

21/02/2025

Projet LPIC-101

Rapport de projet



PERREIN Alcide - NANDILLON Maxence

3ESGI, NANTES

TABLE DES MATIERES

1. Introduction.....	2
1.1 Contexte du projet.....	2
1.2 Objectifs du projet.....	2
1.3 Environnement de travail.....	2
2. Gestion de projet	2
2.2 Répartition des tâches.....	2
2.3 Outils de gestion de projet.....	3
3. Développement technique	4
3.1 Architecture du système	4
3.2 Fonctionnalités implémentées	4
3.3 Outils et technologies utilisés	4
4. Difficultés rencontrées.....	4
4.1 Problèmes techniques	4
4.2 Solutions apportées.....	5
5. Conclusion	5
5.1 Bilan du projet	5
5.2 Perspectives d'amélioration	5
6. Annexes	5
6.1 Guide d'utilisation	5
6.2 Liste des outils et services utilisés	5

1. INTRODUCTION

1.1 CONTEXTE DU PROJET

Le projet LPIC-101 a été initié dans le cadre de notre formation à l'ESGI en troisième année, option Systèmes, Réseaux et Cloud (SRC). Ce projet a été conçu par notre professeur de Linux afin de nous préparer à la certification LPIC-101, une certification reconnue dans le domaine de l'administration système Linux. L'objectif principal était de créer une application en ligne de commande sous Debian pour gérer les tâches d'administration système dans une petite entreprise.

1.2 OBJECTIFS DU PROJET

L'objectif du projet était de développer plusieurs scripts Bash pour offrir un menu interactif permettant de :

- Gérer les utilisateurs et les groupes.
- Automatiser des tâches via Cron.
- Surveiller l'espace disque et les processus.
- Sauvegarder des dossiers spécifiques.
- Configurer la journalisation système.

Ces scripts devaient être conçus pour être utilisés par les agents RH, les employés et l'équipe IT de l'entreprise.

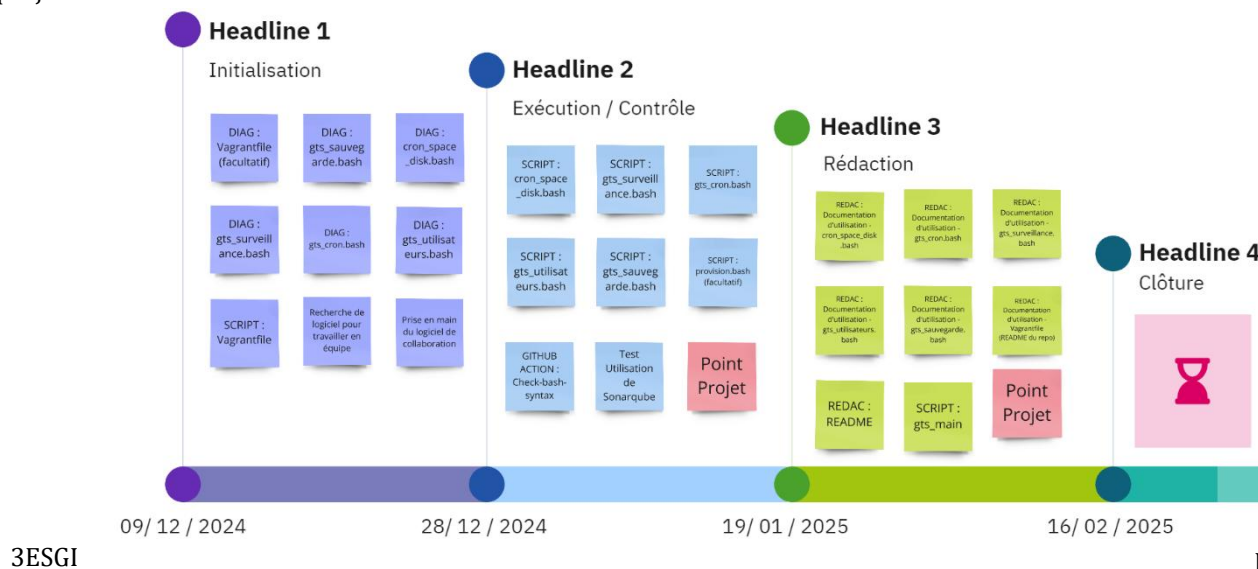
1.3 ENVIRONNEMENT DE TRAVAIL

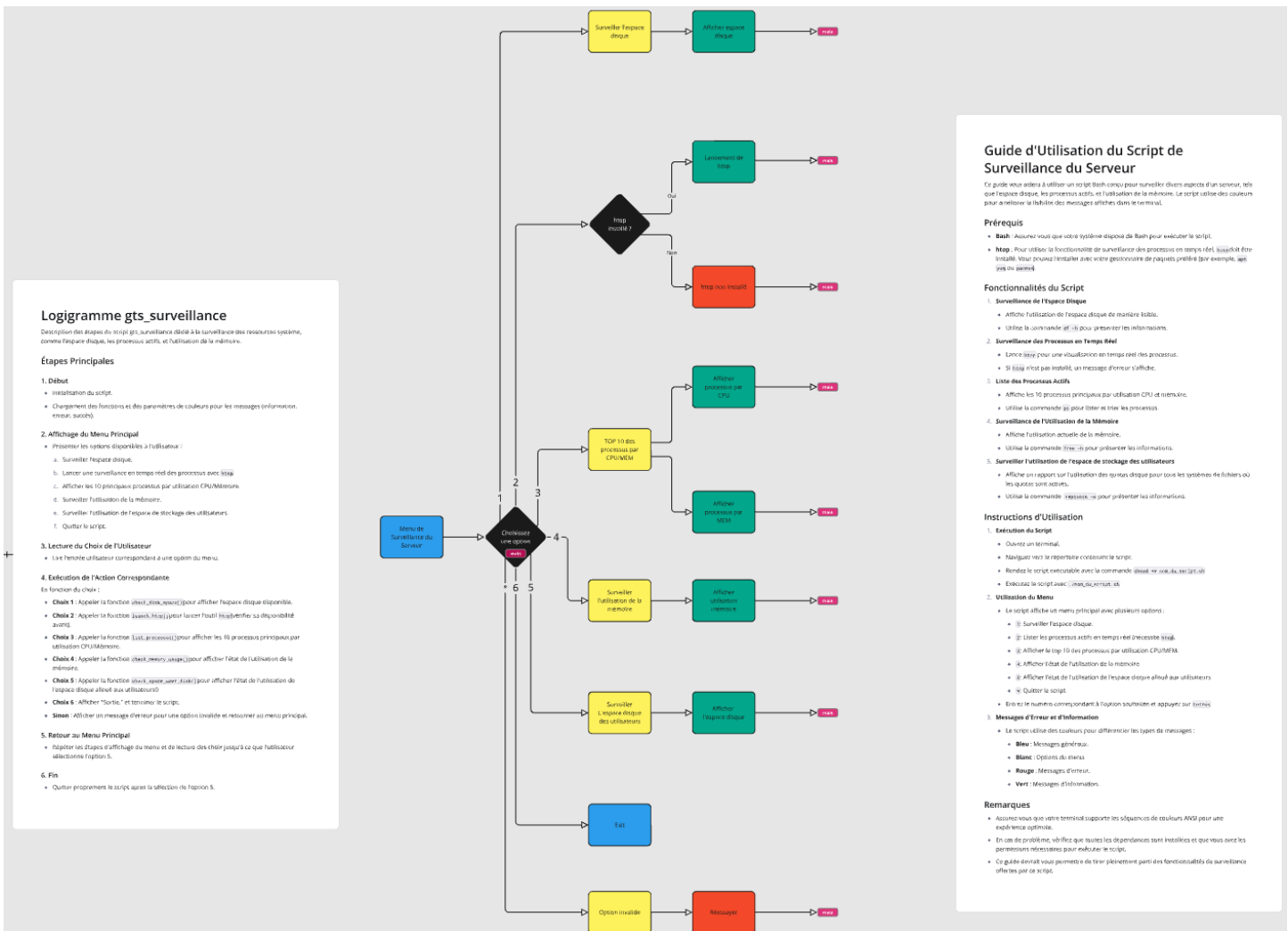
Pour réaliser ce projet, nous étions deux étudiants en troisième année à l'ESGI, en spécialité SRC, préparant le diplôme de chef de projet logiciel et réseaux. Nous avons travaillé en collaboration, en utilisant des outils de gestion de projet et des technologies modernes pour garantir une réalisation efficace et professionnelle.

2. GESTION DE PROJET

2.2 REPARTITION DES TACHES

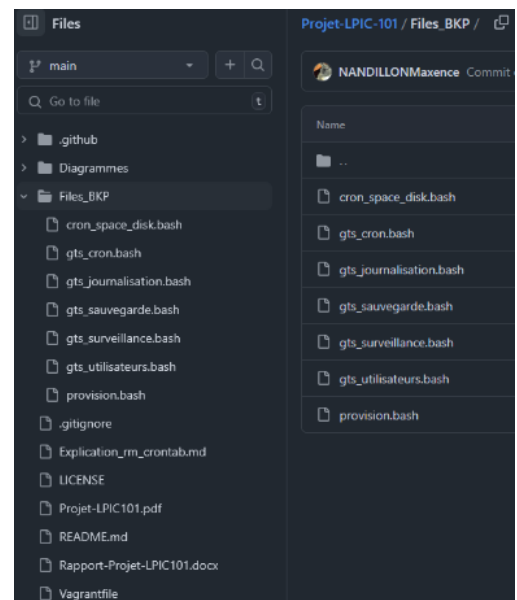
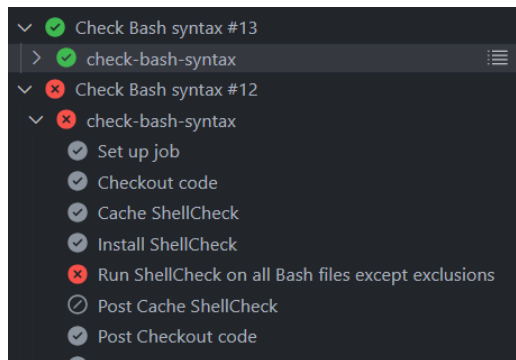
La répartition des tâches a été effectuée en fonction des compétences de chacun. Nous avons utilisé Miro pour concevoir une timeline interactive, facilitant le suivi des tâches et du projet. En complément, nous avons créé des logigrammes, des diagrammes UML et des documents d'utilisation, ce qui nous a permis de clarifier les responsabilités de chacun et de suivre l'avancement du projet de manière structurée.





2.3 OUTILS DE GESTION DE PROJET

- **Git et GitHub** : Nous avons utilisé Git pour le versioning du code et GitHub pour le stockage et la collaboration. Nous avons également mis en place des **Hooks**, des **GitHub Actions** et des pipelines CI/CD pour automatiser les tests et les déploiements.
- **SonarQube** : Pour la détection de vulnérabilités et la qualité du code.
- **Miro** : Pour la création de diagrammes et la planification des tâches.
- **Vagrant, VirtualBox et VMware**: Pour la création d'environnements virtuels reproductibles.



3. DEVELOPPEMENT TECHNIQUE

3.1 ARCHITECTURE DU SYSTEME

Le système est basé sur une architecture modulaire, où chaque fonctionnalité est gérée par un script Bash distinct. Les scripts sont conçus pour être exécutés par des groupes d'utilisateurs spécifiques, garantissant ainsi une sécurité et une gestion des droits adaptées.

Pour plus de détails, veuillez consulter le dépôt du projet, où vous trouverez une documentation complète, de l'installation à l'utilisation des scripts. <https://github.com/NANDILLONMaxence/Projet-LPIC-101/tree/main>

3.2 FONCTIONNALITES IMPLEMENTEES

- **Gestion des utilisateurs et des groupes** : Création, suppression et affectation des utilisateurs aux groupes.
- **Automatisation avec Cron** : Planification et gestion des tâches cron.
- **Surveillance système** : Surveillance de l'espace disque, des processus et de la mémoire.
- **Sauvegarde de dossiers** : Sauvegarde manuelle et automatique des dossiers.
- **Configuration de la journalisation** : Configuration de rsyslog pour la centralisation des logs.

```
agent_RH-1@Debian12:~$ gts_main|
```

```
=== Menu Principal GTS ===  
1. Gestion des utilisateurs  
2. Surveillance du système  
3. Gestion des taches avec cron  
4. Gestion des sauvegardes  
5. Gestion de la journalisation  
5. Quitter  
  
Sélectionnez une option : |
```

3.3 OUTILS ET TECHNOLOGIES UTILISES

- **Bash** : Pour la création des scripts.
- **Vagrant** : Pour la création et la gestion des machines virtuelles.
- **doas** : Alternative à sudo pour la gestion des permissions.
- **chmod 2750** : Pour la gestion des permissions spécifiques.
- **Codviz** : Pour la création d'organigrammes des scripts.
- **Quota** : Pour la gestion et le suivi des ressources système.

4. DIFFICULTES RENCONTREES

4.1 PROBLEMES TECHNIQUES

- **Configuration de GRUB** : La personnalisation de GRUB pour un démarrage sécurisé a été complexe, notamment en raison des spécificités du système Debian.
- **Intégration de Cron** : L'automatisation des tâches via Cron a posé des problèmes de synchronisation et de gestion des erreurs.

4.2 SOLUTIONS APPORTEES

- Pour GRUB, nous avons consulté la documentation officielle et testé plusieurs configurations avant de trouver la solution optimale.
- Pour Cron, nous avons implémenté des scripts de vérification pour garantir que les tâches s'exécutent correctement.

5. CONCLUSION

5.1 BILAN DU PROJET

Le projet LPIC-101 a été une expérience enrichissante qui nous a permis de mettre en pratique les connaissances acquises en cours. Nous avons réussi à développer un environnement fonctionnelle répondant aux besoins d'une petite entreprise en matière d'administration système.

5.2 PERSPECTIVES D'AMELIORATION

- **Gestion des erreurs** : Intégrer des contrôles de validation additionnels afin d'améliorer la précision du dépannage.
- **Automatisation avancée** : Intégrer des outils d'automatisation plus poussés comme Ansible.

6. ANNEXES

6.1 GUIDE D'UTILISATION

Un guide d'utilisation détaillé est disponible sur le dépôt GitHub du projet : <https://github.com/NANDILLONMaxence/Projet-LPIC-101>

6.2 LISTE DES OUTILS ET SERVICES UTILISES

- **Git et GitHub** : Versioning et collaboration.
- **Vagrant et VirtualBox** : Environnements virtuels.
- **Miro** : Diagrammes et planification.
- **SonarQube et Github Action** : Qualité du code.
- **Codviz** : Organigrammes des scripts.

Vagrantfile :

```
Vagrant.configure("2") do |config|
  config.vm.box = "bento/debian-12"

  config.vm.provider :virtualbox do |vb|
    vb.gui = false
    vb.customize ["modifyvm", :id, "--natdnshostresolver1", "on"]
    vb.customize ["modifyvm", :id, "--vram", "128"]
    vb.memory = "2048"
    vb.cpus = 2

    # === Add data disk (50GB) ===
    unless File.exist?("./data_disk.vdi")
```

Codviz :

