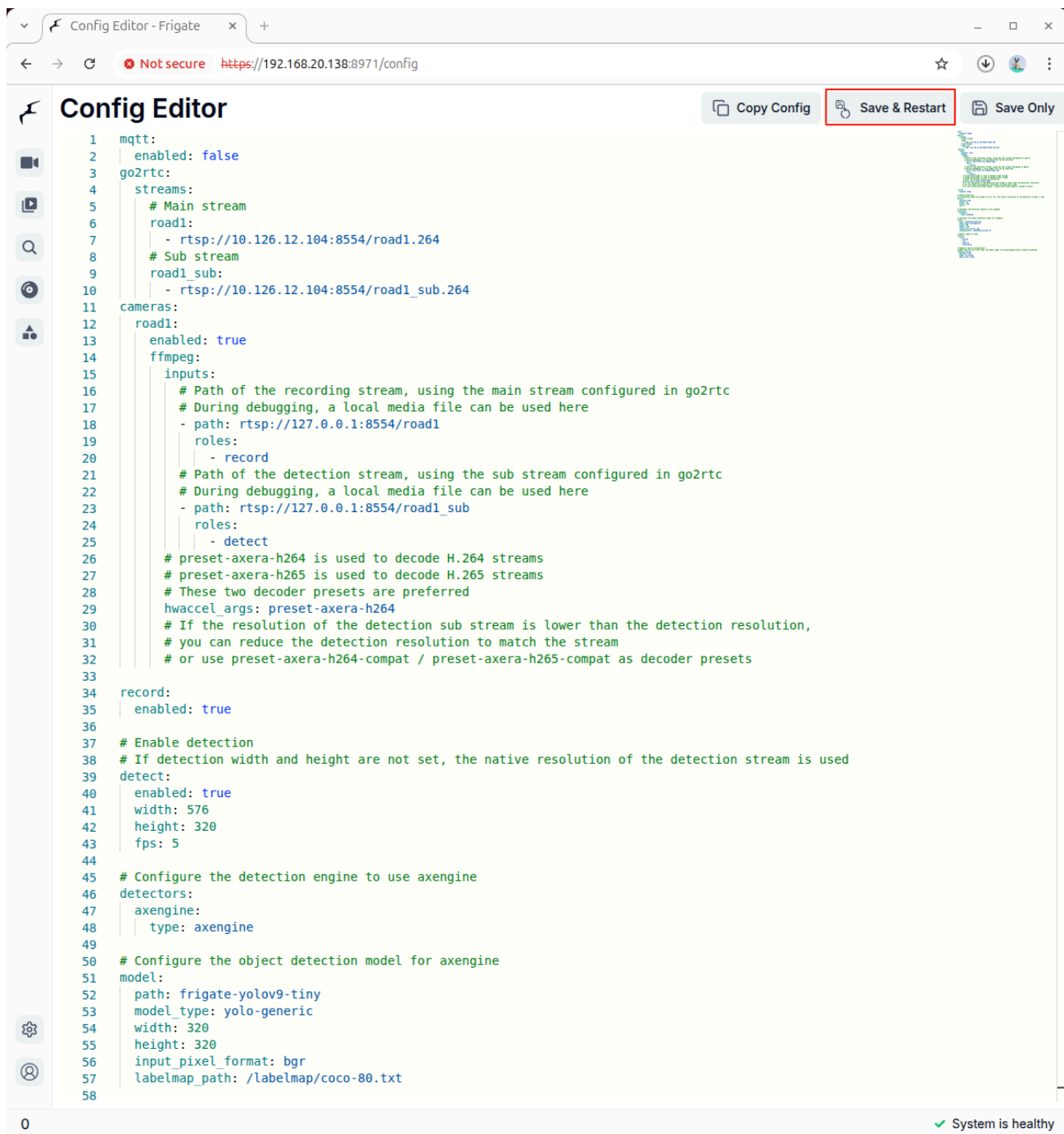
# Configure the object detection model for axengine
model:
  path: frigate-yolov9-tiny
  model_type: yolo-generic
  width: 320
  height: 320

```

```
input_pixel_format: bgr
labelmap_path: /labelmap/coco-80.txt

# Object types to track
objects:
  track:
    - person
    - car
    - bicycle
    - motorcycle

# Semantic search configuration
# When used for the first time, the model needs to be downloaded online, please be patient
semantic_search:
  enabled: true
  model: ax_jinav2
  model_size: large
```

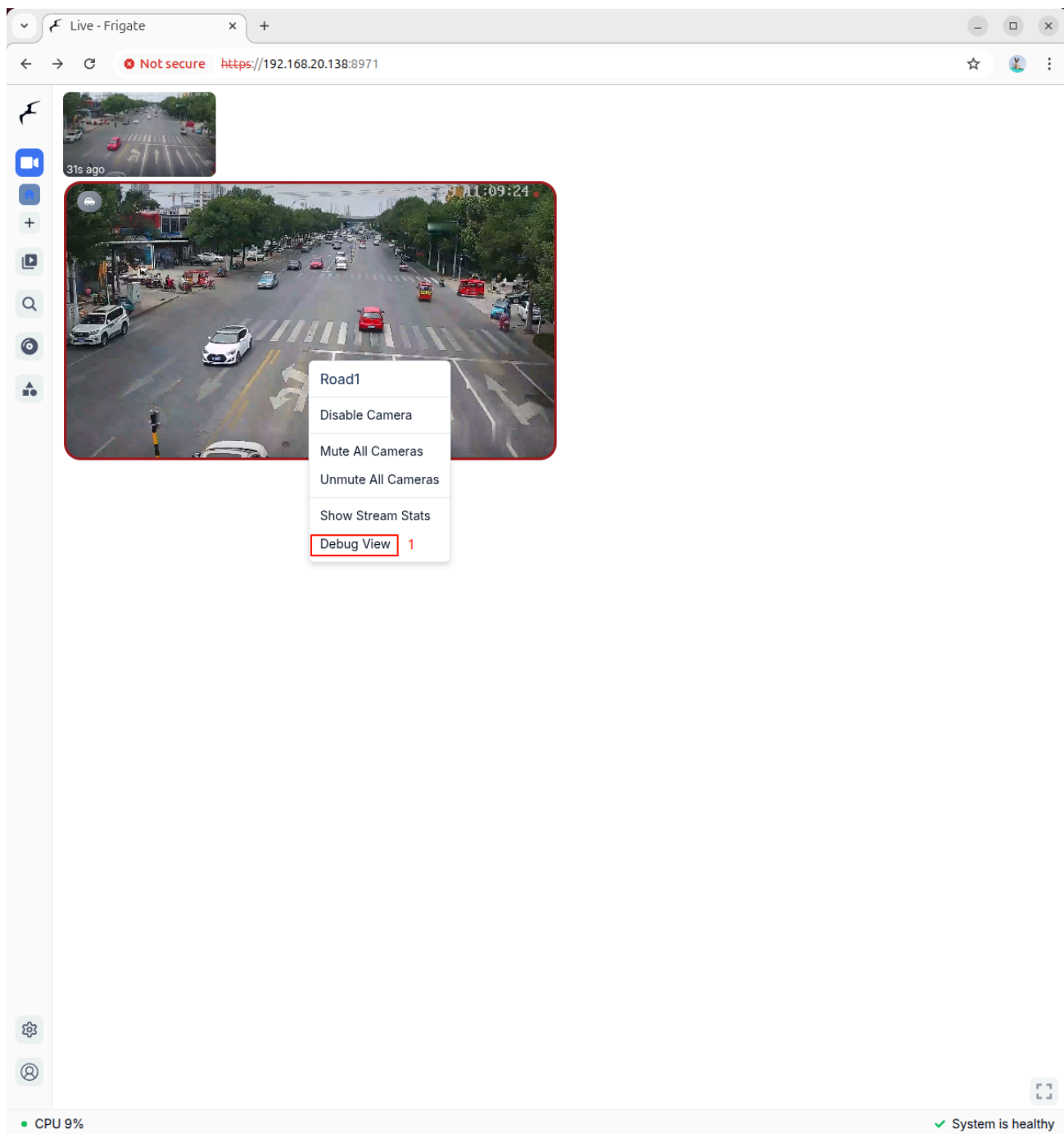


3.3 サービスの再起動と確認

設定を保存した後、Frigate コンテナを再起動して新しい設定を適用します。

```
docker restart frigate
```

デバッグプレビュー画面に入り、ビデオストリームを右クリックしてコンテキストメニューからデバッグプレビューモードを選択します。



設定画面で検出枠やその他の情報をプレビューできます。

Debug - Frigate

Not secure https://192.168.20.138:8971/#road1

Back History

Debug

Frigate uses your detectors (Axengine) to detect objects in your camera's video stream.

Debugging view shows a real-time view of tracked objects and their statistics. The object list shows a time-delayed summary of detected objects.

Debugging Object List

Bounding Boxes ⓘ
Show bounding boxes around tracked objects ☒

Timestamp
Overlay a timestamp on the image ☐

Zones
Show an outline of any defined zones ☐

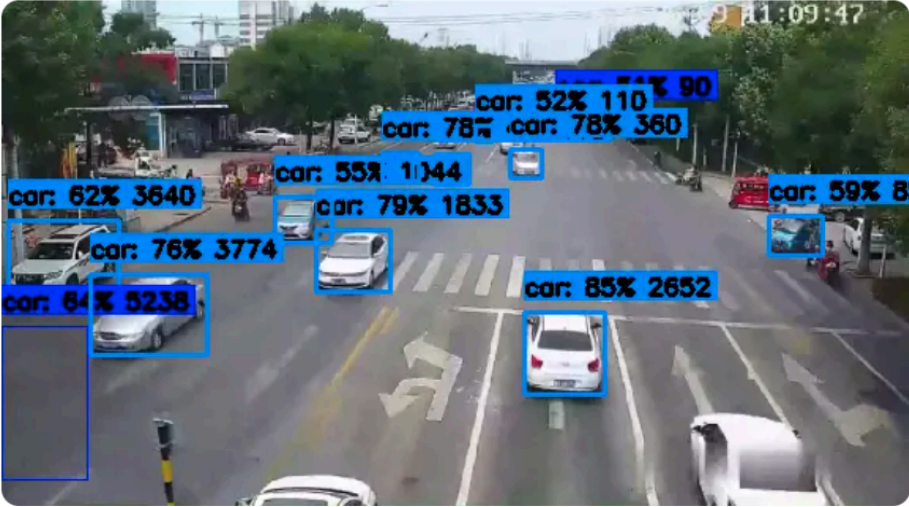
Motion Masks
Show motion mask polygons ☐

Motion Boxes ⓘ
Show boxes around areas where motion is detected ☐

Regions ⓘ
Show a box of the region of interest sent to the object detector ☐

Paths ⓘ
Show significant points of the tracked object's path ☐

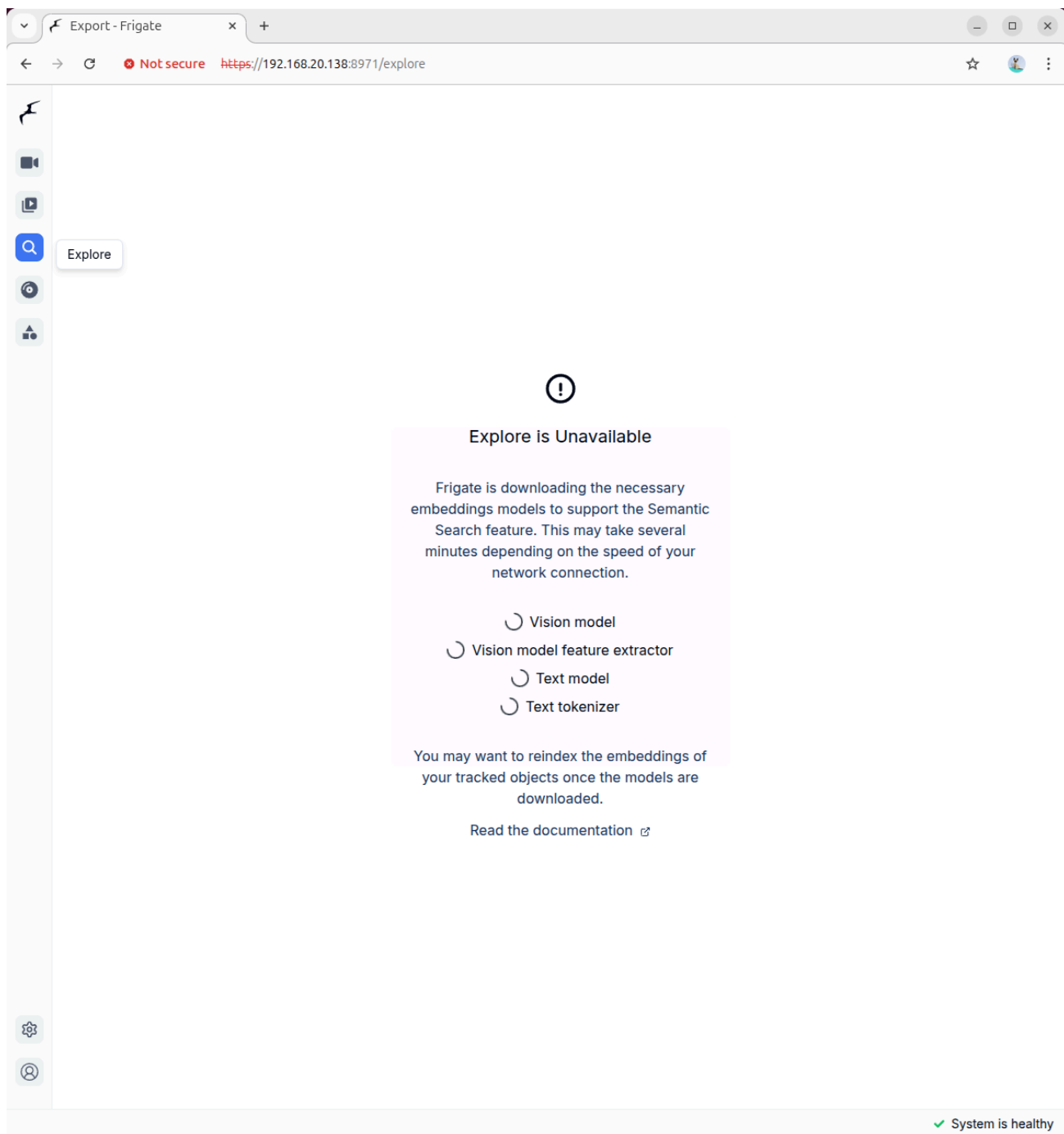
Object Shape Filter Drawing ⓘ
Draw a rectangle on the image to view area and ratio details ☐



CPU 15% System is healthy

4. セマンティック検索とブラウジング

「ブラウズ」ボタンをクリックしてメディアブラウズ画面に入ります。初回アクセス時はセマンティック検索モデルのダウンロードが必要であり、ファイルサイズが大きいため、しばらくお待ちください。



モデルの読み込みが完了した後、検索ボックスに検索したいキーワードを入力すると、システムが自動的に検索し、関連するビデオのスクリーンショットを返します。

Explore - Frigate

Not secure https://192.168.20.138:8971/explore?query=bus

bus

All CamerasAll LabelsAll DatesMore FiltersSortSettings

88%

2m ago
Jan 29, 9:50 AM

81%

50s ago
Jan 29, 9:51 AM

62%

2m ago
Jan 29, 9:50 AM

72%

2m ago
Jan 29, 9:49 AM

72%

1m ago
Jan 29, 9:50 AM

58%

2m ago
Jan 29, 9:49 AM

82%

2m ago
Jan 29, 9:49 AM

82%

59s ago
Jan 29, 9:51 AM

81%

1m ago
Jan 29, 9:50 AM

57%

2m ago
Jan 29, 9:50 AM

57%

45s ago
Jan 29, 9:51 AM

89%

1m ago
Jan 29, 9:50 AM

89%

1m ago
Jan 29, 9:50 AM

56%

1m ago
Jan 29, 9:50 AM

67%

1m ago
Jan 29, 9:50 AM

84%

1m ago
Jan 29, 9:50 AM

CPU 14%

System is healthy

Explore - Frigate

Not secure https://192.168.20.138:8971/explore?query=yellow+car

yellow car

All Cameras All Labels All Dates More Filters Sort Settings

2m ago Jan 29, 9:54 AM

1m ago Jan 29, 9:55 AM

Just now Jan 29, 9:57 AM

39s ago Jan 29, 9:56 AM

2m ago Jan 29, 9:54 AM

1m ago Jan 29, 9:55 AM

1m ago Jan 29, 9:55 AM

42s ago Jan 29, 9:56 AM

3m ago Jan 29, 9:54 AM

41s ago Jan 29, 9:56 AM

22s ago Jan 29, 9:57 AM

1m ago Jan 29, 9:55 AM

CPU 15%

System is healthy

詳細情報

より詳細な高度な設定オプションや機能の説明については、[Frigate 公式ドキュメント](#) を参照してください。