

KYRONEX

NIGHT SCHEDULER

Guide d'utilisation complet

Comment faire travailler ton Jetson pendant que tu dors



By Manix (Emmanuel Gelinne) · KYRONEX A.I. · Jetson Orin Nano Super
2026 · Elastic License 2.0

C'EST QUOI LE NIGHT SCHEDULER ?

1 L'IDEE EN UNE PHRASE

Le Night Scheduler fait travailler ton Jetson tout seul pendant la nuit.
Il lance Claude Code automatiquement a une heure que tu choisis, et Claude améliore l'interface web de KYRONEX sans que tu aies a faire quoi que ce soit.

2 COMMENT CA MARCHE (version enfant de 8 ans)

Tu configures une "fenetre" : par exemple de 22h a 6h du matin.
Quand l'horloge arrive a 22h, le Jetson se reveille tout seul.
Il lance Claude Code N fois (ex: 10 fois).
Chaque fois, Claude lit la page web de KYRONEX et y fait une petite amelioration.
A chaque etape, une sauvegarde est faite. Si Claude fait une erreur -> retour arriere auto.
Le matin, tu trouves une interface un peu mieux qu'avant !

3 CE QU'IL AMELIORE

Il améliore UNIQUEMENT l'interface web locale :
-> Le fichier : /home/kitt/kitt-ai/static/index.html
-> Ta page KYRONEX : <https://192.168.1.4:3000>

Il ne touche PAS :
-> Ton site GitHub (on3egs.github.io/manix-kitt)
-> Le serveur Python (`kyronex_server.py`)
-> Le modele IA (Qwen 2.5)

Exemples de ce qu'il peut ameliorer :
-> Animations plus fluides
-> Meilleure lisibilite des messages
-> Scanner KI2000 plus realiste
-> Bugs CSS corriges
-> Optimisations mobile

4 LES 3 PIECES DU SYSTEME

`kitt_night_improve.sh` -> le script qui lance Claude Code en boucle
`kitt_scheduler.py` -> le daemon qui surveille l'heure
Bouton NIGHT dans l'interface web -> l'interface graphique pour tout controler

UTILISER LE PANNEAU WEB

1 OUVRIR LE PANNEAU

1. Ouvre l'interface KYRONEX dans ton navigateur :
`https://192.168.1.4:3000`
2. Dans le header en haut a droite, clique sur le bouton :
[NIGHT]
3. Le panneau vert MU-TH-UR s'ouvre.

2 CONFIGURER UNE FENETRE NOCTURNE

Dans la section "AJOUTER UNE FENETRE" :

DEBUT : heure de debut (ex: 22 h 00)

FIN : heure de fin (ex: 6 h 00)

ITERATIONS : combien de fois Claude ameliore (ex: 10)

JOURS : quels jours de la semaine (cliquer pour selectionner/deselectionner)

NOM : un nom pour reconnaitre la fenetre (ex: "Nuit semaine")

Puis clique sur [+ PROGRAMMER].

3 DEMARRER LE DAEMON

Apres avoir ajoute ta fenetre, clique sur [START].

Le badge passe de 0 ARRETE a * ACTIF PID:XXXX

Le daemon surveille l'heure toutes les minutes.

Il se lance automatiquement quand on entre dans la fenetre horaire.

4 LANCER MAINTENANT (sans attendre la nuit)

Dans la section "LANCER MAINTENANT" :

1. Tape le nombre d'iterations (ex: 3)
2. Clique sur [> GO]
3. Attends 2-3 minutes par iteration
4. Clique sur [RAFRAICHIR] pour voir les logs

IMPORTANT : Chaque iteration prend environ 2 a 3 minutes.

10 iterations = environ 30 minutes de travail.

Ne pas eteindre le Jetson pendant que ca tourne !

5 GERER LES FENETRES

[ON/OFF] -> activer ou desactiver une fenetre sans la supprimer

[X] -> supprimer definitivement une fenetre

[STOP] -> arreter le daemon (les fenetres sont conservees)

COMMANDES SSH (terminal)

Si tu preferes utiliser le terminal SSH plutot que l'interface web, voici toutes les commandes utiles.

1 CONNEXION SSH AU JETSON

```
# Depuis Windows (PowerShell ou PuTTY) :  
ssh kitt@192.168.1.4  
  
# Mot de passe sudo : 5505
```

2 DEMARRER KYRONEX

```
cd /home/kitt/kitt-ai  
bash start_kyronex.sh
```

3 LANCER DES AMELIORATIONS MAINTENANT (ligne de commande)

```
cd /home/kitt/kitt-ai  
  
# 1 iteration (test rapide, ~2-3 min) :  
bash kitt_night_improve.sh 1  
  
# 10 iterations (session complete, ~30 min) :  
bash kitt_night_improve.sh 10
```

4 DEMARRER LE DAEMON MANUELLEMENT

```
cd /home/kitt/kitt-ai  
python3 kitt_scheduler.py --daemon &  
echo $! > kitt_scheduler.pid  
  
# Verifier qu'il tourne :  
cat kitt_scheduler.pid  
ps aux | grep kitt_scheduler
```

5 ARRETER LE DAEMON

```
# Via l'API web :  
curl -sk -X POST https://localhost:3000/api/scheduler/stop  
  
# Ou manuellement :  
kill $(cat /home/kitt/kitt-ai/kitt_scheduler.pid)
```

6 VOIR LES LOGS EN TEMPS REEL

```
# Logs du daemon :  
tail -f /tmp/kitt_scheduler.log  
  
# Logs du dernier GO (remplacer XXXX par le timestamp) :  
tail -f /tmp/kitt_now_XXXX.log  
  
# Lister tous les logs de nuit :  
ls -lt /tmp/kitt_night_*.log | head -5
```

FICHIERS ET DOSSIERS IMPORTANTS

1 FICHIERS DU NIGHT SCHEDULER

```
/home/kitt/kitt-ai/

kitt_night_improve.sh  <- Le script principal (lance Claude)
kitt_scheduler.py      <- Le daemon planificateur
kitt_schedule.json     <- Tes fenetres configurees (cree auto)
kitt_scheduler.pid     <- PID du daemon (cree auto)

/tmp/

kitt_scheduler.log     <- Log du daemon
kitt_now_XXXX.log      <- Log d'un GO immediat
kitt_night_XXXX.log    <- Log d'une session nocturne
```

2 VERSIONS SAUVEGARDEES

Apres chaque iteration reussie, une version est sauvegardee ici :

```
/home/kitt/kitt-ai/static/versions/

v00_avant_session_XXXX.html  <- Etat avant la session
v01_04h22_Analyse_le_xxx.html <- Version apres iteration 1
v02_04h25_Analyse_le_xxx.html <- Version apres iteration 2
...
```

Si une amelioration casse quelque chose :

-> Le fichier est automatiquement restaure a la version precedente.

-> Tu peux aussi revenir manuellement :

```
# Revenir a une version specifique :
cp /home/kitt/kitt-ai/static/versions/v00_avant_session_XXX.html \
  /home/kitt/kitt-ai/static/index.html

# Puis redemarrer KYRONEX :
cd /home/kitt/kitt-ai && bash start_kyronex.sh
```

3 VOIR CE QUI A ETE FAIT

```
# Comparer avant/apres une session :
diff /home/kitt/kitt-ai/static/versions/v00_avant_session_XXX.html \
  /home/kitt/kitt-ai/static/versions/v10_05h00_xxx.html

# Lister toutes les versions (plus recentes en premier) :
ls -lt /home/kitt/kitt-ai/static/versions/*.html | head -20
```

ROUTES API ET RESUME

1 ROUTES API (pour les curieux)

Toutes ces routes sont disponibles sur <https://192.168.1.4:3000>

```
GET    /api/scheduler/status      <- Etat daemon + fenetres
POST   /api/scheduler/start      <- Demarrer le daemon
POST   /api/scheduler/stop    <- Arreter le daemon
POST   /api/scheduler/window <- Ajouter une fenetre
POST   /api/scheduler/window/{id}/toggle <- Activer/desactiver
DELETE /api/scheduler/window/{id} <- Supprimer
POST   /api/scheduler/run-now <- GO (lancer maintenant)
GET    /api/scheduler/logs      <- Voir les logs
```

Exemple avec curl (depuis le Jetson) :

```
# Voir le statut :
curl -sk https://localhost:3000/api/scheduler/status

# Ajouter une fenetre 22h->6h, 10 iterations, tous les jours :
curl -sk -X POST https://localhost:3000/api/scheduler/window \
  -H 'Content-Type: application/json' \
  -d '{"name":"Nuit auto","start_h":22,"end_h":6,"iterations":10,
      "days":[0,1,2,3,4,5,6]}'

# Lancer 3 iterations maintenant :
curl -sk -X POST https://localhost:3000/api/scheduler/run-now \
  -H 'Content-Type: application/json' \
  -d '{"iterations": 3}'
```

2 RESUME EN 5 ETAPES

1. Ouvre <https://192.168.1.4:3000>
2. Clique NIGHT -> Ajoute une fenetre (22h->6h, 10x, tous les jours)
3. Clique START -> le daemon est actif
4. Eteins ton ecran. La nuit, le Jetson travaille.
5. Le matin, l'interface est amelioree. Les versions sont dans </static/versions/>

CONSEIL : Commence par tester avec 1 iteration via le bouton GO.
Regarde les logs. Si tout va bien, programme une fenetre nocturne.
Commence avec 5 iterations, pas 50 !

ATTENTION : Le Jetson consomme beaucoup de RAM pendant les ameliorations.
Ne pas utiliser le chatbot KYRONEX pendant qu'une session est en cours.
Le Night Scheduler est fait pour tourner quand tu dors.