



Laboratoire

Cahier des charges

Rédigé le 12/10/2022 par Adrien Bourrel , Corentin Bonafos, Liam Milor, Mathias Mirra
Dernière mise à jour: 17/11/2022

Version : 1

A destination de : Classe CDAN

Contexte

L'objectif principal est de pouvoir déployer une infrastructure de type entreprise au sein du laboratoire que l'ensemble des apprenant-es de l'IPI pourront utiliser pour faciliter l'avancement de leurs projets personnels ou tutorés.

Les classes de M2IL et M2RS portant une casquette de chef de projet seront chargées de proposer un cahier des charges aux classes de CDAN et ASRBD qu'ils devront analyser et solutionner.

Ce projet se déroulera sur plusieurs années, il devra donc être accessible, adaptatif et évolutif.

L'infrastructure ne doit pas impacter les activités de l'école ou être la cible d'attaques, sa sécurité est un aspect qu'il faudra absolument prendre en compte.

Explication des besoins

Projet 1 : Espace de gestion de VM

Lors de cours nécessitant une installation de système d'exploitation, les élèves utilisent des machines virtuelles (VM) qui doivent être installées sur les postes personnels. Dû à des configurations matérielles différentes pour chaque élève, il y a toujours des problèmes d'installation à résoudre engendrant de grosses pertes de temps.

L'objectif est de créer une interface permettant d'accéder à des VMs hébergées sur un serveur de l'école. Pour créer ces VMs, des ISO vierges ainsi que des copies de VM créées par les intervenants seront mis à disposition. En passant par cet espace les intervenants évitent tous les problèmes d'incompatibilités, les temps d'installations des systèmes d'exploitations et la surcharge du réseau du Campus.

Chaque élève aura accès à un stockage de 50Go sur un serveur où il pourra créer ses VMs et/ou installer les copies créées par les intervenants.

Projet 2 : Interface de code en ligne

Pour faciliter l'initiation au développement et les tests de petit bout de code nous aimerions créer une interface web dans lequel il serait possible de rédiger du code puis de l'exécuter. Cela permettra à l'étudiant de se concentrer sur le code sans se soucier de l'installation des langages ou de l'IDE.

Il sera possible de coder dans les langages les plus populaires (python, javascript, java, C++, C#).

L'étudiant aura accès à un dossier dans lequel il pourra créer ses fichiers de code, ces fichiers devront être sauvegardés dans le cloud à chaque modification pour ne jamais perdre son travail.

L'intervenant devra pouvoir partager du code aux étudiants directement depuis cette interface pour des corrections d'exercice par exemple.

Autres besoins

Ces deux outils devront être accessibles depuis un portail et nécessitera un compte pour y accéder.

Contraintes

Déploiement de système d'exploitation sur les postes similaires à ceux utilisés en entreprise.

Le portail :

- affiche les outils disponibles
- nécessite de s'y connecter
- possède deux rôles utilisateurs: basique et admin
- le rôle admin permet l'ajout de lien vers de potentiel nouveaux outils et l'ajout d'utilisateurs

- Projet 1 : Espace de gestion au VM :
 - installation de plusieurs systèmes d'exploitation utilisés en entreprise et lors d'apprentissage
 - création de VMs
 - limite de 3 VMs lancées en même temps
 - stockage des VMs sur un espace utilisateur (50 Go maximum)
 - communication des VMs entre VM possible pour les VMs d'un élève
 - exportation des VMs réalisées
 - importation de VMs depuis celles exportées
 - suppression des VMs
 - importation de VM par l'intervenant intervenant
 - les apprenants peuvent importer sur leurs espaces les VMs déposées par les intervenants

- Projet 2 : Interface de code en ligne :
 - une solution web qui permet l'exécution du code en cloud sur le serveur.
 - une sauvegarde en direct dans le cloud
 - compte utilisateur pour récupérer les différentes données
 - un accès intervenant pour accéder au codes des élèves avec une possibilité de correction du code
 - une organisation claire des différents fichiers de développement
 - une console pour l'affichage de réponse après exécution du code
 - cette plateforme doit permettre l'utilisation de plusieurs langages
 - affichage d'une aide et correction de code comme dans un IDE
 - Gestionnaire de versionnement :
 - authentification

- interface web (exemple : GitLab)
- accessible en local

Planning

Document à rendre par projet

24/11 → Première analyse du cahier des charges

15/12 → Analyse avancée du cahier des charges / Conception 1 (Diagramme UML ou/et autre)

19/01 → Conception 2

23/02 → Planning de développement

16/03 → Conception 3

13/04 → Rapport d'avancement du projet

20/04 → Rédaction compte rendu

22/06 → Rédaction compte rendu

20/07 → Oral présentation