**Page de démarrage :**

**Une image contenant texte, capture d’écran, Police

Le contenu généré par l’IA peut être incorrect.**

**Connexion SSH en Git Bash :**

**cd : ssh -i "/c/Users/test/Documents/Tercium\_Stage/ssh\_keys/tercium-instance\_key"** [**ubuntu@37.156.46.238**](mailto:ubuntu@37.156.46.238)

**Commandes à automatiser via un Script :**

**Relancer les services système via le terminal :**

**cd : sudo systemctl restart prometheus**

**sudo systemctl restart node\_exporter**

**sudo systemctl restart blackbox\_exporter**

**sudo systemctl restart telegraf**

**Pour vérifier qu’ils tournent :**

**cd : sudo systemctl status prometheus**

**sudo systemctl status node\_exporter**

**sudo systemctl status blackbox\_exporter**

**sudo systemctl status telegraf**

**Accès aux interfaces Web :**

**Si tous les services sont actifs, l’on peut accéder aux interfaces suivantes :**

| **Composant** | **URL locale** | **URL distante (publique)** |  |
| --- | --- | --- | --- |
| **Prometheus** | **http://localhost:9090** | **http://37.156.46.238:9091 *(si exposé)*** |  |
| **Node Exporter** | **http://localhost:9100** | **http://37.156.46.238:9100 *(si exposé)*** |  |
| **Blackbox Exporter** | **http://localhost:9115** | **http://37.156.46.238:9115 *(si exposé)*** |  |
| **Grafana** | **http://localhost:3000** | [**http://37.156.46.238:3000**](http://37.156.46.238:3000) **✅** |  |

**Objectif du script principal**

* **Assurer la relance, la vérification des statuts, l’analyse des services, et la validation des accès Web.**
* **Centraliser les actions en une seule commande exécutée après chaque redémarrage ou incident.**

**A/ Méthodologie à connaitre et comprendre**

**Modules du script principal check\_and\_restart\_services.sh**

1. **Connexion SSH (facultatif pour intégration avec Git Bash Windows)**

**Une image contenant texte, capture d’écran, noir, Police

Le contenu généré par l’IA peut être incorrect.**

1. **Redémarrage automatique des services**

**Une image contenant texte, capture d’écran, Police, logiciel

Le contenu généré par l’IA peut être incorrect.**

1. **Vérification des statuts de services**

**Une image contenant texte, capture d’écran, Police

Le contenu généré par l’IA peut être incorrect.**

1. **Test des interfaces Web en local et distant**

**Une image contenant texte, capture d’écran, Logiciel multimédia, logiciel

Le contenu généré par l’IA peut être incorrect.**

1. **Export d’un tableau d’état au format Markdown/CSV/Log**

**Une image contenant texte, Appareils électroniques, capture d’écran, Police

Le contenu généré par l’IA peut être incorrect.**

**Reproduire le processus d’automatisation** sur n’importe quelle machine compatible (Ubuntu Server 20.04+) :

**1/ Nom du fichier :** **check\_and\_restart\_services.sh**

**2 / Emplacement recommandé dans l’arborescence :**

| **Emplacement absolu** | **Usage** |
| --- | --- |
| **/home/ubuntu/scripts/monitoring/** | **Répertoire dédié à la supervision et au redémarrage contrôlé** |
| **/opt/monitoring-scripts/** | **Optionnel pour standardisation dans un répertoire système** |

**3/ Création du répertoire :**

**Une image contenant texte, Police, capture d’écran, ligne

Le contenu généré par l’IA peut être incorrect.**

**4/ Création du fichier script (méthode reproductible)**

**Étape 1 – Accès et édition du fichier :**

**cd : nano check\_and\_restart\_services.sh**

**Étape 2 – Copier-coller le contenu suivant dans l’éditeur :**

**Une image contenant texte, capture d’écran, conception

Le contenu généré par l’IA peut être incorrect.**

**Étape 3 – Rendre le script exécutable : cd : chmod +x check\_and\_restart\_services.sh**

**\*Exécution du script : cd : ./check\_and\_restart\_services.sh**

**\*Export automatique du journal d’état**

**Le fichier généré (rapport\_etat\_services\_YYYYMMDD\_HHMM.log) est enregistré dans le même dossier que le script, pour archivage ou partage :**

**cd : cat rapport\_etat\_services\_20250722\_1425.log**

**Pour un autre opérateur : résumé des étapes :**

| **Étape** | **Commande à lancer** | **Description** |
| --- | --- | --- |
| **1** | mkdir -p /home/ubuntu/scripts/monitoring/ | **Crée le dossier** |
| **2** | nano check\_and\_restart\_services.sh | **Ouvre l’éditeur** |
| **3** | chmod +x check\_and\_restart\_services.sh | **Rends le script exécutable** |
| **4** | ./check\_and\_restart\_services.sh | **Lance le script** |

**B/ Récapitulatif de la démarche et Réalisation :**

**Étape 1 :** Démarrage automatique via **.service**

**But :** Lancer tous les composants nécessaires à la supervision dès le boot du système **(Ubuntu/Debian).**

**Fichiers concernés :**

**/etc/systemd/system/prometheus.service**

**/etc/systemd/system/node\_exporter.service**

**/etc/systemd/system/blackbox\_exporter.service**

**/etc/systemd/system/telegraf.service**

**Commandes générales :**

**sudo systemctl daemon-reexec**

**sudo systemctl enable prometheus.service**

**sudo systemctl enable node\_exporter.service**

**sudo systemctl enable blackbox\_exporter.service**

**sudo systemctl enable telegraf.service**

**sudo systemctl start prometheus.service**

**Étape 2 :** Makefile – Structuration des tâches de supervision

**But :** Automatiser et documenter proprement toutes les opérations de supervision et d'analyse en modules :

**Exemple de structure Makefile :**

**Une image contenant texte, capture d’écran, Police

Le contenu généré par l’IA peut être incorrect.**

**Avantages : Réutilisable / Lisible / Modulaire / Compatible CI/CD ou déclenchement externe**

**Étape 3 : cron** – Analyses récurrentes ou planifiées

**But :** Programmer des tâches périodiques ou récursives (vérifications, export, alerte, rotation de logs, simulation de charge...).

**Exemple :**

**0 \* \* \* \* /home/ubuntu/scripts/check\_endpoints.sh >> /var/log/prometheus/endpoint\_check.log**

**Remarques :**

**cat :** affiche le contenu d’un fichier ou d’un flux.

**echo :** imprime une ligne, utile pour écrire dans un fichier (avec >> ou >).

**.service =** lancement unitaire (au boot), pour des programmes indépendants.

**Makefile =** logique séquentielle et modulaire, par tâche définie.

**cron =** répétition dans le temps, automatisation à fréquence déterminé

**Explication des usages et sens de : cat vs echo | makefile vs .service vs cron**

**1. Différence entre cat et echo**

| **Commande** | **Usage principal** | **Exemple** | **Particularités** |
| --- | --- | --- | --- |
| **cat** | **Afficher le contenu complet d’un fichier** | **cat fichier.txt** | **Lecture ligne à ligne d’un fichier. Très utile pour les logs, scripts, configurations.** |
| **echo** | **Afficher une chaîne de caractères (texte brut ou variable)** | **echo "Hello" ou echo $HOSTNAME** | **Sert à insérer une ligne simple. Peu adapté aux fichiers longs.** |

**Donc :**

**cat = affiche un fichier déjà existant.**

**echo = affiche du texte en interne ou à la volée.**

**2. Usage et différence entre : Makefile vs systemd .service vs cron**

| **Mécanisme** | **Usage principal** | **Lancement** | **Exemple d’usage** |
| --- | --- | --- | --- |
| **Makefile** | **Chaînage logique de cibles/actions** | **Manuel : make** | **Lancer check, build, deploy conditionnellement** |
| **.service** | **Intégration au boot système via systemd** | **Auto au boot ou via systemctl** | **Démarrer prometheus, grafana, ou ton script de vérif à chaque redémarrage** |
| **cron** | **Tâche périodique (horaire, fréquence)** | **Automatique (horaire)** | **Vérif toutes les heures ou tous les jours** |

**Comparaison rapide :**

| **Critère** | **Makefile** | **.service (systemd)** | **cron** |
| --- | --- | --- | --- |
| **Lancement manuel** | **✅make target** | **❌** | **❌** |
| **Automatique au boot** | **❌** | **✅** | **❌** |
| **Répétition planifiée** | **❌** | **Possible avec timer** | **✅ crontab** |
| **Gestion de dépendances** | **✅** | **✅ via After= / Requires=** | **❌** |
| **Simplicité d’usage** | **Moyen** | **Moyen** | **Facile** |
| **Bon pour scripts Bash** | **✅** | **✅** | **✅** |

**Recommandations selon ton contexte :**

**✅ Tu veux lancer une séquence conditionnelle de scripts ou routines → Makefile**

**✅ Tu veux que cela démarre avec la machine et survive aux redémarrages → .service**

**✅ Tu veux exécuter toutes les heures / jours / semaines automatiquement → cron**

**Synthèse de la démarche : déplacement et préparation du script check\_and\_restart\_services.sh**

**1. Objectif**

Organiser proprement les scripts de monitoring en plaçant check\_and\_restart\_services.sh dans le bon répertoire (~/monitoring/scripts/), avec des droits corrects, pour une future automatisation (via .service, Makefile ou cron).

**2. Étapes exécutées**

| **Étape** | **Commande** | **Finalité** |
| --- | --- | --- |
| **Vérifier la localisation actuelle** | **ls -l /home/ubuntu/check\_and\_restart\_services.sh** | **Confirmer l’existence du script** |
| **Déplacer dans le répertoire prévu** | **sudo mv /home/ubuntu/check\_and\_restart\_services.sh /home/ubuntu/monitoring/scripts/** | **Structurer les scripts** |
| **Changer propriétaire** | **sudo chown ubuntu:ubuntu /home/ubuntu/monitoring/scripts/check\_and\_restart\_services.sh** | **Rendre le script modifiable/exécutable par l’utilisateur** |
| **Rendre exécutable** | **sudo chmod +x /home/ubuntu/monitoring/scripts/check\_and\_restart\_services.sh** | **Permettre son exécution** |
| **Vérification finale** | **ls -l /home/ubuntu/monitoring/scripts/** | **Contrôle de la présence et des droits** |

**3. Étape suivante**

* Ajouter au **démarrage automatique** via un fichier .service
* Inclure dans le **Makefile** avec logique conditionnelle
* Planifier avec **cron** (check toutes les X heures ou avant test de charge

**Deux Méthodes :**

**Deux méthodes possibles** pour déclencher automatiquement le **Makefile** situé dans **/opt/monitoring**/ au démarrage de la machine :

**Option A — Créer un service systemd dédié**

**Méthode propre, standard, compatible avec toutes les distributions modernes.**

**Fichier : /etc/systemd/system/monitoring-start.service**

Une image contenant texte, capture d’écran, Police

Le contenu généré par l’IA peut être incorrect.

**Commandes d’activation : cd : sudo systemctl daemon-reexec # ou daemon-reload cd : sudo systemctl enable monitoring-start.service cd : sudo systemctl start monitoring-start.service**

**Vérification : cd : systemctl status monitoring-start.service Option B — Ajout dans /etc/rc.local (moins propre)**

Une image contenant texte, capture d’écran, Police

Le contenu généré par l’IA peut être incorrect.

Il faut que le fichier **/etc/rc.local** existe, soit exécutable, et que le service **rc-local** soit activé :

**cd : sudo chmod +x /etc/rc.local**

**sudo systemctl enable rc-local**

**sudo systemctl start rc-local**

**Recommandation finale : Méthode préférée** : **systemd** → plus propre, gérable, observable avec **journalctl.**

**Script Bash :** crée automatiquement le **service systemd monitoring-start.service**, **Puis** l’active au démarrage, et l’exécute immédiatement.

**Script Bash : create\_monitoring\_service.sh**

**Une image contenant texte, capture d’écran, logiciel

Le contenu généré par l’IA peut être incorrect.**

**Utilisation :**

1. **Sauvegarde le fichier :**

**cd : nano create\_monitoring\_service.sh**

1. **Donne les droits d’exécution : cd : chmod +x create\_monitoring\_service.sh**
2. **Exécute le script : cd : ./create\_monitoring\_service.sh**

**Autre version avec deux objectifs :**

* Vérifier et créer un répertoire de travail /opt/monitoring s’il est vide ou absent.
* Générer un **Makefile** standardisé avec les cibles **start, reload, restart** pour **prometheus, node\_exporter** et **blackbox\_exporter.**

**Script : setup\_prometheus\_makefile\_and\_service.sh**

**Une image contenant texte, capture d’écran

Le contenu généré par l’IA peut être incorrect.**

**Une image contenant texte, capture d’écran, Police, logiciel

Le contenu généré par l’IA peut être incorrect.**