



Mon rapport de stage

Projet Naruto RP (infrastructure serveur & collaboration à distance)

Ayman boudjemaa

mois de juin

1. Présentation de l'entreprise

Naruto RP est un jeu en ligne amateur développé par une équipe de passionnés, basé sur l'univers du manga Naruto. Ce projet communautaire propose une expérience multijoueur en monde ouvert, avec des mécaniques de combat, de progression, et des interactions entre joueurs.

Le développement du jeu repose sur une organisation en travail collaboratif à distance, regroupant plusieurs rôles : développeurs, graphistes, modérateurs, animateurs, et techniciens serveurs.

2. Objectifs du stage

Ce stage avait pour but de :

Découvrir les rouages techniques derrière l'hébergement et la stabilité d'un serveur de jeu.

Comprendre les enjeux de la gestion réseau, de la sécurité, et des performances serveur .

Participer à la maintenance, la surveillance et l'optimisation des services liés au jeu.

intégrer une équipe projet 100 % distante, avec outils collaboratifs.

Missions principales

Supervision et stabilité des serveurs

Durant le stage, une de mes tâches principales était d'assister les administrateurs dans la gestion des serveurs de jeu :

Surveillance en temps réel des ressources : charge CPU, mémoire vive, connexions simultanées

Analyse des logs système pour détecter les erreurs fréquentes ou les redémarrages anormaux

Mise en place de redémarrages automatiques planifiés, pour éviter les crashes liés à la surcharge mémoire

Participation à la mise à jour des services de sécurité (firewall) pour bloquer les connexions suspectes

Tests de limitation de bande passante et suivi du ping pour éviter la latence excessive

Gestion des incidents serveur

L'équipe rencontrait régulièrement des incidents liés à l'instabilité du serveur, ce qui m'a permis de participer à :

L'analyse des causes de crash serveurs : erreurs mémoire, scripts mal exécutés, saturation RAM

- La mise en place d'un système de sauvegarde automatique toutes les 30 minutes
- ⑩ La gestion des attaques extérieures :
 - Tentatives de DDoS
 - ⑩ Tentatives de scan de ports
 - ⑩ Surconsommation par bots

Ces incidents m'ont sensibilisé à la protection d'un serveur public, aux bonnes pratiques en termes de résilience, et à la nécessité d'un plan de secours (reboot rapide, monitoring externe).

Création de contenus vidéo et communication

En complément du travail technique, j'ai participé à la réalisation de vidéos promotionnelles pour le jeu :

- **Capture de gameplay** : enregistrement d'extraits illustrant les fonctionnalités, les combats et les mises à jour
- **Montage vidéo** : utilisation de logiciels comme CapCut et DaVinci Resolve pour assembler les séquences, ajouter du texte et intégrer une bande-son
- **Formats courts** : création de vidéos verticales adaptées à TikTok et Instagram Reels
- **Graphisme simple** : ajout de titres dynamiques, effets d'apparition, transitions douces
- **Publication** : export en bonne qualité et transmission aux community managers pour diffusion

Ces vidéos ont servi à **promouvoir les mises à jour** du jeu et à **attirer de nouveaux joueurs** en mettant en avant l'univers visuel du projet.

Outils et logiciels utilisés pour la gestion serveur et réseau

Surveillance & monitoring :

- **htop** – Pour surveiller en temps réel l'usage du CPU, de la mémoire et des processus
- **screen** – Pour garder des processus actifs sur un serveur distant même après déconnexion
- **vnStat** – Pour surveiller la bande passante (upload / download)
- **netstat** – Pour voir les connexions réseau actives et ports utilisés
- **iftop** – Pour afficher l'utilisation réseau en direct par IP

Sécurité et protection du serveur :

- **fail2ban** – Pour bloquer automatiquement les adresses IP suspectes ou malveillantes (tentatives de connexion SSH, scans)
- **UFW** (Uncomplicated Firewall) – Pour gérer les règles de pare-feu simplement sur Ubuntu/Debian
- **iptables** – Pour les règles avancées de filtrage réseau
- **rkhunter** – Pour vérifier la présence de rootkits ou malwares sur le serveur

Sauvegarde et automatisation :

- **cron** – Pour exécuter des tâches automatiques planifiées (comme redémarrer le serveur, sauvegarder des fichiers, etc.)
- **rsync** – Pour copier ou sauvegarder les fichiers du serveur vers un autre emplacement
- **tar** – Pour compresser les sauvegardes en .tar.gz

Accès à distance & gestion :

- **PuTTY** – Logiciel client SSH sur Windows pour se connecter au serveur
- **FileZilla** – Pour le transfert de fichiers via SFTP (par ex. logs, sauvegardes)
- **WinSCP** – Alternative graphique pour accéder au serveur via SSH et transférer des fichiers

Utilitaires serveur courants :

- **systemctl** – Pour démarrer, redémarrer ou surveiller les services système (ex. : redémarrer le service du jeu)
- **journalctl** – Pour consulter les logs système et services en profondeur

Conclusion

Ce stage m'a offert une immersion dans l'envers du décor d'un jeu en ligne.

Le fonctionnement technique d'un serveur de jeu en ligne (RAM, latence, sockets, connexions)

La complexité de maintenir une plateforme stable avec des joueurs actifs en permanence

L'importance de documenter les incidents et les correctifs apportés

Le besoin d'une bonne organisation et de réactivité dans les interventions techniques à distance

La complémentarité entre administration système et collaboration projet

Cette expérience m'a donné envie de creuser davantage la gestion des infrastructures réseaux et serveurs, et m'a apporté des bases solides que je compte approfondir dans la suite de ma formation.