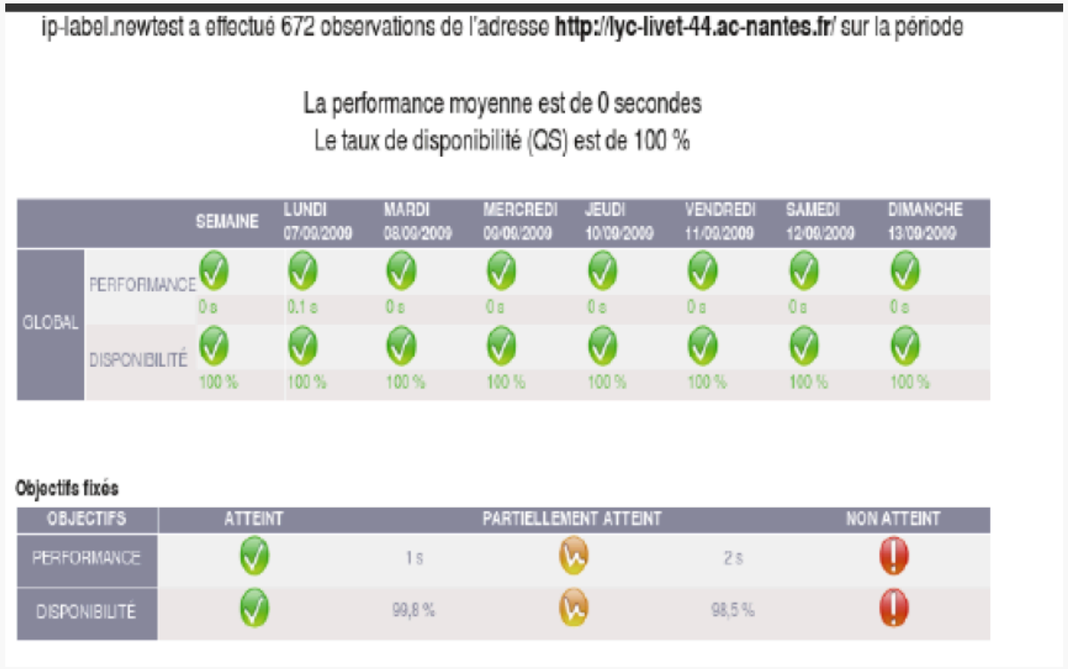
PROJET DATAMETRIE

PRÉSENTÉE PAR
Raphaël Gillet et
Jehannin Pier-André

Sommaire

Présentation et objectif du projet

L’objectif est d’envoyer au client un rapport d’état de son site web toutes les 24h via email



ip-label.newtest a effectué 672 observations de l'adresse **http://lyc-livet-44.ac-nantes.fr/** sur la période

La performance moyenne est de 0 secondes
Le taux de disponibilité (QS) est de 100 %

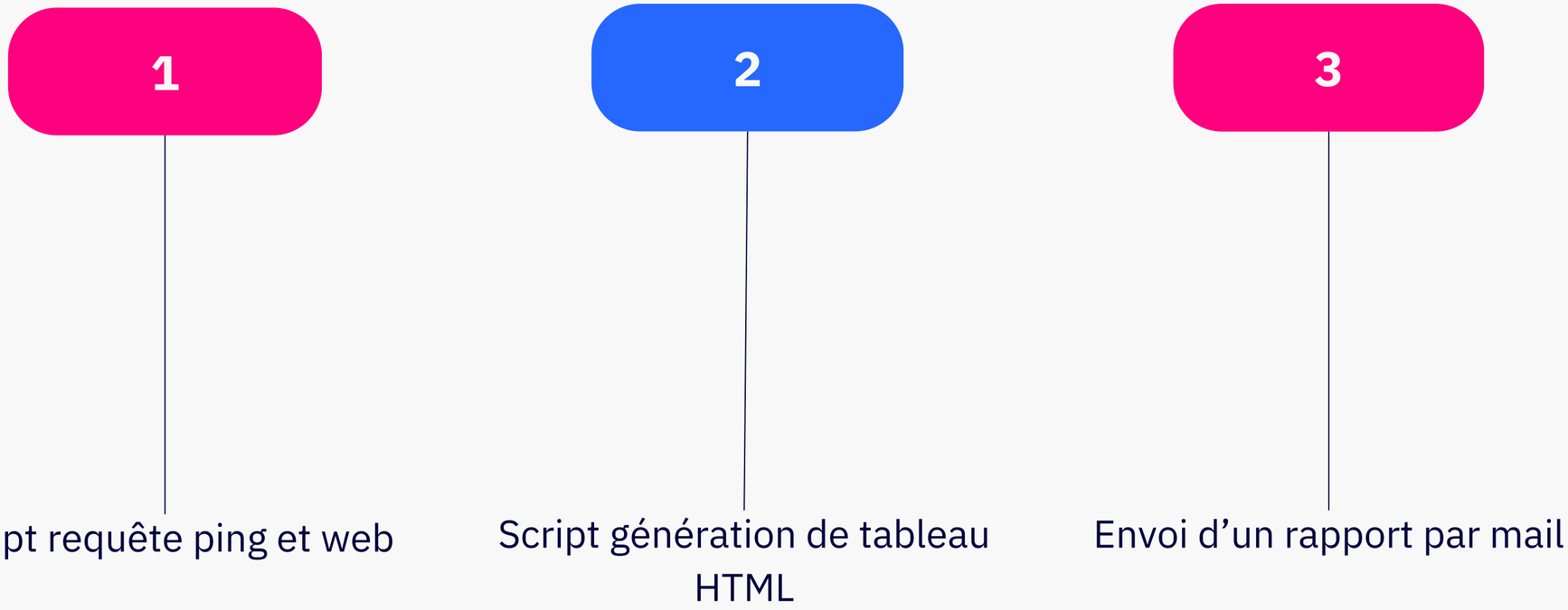
		SEMAINE	LUNDI 07/09/2009	MARDI 08/09/2009	MERCREDI 09/09/2009	JEUDI 10/09/2009	VENDREDI 11/09/2009	SAMEDI 12/09/2009	DIMANCHE 13/09/2009
GLOBAL	PERFORMANCE	✓	✓	✓	✓	✓	✓	✓	✓
		0 s	0.1 s	0 s	0 s	0 s	0 s	0 s	0 s
	DISPONIBILITÉ	✓	✓	✓	✓	✓	✓	✓	✓
		100 %	100 %	100 %	100 %	100 %	100 %	100 %	100 %

Objectifs fixés

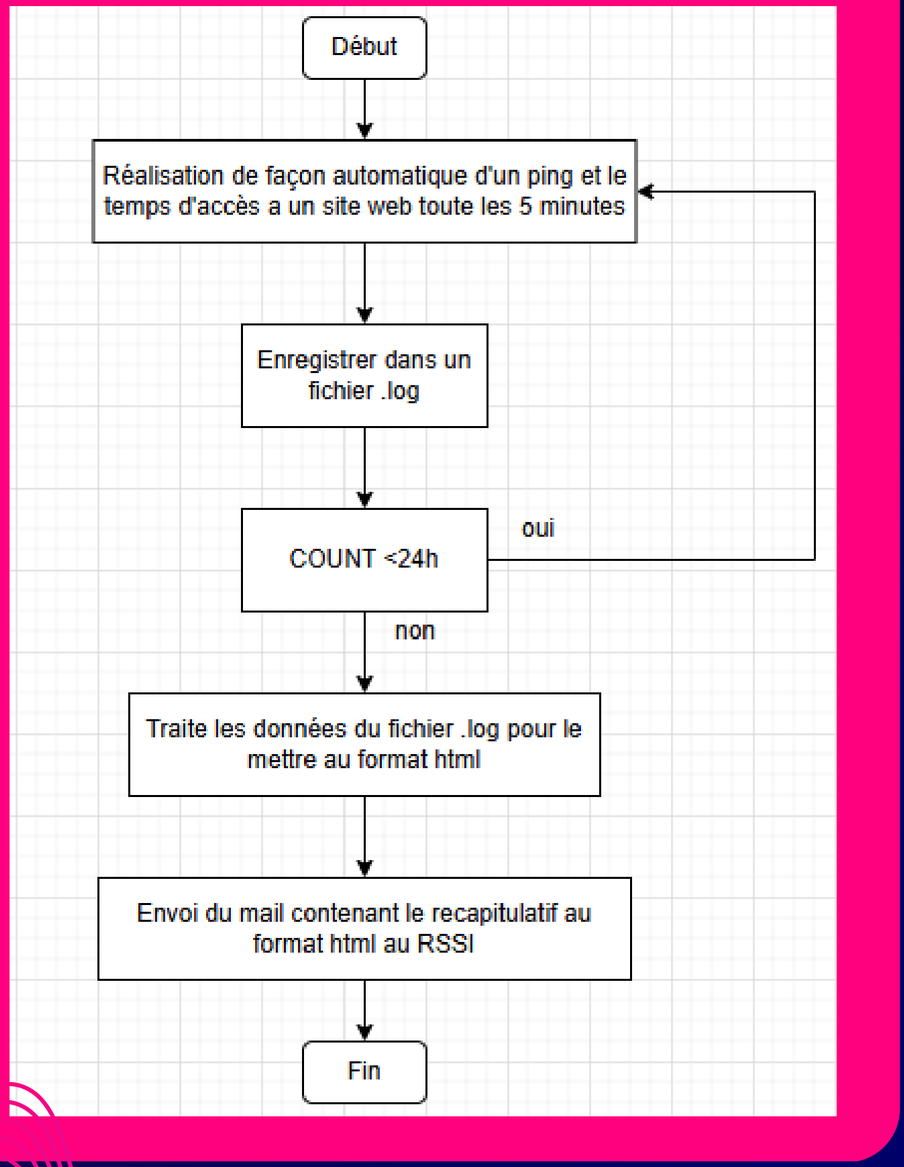
OBJECTIFS	ATTEINT		PARTIELLEMENT ATTEINT		NON ATTEINT
PERFORMANCE	✓	1 s	⚠	2 s	!
DISPONIBILITE	✓	99,8 %	⚠	98,5 %	!

Tâches élémentaires

Répartition du travail en 3 tâches élémentaires



Algorigramme



Le code: Script requête ping et web

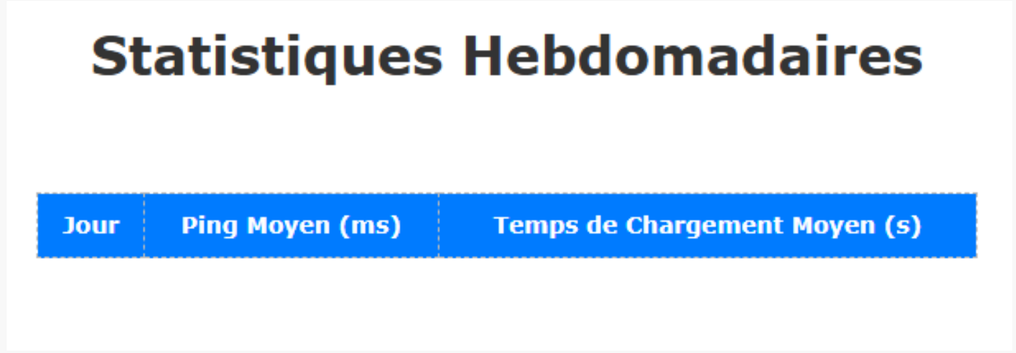
```
# Boucle principale pour collecter les données toutes les 5 minutes
while true; do
    # Récupérer la date et l'heure actuelle
    TIMESTAMP=$(date "+%Y-%m-%d %H:%M:%S")

    # Mesurer le temps total avec curl (en secondes)
    CURL_TIME=$(curl -s -w '%{time_total}' -o /dev/null "$URL")

    # Convertir le temps en millisecondes
    CURL_TIME_MS=$(echo "$CURL_TIME * 1000" | bc)

    # Formater le temps de chargement pour avoir deux décimales
    CURL_TIME_FORMATTED=$(printf "%.2f" $CURL_TIME_MS)

    # Mesurer le ping moyen
    PING_RESULT=$(ping -c 1 "$HOST" | grep 'time=' | awk -F 'time=' '{print $2}' | cut -d ' ' -f1)
    if [ -z "$PING_RESULT" ]; then
        PING_RESULT=0
    fi

    # Ajouter les résultats au total
    PING_TOTAL=$(echo "$PING_TOTAL + $PING_RESULT" | awk '{print $1 + $2}')
    TIME_TOTAL=$(echo "$TIME_TOTAL + $CURL_TIME_MS" | awk '{print $1 + $2}')
    COUNT=$((COUNT + 1))

    # Enregistrer les résultats dans le fichier log
    echo "[$TIMESTAMP] Ping: ${PING_RESULT} ms | Temps Chargement: ${CURL_TIME_FORMATTED} ms" >> "$FICHIER_LOG"
```

Le code: Script génération de tableau HTML

```
# Fichier HTML
HTML_FILE="tableau.html"
URL="https://www.ac-rennes.fr"
HOST="ac-rennes.fr"

# Initialisation des variables pour le calcul des moyennes
PING_TOTAL=0
TIME_TOTAL=0
COUNT=0

# Jours de la semaine
JOURS=("Lundi" "Mardi" "Mercredi" "Jeudi" "Vendredi" "Samedi" "Dimanche")

# Tableau pour stocker les moyennes
declare -A PING_MOYEN
declare -A TIME_MOYEN

# Calculer l'indice du jour actuel
get_jour_actuel() {
    DAY_OF_WEEK=$(date +%u)  # 1 = Lundi, 2 = Mardi, ..., 7 = Dimanche
    echo $((DAY_OF_WEEK - 1))  # Ajuster l'index pour correspondre à l'array JOURS
}

# Fonction pour générer ou mettre à jour le tableau HTML
generer_html() {
    local jour=$1
    local ping_moyen=$2
    local time_moyen=$3

    # Créer l'en-tête du fichier HTML si il n'existe pas
    if [ ! -f "$HTML_FILE" ]; then
        cat <<EOF > "$HTML_FILE"
```

Le code: Script génération de tableau HTML

```
<!DOCTYPE html>
<html lang="fr">
<head>
  <meta charset="UTF-8">
  <title>Stats Hebdomadaires</title>
  <style>
    table { width: 80%; margin: 20px auto; border-collapse: collapse; font-family: Arial, sans-s
    th, td { border: 1px solid #ccc; padding: 10px; text-align: center; }
    th { background: #007bff; color: white; }
    tr:nth-child(even) { background: #f2f2f2; }
    h1 { text-align: center; color: #333; }
  </style>
</head>
<body>
  <h1>Statistiques Hebdomadaires</h1>
  <table>
    <thead>
      <tr>
        <th>Jour</th>
        <th>Ping Moyen (ms)</th>
        <th>Temps de Chargement Moyen (s)</th>
      </tr>
    </thead>
    <tbody>
EOF
    fi

    # Vérifier si la ligne pour le jour existe déjà, sinon l'ajouter
    if ! grep -q "<tr><td>${JOURS[$jour]}</td>" "$HTML_FILE"; then
        echo "<tr><td>${JOURS[$jour]}</td><td>${ping_moyen}</td><td>${time_moyen}</td></tr>" >>
```



```bash
#!/bin/bash

# Adresse email de l'expéditeur (configuré dans ssmtp.conf)
#FROM_EMAIL="votre_email@gmail.com"

# Adresse email du destinataire
TO_EMAIL="miniprojetscriptdatametrie@gmail.com"

# Sujet de l'email
SUBJECT="Email envoyé automatiquement"

# Corps de l'email
MESSAGE="Ci-joint un exemple de fichier"

FILE="doc_mail.txt"

# Nombre maximum d'e-mails à envoyer (ou utilisez `while true` pour une boucle infinie)
MAX_EMAILS=10

# Compteur
COUNT=0

# Boucle pour envoyer des e-mails toutes les n secondes
while [ $COUNT -lt $MAX_EMAILS ]; do
  echo "$MESSAGE" | mutt -s "$SUBJECT" -a "$FILE" -- "$TO_EMAIL"

  # Afficher un message de confirmation dans le terminal
  echo "E-mail $((COUNT + 1)) envoyé à $TO_EMAIL"

  # Augmenter le compteur
  COUNT=$((COUNT + 1))

  # Pause de n secondes
  sleep 30
done

echo "Tous les e-mails ont été envoyés."
```

Le code: Envoi d’un rapport par mail

FIN

Merci pour votre écoute !

Présenté et codé par
GILLET Raphaël
&
JEHANNIN Pier-André