

Installer, configurer Moodle et ses plugins en 10 minutes

Victor Da Silva Caseiro

Sorbonne Université – Faculté des Sciences et Ingénierie - CAPSULE
4 place Jussieu
75005 Paris

Thomas Naudin

Sorbonne Université – Faculté des Sciences et Ingénierie – CAPSULE
4 place Jussieu
75005 Paris

Cyrille Selami

Sorbonne Université – Faculté des Sciences et Ingénierie – CAPSULE
4 place Jussieu
75005 Paris

Résumé

Depuis son lancement en 2017, l'architecture serveur du projet "Moodle Sciences", destiné à la faculté des Sciences et de l'Ingénierie de Sorbonne Université, a connu une augmentation significative de sa complexité. Ce projet, visant à fournir une plateforme pédagogique robuste et évolutive, a vu son infrastructure se diversifier et se sophistiquer au fil des années. En conséquence, le temps requis pour déployer de nouvelles plateformes s'est allongé, et le risque d'erreurs lors des déploiements a considérablement augmenté.

Pour faire face à cette complexité croissante et à la charge de travail accrue associée, l'équipe en charge de la gestion de la plateforme a décidé de se tourner vers une solution d'automatisation des tâches : Ansible. Cet outil permet d'optimiser les processus d'installation, de configuration et de maintenance des serveurs, réduisant ainsi les erreurs manuelles et accélérant le déploiement des nouvelles instances.

Dans cet article, nous allons examiner en détail la situation actuelle en trois parties distinctes. Tout d'abord, nous présenterons le contexte historique et l'évolution des infrastructures de la plateforme pédagogique, en retraçant son développement depuis ses débuts jusqu'à son architecture actuelle. Ensuite, nous nous pencherons sur le projet Ansible, qui a été mis en place pour répondre à la nécessité de gagner en efficacité et en productivité. Enfin, nous aborderons les perspectives d'avenir pour ce projet Ansible, en abordant les évolutions prévues et les prochaines étapes pour continuer à améliorer la gestion de la plateforme.

Mots-clefs

Ansible, Moodle, Automatisation, Mise en conformité

1 Historique de l'infrastructure Moodle

Le projet Moodle Sciences a débuté à la fin de l'année 2016 avec pour objectif de fournir une plateforme Moodle commune à toute la faculté des Sciences et de l'Ingénierie de l' Université Pierre et Marie Curie (UPMC), pour une mise en production au début de l'année universitaire 2017.

À la rentrée 2017, l'infrastructure mise en place comprenait un total de neuf serveurs, répartis sur deux environnements distincts : un environnement de développement et un environnement de production.

1.1 Environnement de développement (Moodle Dev)

L'environnement de développement était composé d'un serveur Web unique, où le répertoire « moodledata » était stocké localement, ainsi que de deux serveurs de bases de données en cluster Galera¹.

Il servait aux tests techniques de la plateforme et de ses plugins additionnels, ainsi qu'aux tests fonctionnels. Tout ceci était géré par un seul administrateur Moodle.

1.2 Environnement de production (Moodle Science)

L'environnement de production était composé de quatre serveurs Web, placés derrière un répartiteur de charge. Le répertoire « moodledata » était hébergé sur un partage NFS (*Network File System*) accessible par les quatre serveurs Web. Pour la base de données, deux serveurs fonctionnaient également en cluster Galera.

1.3 Annualisation des instances Moodle

Au cours de l'année universitaire 2017, et après avoir consulté plusieurs services TICE (Technologies de l'Information et de la Communication pour l'Enseignement) d'universités françaises ayant déployé Moodle, il a été décidé d'adopter une stratégie d'annualisation des instances de la plateforme.

Cette stratégie consiste à déployer chaque année une nouvelle instance de Moodle sur la même infrastructure, en parallèle de celle de l'année précédente. Ce modèle fonctionne sur un cycle de roulement de deux ans.

Cette approche présente plusieurs avantages significatifs. Le fait de déployer une nouvelle instance de Moodle chaque année permet de maintenir un répertoire « moodledata » et une base de données de tailles raisonnables, permettant ainsi de ne pas dégrader les performances de la plateforme au fil des années. De plus, un cycle de renouvellement annuel facilite l'adaptation des utilisateurs à ce mode de gestion, contrairement à des changements bisannuels ou trisannuels, où les utilisateurs pourraient se retrouver pris de cours par un changement inattendu.

Cependant, cette méthode d'annualisation comporte également des inconvénients notables. Le déploiement de chaque nouvelle instance de Moodle nécessite un travail important de la part de l'administrateur. Ce processus, réalisé à la main sur de nombreux serveurs, chronophage, augmente également les risques d'erreur de configuration.

2 Infrastructure Moodle actuelle

Lors de la création du Centre d'Accompagnement pour la Pédagogie et SUpport à L'Expérimentation (CAPSULE), et face à l'augmentation progressive de l'utilisation de Moodle Sciences au fil des ans, il a été décidé de repenser l'infrastructure. Cela a impliqué un redimensionnement des environnements existants (Dev et Sciences) et l'ajout d'un environnement intermédiaire dédié aux tests fonctionnels (Préprod).

¹Galera est un outil de réplication synchrone pour MySQL et MariaDB, assurant haute disponibilité et continuité de service en cas de panne.

De plus, il a été décidé d'annualiser non seulement les instances de Moodle, mais également les environnements d'hébergement, afin de ne plus être contraints par les limitations de compatibilité entre les versions de PHP et de MariaDB entre deux versions de Moodle.

Depuis 2022, l'architecture serveur de Moodle Sciences comprend :

- **Moodle Dev** : 2 serveurs web, 2 serveurs de bases de données, 1 serveur Redis² ;
- **Moodle Préprod** : 4 serveurs web, 2 serveurs de bases de données, 1 serveur Redis ;
- **Moodle Sciences** : 4 serveurs web, 2 serveurs de bases de données, 1 serveur Redis.

2.1 Moodle Dev (Tests techniques)

L'instance Moodle Dev est composée de deux serveurs web (Apache) et de deux serveurs de bases de données (MariaDB en mode réplique).

Cette plateforme est exclusivement utilisée par les développeurs de CAPSULE pour valider les mises à jour de Moodle avant leur déploiement en production. Cette instance permet notamment la sélection de la version de Moodle à déployer en production, qui est une étape clé du processus.

Elle s'accompagne de la validation, de l'intégration des nouveaux plugins, qui sont rigoureusement testés pour garantir leur bon fonctionnement. Parallèlement, le développement interne des plugins y est également réalisé.

2.2 Moodle Préprod (Tests fonctionnels)

Moodle Préprod, composé de quatre serveurs web (Apache) et de deux serveurs de bases de données (MariaDB en mode réplique), hérite de la version de Moodle validée par les tests techniques sur Moodle Dev. Tous les plugins validés y sont également déployés.

Son rôle est de permettre des tests fonctionnels réalisés par l'équipe pédagogique (7 personnes). Chaque fonctionnalité, chaque plugin, ainsi que toutes les configurations nécessaires, sont minutieusement validés. Cette instance se veut identique à l'environnement de production pour garantir une transition fluide.

2.3 Moodle Sciences (Production)

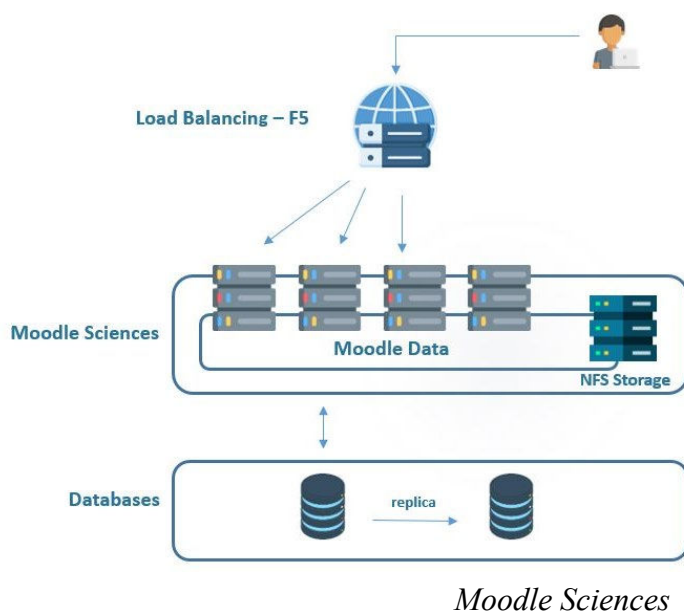


Figure 1: Architecture

²Redis est une base de données en mémoire rapide, utilisée pour la mise en cache, la gestion de sessions et le traitement de données en temps réel.

Moodle Sciences est la plateforme officielle de la Faculté des Sciences et de l'Ingénierie de Sorbonne Université (issue de la fusion entre l'UPMC et l'université Paris-Sorbonne). Composée également de quatre serveurs web (Apache) et de deux serveurs de bases de données (MariaDB en mode réplica) ainsi que son serveur Redis dédié. Elle est l'aboutissement et l'héritage du travail réalisé sur Moodle Dev et Moodle Préprod.

En cette rentrée de septembre 2024, Moodle Sciences compte :

- 13 599 comptes actifs ;
- 1 391 cours ;
- 51 plugins additionnels.

Ces trois instances (Moodle Dev, Moodle Préprod et Moodle Sciences) bénéficient toutes d'une authentification CAS, d'une répartition de charge (*load balancing*) et d'un point de montage commun (stockage NFS).

2.4 Avantages de l'infrastructure Moodle actuelle

L'attribution d'un rôle spécifique à chaque plateforme (test, préproduction et production) présente l'avantage de segmenter efficacement les besoins et de cloisonner les dysfonctionnements autant que possible. Cette approche permet, tout au long de l'année, de procéder à des tests ciblés sur l'environnement de test, puis de vérifier l'intégrité et la stabilité des fonctionnalités en environnement de préproduction, avant de déployer les mises à jour en toute sécurité dans l'environnement de production. Ce processus garantit une meilleure maîtrise des risques, tout en offrant une flexibilité pour ajuster et affiner les évolutions avant une mise en production définitive.

2.5 Inconvénients de l'infrastructure Moodle actuelle

En janvier 2022, nous avons été confrontés à l'installation et la configuration de tous ces environnements, ce qui nous a fait prendre conscience de l'ampleur du travail que cette infrastructure nécessitait.

Toute la configuration des environnements était effectuée manuellement, notamment l'installation et la configuration des plugins. Ce processus était redondant et une source d'erreurs fréquentes.

Par exemple, l'installation d'un plugin sur Moodle Dev nécessite une intervention sur chacun des serveurs. Étant donné qu'il y a deux serveurs web dans cet environnement, le plugin doit être installé deux fois, une fois sur chaque serveur, sans possibilité de passer par l'interface graphique de la plateforme. Ce même processus devait être répété quatre fois pour Moodle Préprod, puis encore quatre fois si le plugin était validé pour une mise en production.

3 Découverte d'Ansible

Lors des JRES 2019, nous avons découvert la technologie Ansible, largement adoptée au sein des universités. Un échange avec l'équipe informatique COMETE (Centre Optimisé de MEdiatisation et de Technologies Educatives) de l'Université de Nanterre a mis en lumière les nombreuses possibilités qu'offre Ansible pour le déploiement de Moodle. C'est ainsi que nous nous sommes orientés vers l'utilisation de cet outil pour la gestion de nos propres plateformes.

3.1 Présentation de la technologie

Ansible permet l'automatisation de tâches informatiques complexes, telles que la gestion des configurations, le déploiement d'applications et l'orchestration de services sur plusieurs serveurs. Il se distingue par sa simplicité : il ne nécessite pas l'installation d'agent sur les machines cibles, s'appuie sur des connexions SSH sécurisées et utilise des *playbooks* écrits en YAML, un langage lisible par l'humain, pour décrire les tâches à exécuter, ce qui en fait un outil accessible et puissant pour gérer des infrastructures à grande échelle.

Tous nos environnements Moodle sont hébergés au sein de notre DSI. Chaque environnement (Moodle Dev, Préprod et Sciences) a son propre serveur Ansible dédié pour des raisons de sécurité réseaux imposés par la DSI. Chaque environnement comprend quatre serveurs Moodle, gérés par Ansible via des connexions SSH. Les serveurs Ansible, indépendants les uns des autres, automatisent les tâches grâce à des *playbooks*.

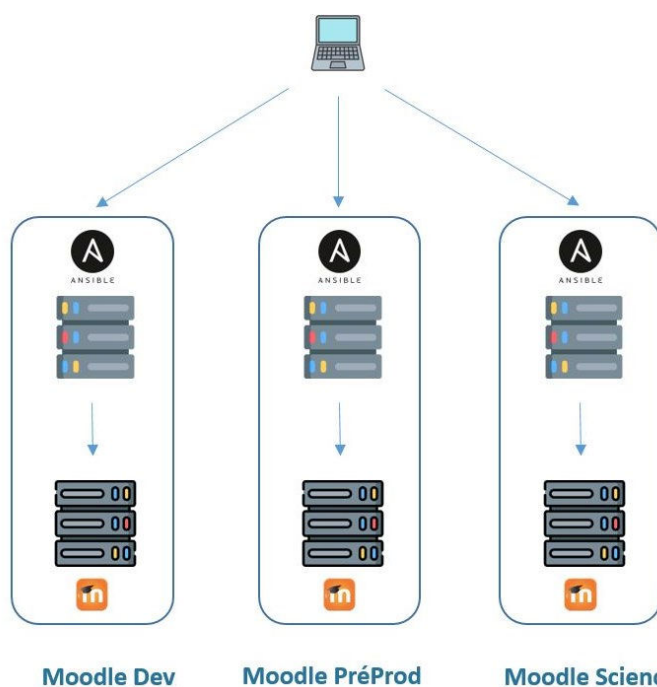


Figure 2: Architecture Ansible à CAPSULE

3.2 Avantages de l'infrastructure Ansible

- **Isolation des environnements** : chaque environnement (Dev, Préprod, Prod) est géré indépendamment, ce qui réduit les risques d'erreurs ou de conflits entre les environnements. Cela permet de tester les modifications dans Dev et Préprod avant de les appliquer à la production.
- **Sécurité renforcée** : les serveurs Ansible étant isolés pour chaque environnement, les erreurs dans un environnement n'affectent pas les autres. Les connexions via SSH assurent des communications sécurisées entre le serveur Ansible et les serveurs Moodle.
- **Facilité de gestion et d'automatisation** : chaque serveur Ansible peut gérer ses propres tâches spécifiques à son environnement via des *playbooks*, simplifiant l'automatisation et l'administration des serveurs Moodle.
- **Évolutivité et flexibilité** : nous permet d'adapter ou modifier un environnement (ajouter des serveurs, ajuster la configuration) sans impacter les autres. Cela permet d'expérimenter et de mettre à jour les environnements de développement ou de préproduction en toute sécurité.
- **Réduction des temps d'arrêt** : les environnements distincts permettent de maintenir la production en fonctionnement tout en apportant des changements ou en effectuant des tests dans les environnements de Dev et Préprod.

Cette structure garantit une meilleure organisation, une gestion centralisée et sécurisée, tout en minimisant les risques pour l'environnement de production.

3.3 Inconvénients de l'infrastructure Ansible

- **Complexité de gestion** : avoir trois serveurs Ansible distincts nécessite de maintenir et configurer chacun d'eux séparément, augmentant la charge administrative et le risque d'erreurs humaines.
- **Répétition des configurations** : des tâches identiques doivent être configurées dans chaque environnement (Dev, Préprod, Sciences), entraînant une duplication de travail et compliquant les mises à jour.
- **Coûts d'infrastructure** : la multiplication des serveurs (trois serveurs Ansible et douze serveurs Moodle, tous virtualisés) accroît les coûts en termes d'hébergement et de gestion des ressources.
- **Synchronisation des configurations** : assurer une cohérence totale entre les environnements peut être difficile, car chaque serveur Ansible fonctionne indépendamment, ce qui peut générer des divergences dans les configurations.
- **Temps de déploiement plus long** : les mises à jour doivent être appliquées séparément dans chaque environnement, ce qui rallonge le processus de déploiement.
- **Surcharge des administrateurs** : gérer trois environnements Ansible distincts peut alourdir la charge de travail pour les administrateurs de la DSI.
- **Impact écologique accru** : l'utilisation de multiples serveurs consomme plus de ressources énergétiques, ce qui est moins écologique. Une infrastructure plus centralisée ou mutualisée pourrait réduire l'empreinte carbone et limiter la consommation d'énergie.

4 Le projet Ansible

4.1 Premiers Essais

Nous avons étudié les *playbooks* présents dans le projet buluma/ansible-role-moodle³. Cependant, l'installation sommaire de Moodle et l'absence de configuration adaptée à nos besoins ont rapidement conduit à son abandon.

Une première version a été conçue pour l'année universitaire 2023. Celle-ci permettait d'installer et de configurer l'infrastructure de base (un serveur Apache avec PHP) avec les outils annexes (Postfix, Unoconv, Mimetex) nécessaire au bon fonctionnement de Moodle sur l'ensemble des serveurs Web. Le *playbook* d'installation récupérait les sources de la plateforme depuis les dépôts Moodle.org.

Nous avons ajouté la possibilité de télécharger les plugins, soit depuis les dépôts de moodle.org, soit depuis le GitLab du centre CAPSULE, ainsi que de leur installation, comprenant le placement du répertoire du plugin au bon endroit dans l'arborescence de Moodle et la mise à jour de la base de données pour l'enregistrement. Nous avons également mis en place une configuration succincte de la plateforme et des plugins.

Cependant, le manque de modularité de la structure logique, où un seul *playbook* regroupait toutes les actions à effectuer, l'absence de standardisation des conventions de nommage des paramètres Moodle, ainsi qu'une organisation inadéquate des variables Ansible, nous ont conduit à repenser l'architecture du projet.

³<https://github.com/buluma/ansible-role-moodle>

4.2 Version actuelle

4.2.1 Architecture du projet

Le *playbook* de la première version du projet a été structuré en plusieurs fichiers, chacun correspondant à un *playbook* dédié à un rôle spécifique. Ces fichiers sont nommés en fonction de l'équipe responsable (développement ou système) et des éléments installés ou configurés. Tous les *playbooks* sont situés à la racine de l'arborescence pour garantir un accès rapide et uniforme aux différentes sections, en utilisant des chemins système relatifs constants. Cette organisation facilite l'ajout, la suppression ou la mise à jour des éléments à installer ou à configurer, tout en simplifiant la maintenance.

L'organisation des variables par *playbook* a été abandonnée pour deux raisons principales. Premièrement, certaines variables, comme la version de PHP, étaient utilisées dans plusieurs *playbooks*, entraînant soit des duplications inutiles dans les fichiers de variables, soit rendant cette organisation inefficace. Deuxièmement, leurs mises à jour liées aux nouvelles années ou aux changements d'environnement exigeaient de vérifier chaque fichier, augmentant le risque d'erreurs et d'oubli. Désormais, les variables sont regroupées en quatre fichiers dans le dossier */vars* selon les événements qui nécessitent leur mise à jour :

- *constant.yml* : pour les variables qui changent rarement, voire jamais ;
- *annual.yml* : pour celles à mettre à jour chaque année universitaire ;
- *environment.yml* : pour les variables à modifier à chaque changement d'environnement (Dev, Préprod, ou Sciences) ;
- *instance.yml* : pour celles nécessitant des ajustements à la fois lors des changements d'année universitaire et d'environnement ;

Les sous-dossiers nommés *.archives* contiennent les fichiers de variables des autres années ou environnements. Ces fichiers peuvent être copiés dans les fichiers actuels pour être réutilisés dans les *playbooks*, simplifiant ainsi le processus de mise à jour tout en restant accessible à travers l'IDE. Grâce à une meilleure maîtrise d'Ansible acquise lors de la première version du projet, nous avons pu généraliser et factoriser certaines parties du code en utilisant des *blocks* Ansible. Ces portions de code, principalement dédiées à l'installation et à la configuration des plugins, sont organisées dans le répertoire */include*. Dans le même souci de maintenir la flexibilité et la modularité du projet, la logique de configuration de Moodle a été structurée dans les sous-dossiers *core_config* et *plugins_config*, facilitant ainsi l'ajout et le retrait de composants.

L'unique élément organisationnel n'ayant pas été modifié entre la version précédente et la version actuelle est le répertoire */templates*. Ce dossier contient les modèles Jinja utilisés par Ansible pour générer dynamiquement des fichiers de configuration en fonction des variables définies. Cette approche permet d'adapter les configurations des différents outils installés sur le serveur en fonction des exigences spécifiques du projet.

4.2.2 Fichier Hosts

Le fichier *hosts*, qui contient l'inventaire des machines sur lesquelles Moodle est déployé pour une instance donnée, a été réorganisé en réponse aux erreurs rencontrées avec l'architecture multi-VM de Moodle. En plus de définir des groupes de serveurs pour appliquer les tâches des *playbooks*, deux nouveaux groupes ont été ajoutés : le premier identifie le nœud principal de la base de données en replica, permettant d'exécuter des opérations sans perturber la réplication ; le second désigne la machine auxiliaire web, permettant d'effectuer des appels depuis Moodle vers sa base de données sans générer d'erreurs d'accès concurrentiel, ce qui est particulièrement utile lors des phases d'installation de la plateforme ou de ses plugins.

4.2.3 Le Git des sources Moodle

Hébergé sur le GitLab de CAPSULE, ce dépôt centralise l'intégralité du code source de la plateforme Moodle ainsi que celui de ses plugins. Les *playbooks* d'installation de la plateforme et des plugins s'appuient sur ce dépôt pour récupérer les sources nécessaires à leur déploiement. Cette organisation garantit une cohérence dans la récupération du code, qui auparavant était dispersé sur plusieurs plateformes. Cela simplifie l'écriture des *playbooks* d'installation et assure que toutes les instances de Moodle déployées sur une année donnée sont identiques et cohérentes.

4.3 Résultats et conclusion

L'utilisation d'Ansible pour l'installation et la configuration de Moodle a permis d'atteindre un niveau d'automatisation très élevé, avec environ 95 % du processus entièrement géré par des *playbooks*. L'installation de la quasi-totalité des composants nécessaires (serveur web, PHP et autres) ainsi que celle de Moodle ont été automatisées. Cependant, certains aspects, comme la configuration fine de certains plugins nécessite encore une intervention humaine. Ces 5 % restants sont liés aux éléments de configuration de la plateforme qu'il est impossible de faire en ligne de commandes, ou à des plugins dont la configuration n'est pas classique.

Ansible permet de nous assurer que chaque instance suit exactement le même processus d'installation, avec les mêmes versions de composants et les mêmes configurations en n'adaptant pour cela que des fichiers de variables externes.

Un autre résultat significatif est le gain de temps considérable. Le déploiement manuel d'une instance Moodle, incluant l'installation des composants, la configuration et l'ajout des plugins, pouvait auparavant prendre jusqu'à une semaine de travail intermittent. Grâce à l'automatisation, ce processus ne prend désormais que quelques minutes, le temps d'un café.

5 Évolutions

5.1 Serveur Ansible centralisé

Nous envisageons de centraliser la gestion des trois environnements (Dev, Préprod et Sciences) en utilisant un seul serveur Ansible. Cette approche permet de simplifier la gestion des tâches d'automatisation tout en évitant la duplication des serveurs.

Les principaux avantages de cette centralisation incluent :

- **Simplicité accrue** : la gestion des environnements est concentrée sur un seul serveur Ansible, réduisant ainsi la complexité administrative ;
- **Réduction de la duplication** : la configuration est centralisée, éliminant les redondances entre serveurs et améliorant la structuration des paramètres ;
- **Économie d'énergie** : la diminution du nombre de serveurs nécessaires permet de réduire les coûts et l'empreinte carbone ;
- **Flexibilité et évolutivité** : la gestion des variables permet d'adapter facilement les configurations aux différents environnements tout en maintenant une base commune.

En adoptant cette solution centralisée, nous optimisons la gestion des environnements tout en diminuant les coûts et l'impact écologique.

5.2 Gestion des inventaires

Chaque instance de Moodle (Dev, Préprod, Sciences) sera associée à un fichier d'inventaire spécifique. Lors de l'exécution des commandes, ces fichiers d'inventaire seront désignés à l'aide de l'argument *-inventory*.

5.3 Sélection des fichiers de variables

Modifier les fichiers de variables principaux à chaque changement d'environnement ou d'instance s'est révélé contraignant. À l'avenir, il est prévu de simplifier cette gestion en passant les paramètres d'environnement et d'année lors du lancement des *playbooks*, en utilisant l'argument `--extra-vars`.

Par exemple, pour déployer l'instance "Préprod 2025", il suffira de spécifier :

```
--extra-vars "env=preprod year=2025"
```

Ainsi, les fichiers de variables appropriés seront automatiquement sélectionnés, à savoir :

- `vars/environments/preprod.yml`
- `vars/years/2025.yml`
- `vars/instances/preprod2025.yml`
- `vars/constants.yml`

5.4 Utilisation de Moosh

Pour les 5 % de la configuration qui nécessitent encore une intervention manuelle, l'intégration de Moosh (MOOdle SHell)⁴ nous permettra d'automatiser certaines tâches qui n'étaient pas réalisables avec les commandes en ligne de Moodle comme, par exemple, la création et la gestion des rôles Moodle.

5.5 Gestion des *playbooks*

Actuellement, chaque *playbook* doit être exécuté manuellement, un par un, dans un ordre précis. Nous prévoyons de créer un *playbook* général qui regroupera les *playbooks* existants et les exécutera dans le bon ordre automatiquement, sans intervention humaine.

5.6 Conclusion

L'intégration d'Ansible dans nos processus de déploiement Moodle nous a permis de réduire significativement le temps y étant alloué et les risques d'erreurs manuelles. Malgré les difficultés rencontrées, les résultats sont positifs tout en ayant su garder une structure Ansible simple et accessible à tous, même aux néophytes. Les marges d'amélioration restent larges et nous sommes confiants de pouvoir arriver à une architecture adéquate pour le déploiement de la plateforme de l'année universitaire 2025.

⁴<https://moosh-online.com/>