
1 INFORMATIONS GÉNÉRALES :

Élèves	Travail de groupe (4 personnes)
Lieu de travail :	Ecole Technique - Ecole des Métiers de Lausanne
Période de réalisation :	8 semaines
Nombre de périodes :	3 périodes par semaine

2 PROCÉDURE :

- Le groupe d'élèves réalise un travail sur la base du cahier des charges décrit au **point 6**. Ce dernier est en outre présenté, commenté et discuté avec les élèves en classe.
- Le groupe d'élèves sont entièrement responsables de la sécurité des données ainsi que la mise en place de sauvegardes.
- A la fin du délai imparti pour la réalisation du projet, le groupe devra livrer le projet au complet comprenant les sources, un journal de travail ainsi qu'un rapport.

3 TITRE :

Développement d'un serveur Web en C# console.

4 MATÉRIEL ET LOGICIELS À DISPOSITION :

- Une machine virtuelle en Windows Server 2019.
- Une machine virtuelle en Windows 10.
- Une machine virtuelle Linux, distribution au choix.
- Document ConventionsDeCodage.pdf.

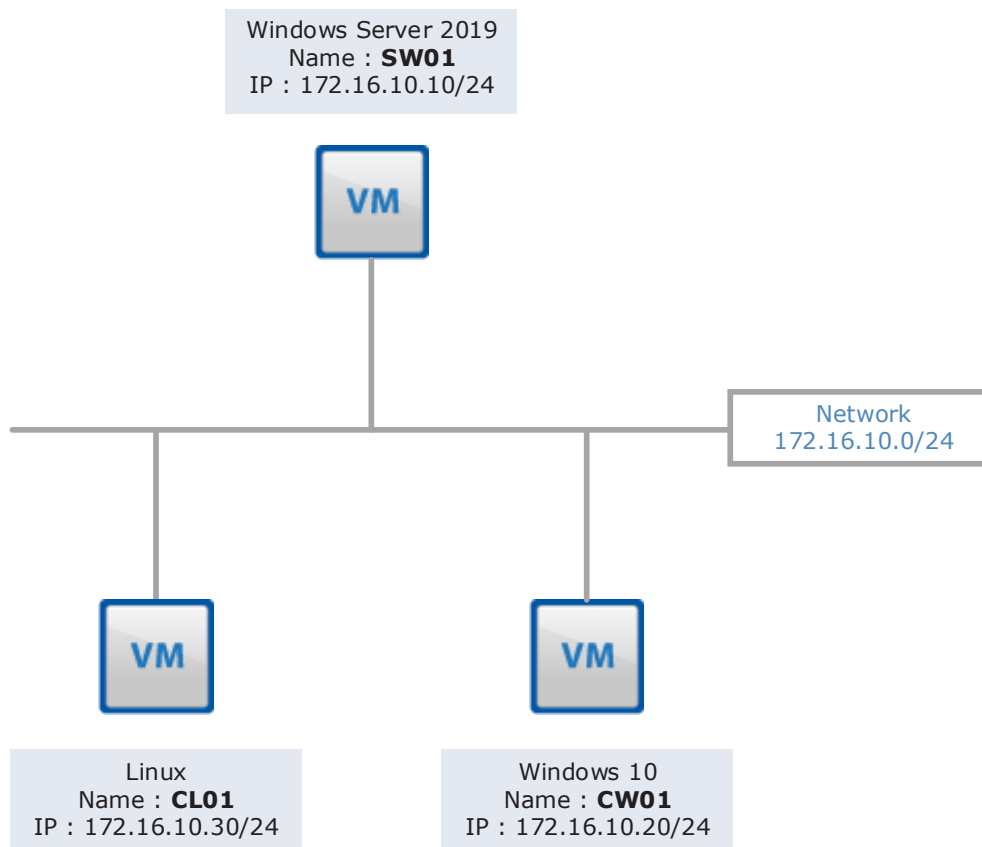
5 CONSIGNES :

- **Travailler dans le calme et limiter les distractions.**
- Toute source d'information externe doit être mentionnée. [URL Web, titre et page des livres, enseignant, collègues, etc...] à mettre dans le rapport.
- Le travail est réalisé par un groupe de **4 élèves**.
- Respecter les conventions de codage et nommage de l'ETML en vous référant au document **ConventionsDeCodage.pdf**.
- Réaliser un journal de travail de groupe sans planification (en annexe du rapport).
- Réaliser un Trello contenant toutes les tâches de votre travail (en annexe du rapport).
- Réaliser un rapport au format Word et PDF contenant les éléments suivants :
 - Texte justifié et police en Verdana.
 - Une page de garde avec un titre.
 - Entête et pied de page (nom du groupe, nombre de pages, date de création et de modification).
 - Table des matières,
 - Introduction.
 - Réalisation.
 - Conclusion.
- Déposer une archive au format zip selon les indications de votre enseignant.

6 DESCRIPTIF DU PROJET :

1. Architecture virtuelle :

Monter l'architecture virtuelle en respectant le schéma ci-dessous :



2. Développer un serveur Web en C# console :

En effectuant des recherches sur Internet (sans oublier de citer les sources), vous avez la possibilité de vous inspirer de milliers d'exemples pour réaliser un serveur Web en C# console.

Ce dernier devra incorporer les fonctionnalités suivantes :

2.1. Accès au serveur Web en utilisant les URL's ci-dessous :

- a) `http://localhost:8081` (`http://127.0.0.1:8081`) => depuis votre serveur
- b) `http://localhost:8081/votrePage.html` => depuis votre serveur
- c) `http://adresse_ip_vm:8081` => depuis votre serveur et vos clients
- d) `http://adresse_ip_vm:8081/votrePage.html` => depuis votre serveur et vos clients

2.2. Si la page n'existe pas, il faut rediriger sur une page d'erreur 404.

2.3. Mettre en place un fichier de configuration (*.config / *.json) afin de paramétrer l'IP et le port d'écoute de votre serveur Web (ne pas mettre ces informations en dur directement dans votre code).

2.4. Au démarrage de votre programme console, il faut afficher les messages suivants :

- a. [Web Server Running @IP:8081], remplacer @IP par l'adresse IP de votre machine.
- b. Afficher les informations du client qui se connecte à votre serveur Web, par exemple [Client @IP:Port URL], en remplaçant les données :
 - @IP par l'adresse IP du client.
 - Port le port du client.
 - URL par l'URL entrée par l'utilisateur.

2.5. Pages Web : au moins 3 pages html :

- a) Une simple page d'index et page de votre choix, affichant :
 - Un titre
 - Un texte de votre choix avec une image.
- b) Une page 404 affichant :
 - Un titre
 - Un texte contenant un message d'erreur.

2.6. Un point important consiste à loguer dans un fichier texte toutes les actions et informations de votre serveur au format [DD/MM/YYYY hh:mm:ss] [log level] [message]. Exemple :

- a) [DD/MM/YYYY hh:mm:ss] [INFO] [Web Server Running @IP:8081].
- b) [DD/MM/YYYY hh:mm:ss] [INFO] [Access client @IP:Port/URL].
- c) [DD/MM/YYYY hh:mm:ss] [INFO] [Send File XYZ to client @IP:Port].
- d) [DD/MM/YYYY hh:mm:ss] [ERROR] [Error 404 @IP:Port/URL not found].
- e) Etc.

2.7. Le serveur Web devra être accessible depuis une VM cliente Windows et Linux.

2.8. Tester votre application avec les navigateurs Chrome, Edge et Firefox.

3. Le rapport :

Il faut incorporer les éléments ci-dessous en justifiant les points par des captures d'écran dans la partie de réalisation :

3.1. Décrire votre application et vos pages Web.

3.2. Réaliser un diagramme de communication HTTP lors d'une connexion entre un client et le serveur (voir exemple en page 4). En vous basant sur vos logs, il faudra y indiquer à chaque communication l'**@IP:Port** source et l'**@IP:Port** de destination.

3.3. Décrire la configuration concernant l'ouverture de port de votre serveur.

3.4. Démontrer le bon fonctionnement de votre serveur Web depuis le réseau local et les clients.

3.5. Réaliser des tests de connexion au serveur Web depuis les clients Windows et Linux.

3.6. Contrôler que votre serveur Web est en **écoute** sur port **8081** en vous aidant de la commande **netstat** :

- a) Protocole / Adresse locale et Port / Adresse distante et Port / Etat.
- b) Décrire et expliquer ce que vous observez.

3.7. Contrôler la connexion à votre serveur Web en vous aidant de la commande **netstat** (depuis les clients Windows et Linux) :

- Protocole / Adresse locale et Port / Adresse distante et Port / Etat.
- Décrire et expliquer ce que vous observez.
- Décrire le résultat si le firewall de votre serveur est actif et que le port **8081** n'est pas ouvert.

- Exemple de diagramme de communication :

