

الآن وقد أصبح "العقل المدبر" (ملف PHP) جاهزاً، نتقل إلى "قلب النظام" وهو سيرفر أوراكل كلاود (Oracle Cloud).
أول شيء يجب القيام به هو تجهيز بيئة السيرفر لاستقبال اليبث والتحكم به. إليك الخطوات بالترتيب الصحيح:

1. ضبط جدار الحماية (Ingress Rules)

هذه أهم خطواته؛ فيدونها لن يستطع تطبيق Flutter رؤية اليبث، ولن يستطع ملف PHP إرسال الروابط.

- اذهب إلى لوحة تحكم أوراكل -> **Virtual Cloud Networks** -> **Networking**.
- اختر الـ **VCN** الخاص بسيرفرك، ثم اضغط على **Security Lists**.
- أضف القواعد التالية (Ingress Rules):

الوظيفة	البروتوكول	المنفذ (Port)
لاستقبال أوامر الـ PATCH من ملف PHP	TCP	9997
لبث الـ WebRTC إلى تطبيق Flutter	TCP	8889
لبث الـ RTSP (إذا كنت تستخدمه)	TCP	8554
ضروري جداً لاستقرار فيديو WebRTC	UDP	10000-20000

2. تثبيت MediaMTX على السيرفر

إذا لم تكن قد ثبتته بعد، فهذه هي الطريقة الأسرع عبر الـ Terminal (SSH):

1. تحميل البرنامج:

```
Bash
envion/mediamtx/releases/download/v1.9.0/mediamtx_v1.9.0_linux_amd64.tar.gz
```

2. فك الضغط:

```
Bash
tar -xvzf mediamtx_v1.9.0_linux_amd64.tar.gz
```

3. تعديل إعدادات (mediamtx.yml) MediaMTX

افتح الملف وقم بتفعيل الخيارات التي تجعله "يسمع" لملف الـ PHP الخاص بك:

- ابحث عن `api: no` واجعلها `api: yes`.
- ابحث عن `apiAddress: 127.0.0.1:9997` واجعلها `apiAddress: :9997` (لكي يستقبل من IP الاستضافة الخارجية).
- في قسم الـ `paths` تأكد من إضافة سطر الـ `runOnDemand` الذي اتفقنا عليه:

```
paths:
  all:
    runOnDemand: php /path/to/your/script.php $RTSP_PATH
```

ملاحظة ذكية جداً، ولكن الإجابة هي (لا)، لا تضع IP الاستضافة بشكل محدد هناك.

الصحيح هو وضع النقطتين الرأسيتين فقط: `9997`.

لماذا تضع `9997` بدلاً من IP الاستضافة؟

1. المعنى التقني: في لغة السيرفرات، كتابة `9997` (بدون IP قبله) تعني "استقبل الطلبات من أي مكان" (0.0.0.0).
2. لماذا لا تضع IP الاستضافة؟ لأن الـ `apiAddress` في ملف الإعدادات يحدد للسيرفر "على أي واجهة محلية يستمع" وليس "من أي جهاز يستقبل". إذا وضعت IP خارجي هناك، قد يفشل السيرفر في التشغيل لأنه لا يملك هذا الـ IP برمجياً على كرت الشبكة الخاص به.
3. الأمان: إذا كنت قلقاً من أن يستطيع أي شخص الوصول للـ API، فالتحكم في "من يُسمح له بالدخول" يتم عن طريق جدار الحماية (Firewall) في أوراكل، وليس من داخل ملف `mediamtx.yml`.

- التفسير: عندما يطلب التطبيق مساراً، يقوم MediaMTX بفتح "نافذة أوامر" صغيرة داخل السيرفر وينفذ أمر `curl`. هذا الأمر يطرق باب رابط الـ PHP الخاص بك، فصيقظ الـ PHP، وتنفذ الكود، ثم تعود بطلب `PATCH` إلى أوراكن لتحديث الرابط.

نقطة قنية هامة (لضمان التحديث التلقائي):

لكي تضمن أن MediaMTX لا يحاول استخدام الرابط "القديم" الفاشل لفترة طويلة، يجب التأكد من ضبط الإعدادات التالية في `mediamtx.yml`:

- `sourceOnDemand: yes`: لكي لا يبدأ الاتصال إلا عندما يطلبه شخص ما.
- `sourceOnDemandCloseAfter: 10s`: لكي يقطع الاتصال فوراً بعد خروج المستخدمين، مما يجبر النظام على طلب بيانات جديدة من PHP في المرة القادمة.

2. كيف يتعامل النظام مع الروابط القديمة أو تغيير كلمة المرور؟

هذا يأتي دور ميزة `sourceOnDemand` التي وضعناها في الكود، مع خاصية الـ `Timeout` في MediaMTX.

السيناريو كالتالي:

1. في الحالة الطبيعية: إذا لم يستخدم أحد الكاميرا لمدة (مثلاً 10 ثوانٍ)، يقوم MediaMTX بحذف المسار تماماً من ذاكرته (`Unpublish`).
2. عند الطلب الجديد: في كل مرة يفتح فيها شخص ما الكاميرا بعد فترة انقطاع، يُعتبر المسار "جديداً" أو "فارغاً"، فيقوم MediaMTX بإعادة تشغيل ملف PHP.
3. تحديث البيانات: ملف PHP في كل مرة "يسيقظ" فيها، يذهب لقاعدة البيانات ويجلب أحدث كلمة مرور وأحدث IP.

ملخص الإجابة:

- الوصول لـ PHP: يتم عبر أمر `curl` نضعه في إعدادات أوراكن، يعمل كرسول يخبر الـ PHP بأن هناك من يطلب الكاميرا.
- الروابط القديمة: النظام لا يحفظ الروابط للأبد؛ هو "ينساها" بمجرد خروج المستخدمين، و"يجدها" من قاعدة البيانات عبر PHP في كل مرة يبدأ فيها مستخدم جديد بالمشاهدة.

هذا التصميم يضمن أنك لو غيرت IP الكاميرا أو كلمة مرورها في قاعدة البيانات الآن، فإن المستخدم القادم سيُشاهد البث بالبيانات الجديدة فوراً دون الحاجة لإعادة تشغيل سيرفر أوراكن.

4. تشغيل الخدمة (Service)

بما أننا شرحنا كيفية عمل الـ **Systemd Service** سابقاً، قم بتفعيلها الآن لضمان أن السيرفر يعمل بشكل دائم:



Bash

```
sudo systemctl enable mediamtx
sudo systemctl start mediamtx
```

كيف نتأكد أن أول خطوة نجحت؟

بمجرد الانتهاء من هذه الخطوات، قم بزيارة هذا الرابط من متصفحك الشخصي:

```
http://[ORACLE_IP]:9997/v3/config/paths/list
```

- إذا ظهرت لك صفحة فيها بيانات **JSON**: مبروك! السيرفر جاهز تماماً لاستقبال الأوامر.
- إذا لم يظهر شيء: راجع إعدادات جدار الحماية (الخطوة 1).

بمجرد التأكد من فتح الـ **API**، سنتنقل فوراً لاختبار "الحقن" من الـ **PHP** إلى أوراكل. هل قمت بفتح المتصفح في لوحة أوراكل؟

إعداد ملف الخدمة (Service) هو الخطوة التي ستجعل سيرفرك يعمل بشكل احترافي ومستقل. يتم إنشاء هذا الملف في نظام التشغيل Linux (أوراكل كاتود غالباً يستخدم Ubuntu أو Oracle Linux) لإداره البردمج كخلفية.

1. أين يتم إنشاؤه؟

يتم إنشاء الملف في المسار المخصص لخدمات النظام:

```
etc/systemd/system/mediamtx.service/
```

2. ماذا يحتوي؟

سأعطيك المحتوى الجاهز. افتح الملف باستخدام الأمر التالي:

```
❏
```

Bash

```
sudo nano /etc/systemd/system/mediamtx.service
```

ثم انسخ وألصق النص التالي (مع التأكد من تعديل المسارات إذا كان مجلد البردمج في مكان مختلف):

```
❏
```

Ini, TOML

```
[Unit]
Description=MediaMTX Service
After=network.target

[Service]
# ولف الإعدادات mediamtx المسار الذي يحتوي على ملف
WorkingDirectory=/home/ubuntu/mediamtx
# أمر التشغيل (المسار الكامل للبرنامج ثم المسار الكامل للإعدادات)
ExecStart=/home/ubuntu/mediamtx/mediamtx /home/ubuntu/mediamtx/mediamtx.yml

# إعادة التشغيل تلقائياً في حال حدوث أي خلل
Restart=on-failure
RestartSec=5

# توجيه المخرجات والأخطاء إلى سجل النظام
StandardOutput=syslog
StandardError=syslog
SyslogIdentifier=mediamtx

[Install]
WantedBy=multi-user.target
```

```
[Unit]
Description=MediaMTX Service
After=network.target

[Service]
# و ملف الإعدادات mediamtx المسار الذي يحتوي على ملف
WorkingDirectory=/home/ubuntu/mediamtx
# أمر التشغيل (المسار الكامل للبرنامج ثم المسار الكامل للإعدادات)
ExecStart=/home/ubuntu/mediamtx/mediamtx /home/ubuntu/mediamtx/mediamtx.yml
# إعادة التشغيل تلقائياً في حال حدوث أي خلل
Restart=on-failure
RestartSec=5
# توجيه المخرجات والأخطاء إلى سجل النظام
StandardOutput=syslog
StandardError=syslog
SyslogIdentifier=mediamtx

[Install]
WantedBy=multi-user.target
```

3. كيف يتم تشغيله؟

بعد حفظ الملف (Ctrl+O ثم Enter ثم Ctrl+X)، يجب تنفيذ الأوامر التالية بالترتيب لتفعيل الخدمة:

- تحديث قائمة الخدمات: ليُعرف النظام على الملف الجديد.



Bash

```
sudo systemctl daemon-reload
```

- تفعيل الخدمة عند الإقلاع: ليحمل البرنامج تلقائياً إذا أُعدت تشغيل السيرفر.



Bash

```
sudo systemctl enable mediamtx
```

- بدء التشغيل الآن:



Bash

```
sudo systemctl start mediامتخ
```

كيف تتأكد أن كل شيء يعمل؟

لمعرفة حالة الخدمة وما إذا كانت تعمل بنجاح (Active)، استخدم هذا الأمر:



Bash

```
sudo systemctl status mediامتخ
```

ملاحظات هامة:

1. الصلاحيات: تأكد أن ملف `mediامتخ` لديه صلاحية التنفيذ. إذا لم تكن متأكدًا، نفذ: `chmod +x`

`./home/ubuntu/mediامتخ/mediامتخ`

2. المسارات: في ملف الخدمة أعلاه، افترضت أن البرنامج في مجلد `home/ubuntu/mediامتخ/`. إذا كان في مكان آخر

(مثل `usr/local/bin/`)، تأكد من تغيير المسار في `WorkingDirectory` و `ExecStart`.

بمجرد تشغيل الخدمة وظهور الحالة باللون الأخضر (`active running`)، يكون سيرفر أوراكل جاهزاً لاستقبال أول "حقيقة" بث من

ملف `PHP` الخاص بك! هل ظهرت الحالة `Active`؟