

## **Déroulement de la validation du projet DevOps**

- ★ La validation du projet DevOps se déroule en mode distanciel, avec la caméra activée. Il s'agit d'un projet individuel. Chaque étudiant devra disposer de sa propre solution sur sa propre machine.
- ★ Il est important de noter que **la note obtenue** dans le cadre de ce projet représente **100 % de la moyenne du module**.
- ★ Chaque groupe devra ajouter son enseignant au référentiel Git de son groupe en utilisant l'adresse e-mail de l'enseignant.

### **A. Procédure d'évaluation :**

1. Se connecter à la réunion Meet (invitation déjà envoyée par votre enseignant) en suivant l'ordre de passage précédemment communiqué, de manière individuelle, à partir de 09h. Les groupes peuvent permuter entre eux. Veuillez accéder à la réunion Meet uniquement lorsque le **groupe** précédent a quitté la réunion.
2. Présenter à l'enseignant votre adresse MAC (`ipconfig /all` puis recherchez la ligne Adresse physique pour récupérer l'adresse MAC sous le format `xx-xx-xx-xx-xx-xx`) ainsi que votre carte d'identité, car il s'agit d'un examen.
3. Préparer en avance tous les outils nécessaires, tels que les machines virtuelles, Docker, Jenkins, Nexus, SonarQube, Grafana, Prometheus, etc.
4. Ouvrir les interfaces graphiques des outils et plateformes, y compris les IDE, GitHub, DockerHub, SonarQube, Nexus, Jenkins, Grafana, Prometheus, et autres.
5. Ouvrir les fichiers pertinents, notamment les fichiers Dockerfile(s), docker-compose.yml, les fichiers de mock, etc.
6. **L'étudiant doit lancer le pipeline complet 30 minutes avant son passage.** La réalisation du pipeline est un travail individuel, et il est strictement interdit d'utiliser le même code pour tous les membres de l'équipe.

7. Veuillez modifier la date système d'Ubuntu afin de bien vérifier l'horaire de lancement de pipeline avant la validation en utilisant la commande suivante dans le terminal :
- ```
sudo date --set "YYYY-MM-DD HH:MM:SS"
```

Assurez-vous de remplacer YYYY-MM-DD par la date et HH:MM:SS par l'heure souhaitées.

## **B. Configuration de la machine :**

Le pipeline doit inclure les étapes suivantes :

1. Récupération du projet depuis GitHub/GitLab.
2. Exécution des tests unitaires avec l'utilisation de Mockito et JUnit.
3. Évaluation de la qualité du code grâce à l'outil SonarQube.
4. Dépôt du livrable sur Nexus.
5. Construction de l'image Docker (pour la partie Spring) à partir du fichier Dockerfile.
6. Publication de l'image créée sur Docker Hub ou Nexus.
7. Démarrage simultané de l'image contenant le livrable Spring et de l'image MySQL en utilisant docker-compose.
8. Affichage d'un graphique de surveillance des services spécifiques, tels que Jenkins, l'application Spring Boot, ou la machine.
9. Test des services de l'application avec Postman ou Swagger (un simple ajout d'un élément, suivie d'une requête GET sur l'élément ajouté, est suffisant).

## **C. Notes :**

- Chaque étudiant sera interrogé sur différents outils et concepts liés au cours, notamment DevOps en général, l'intégration continue, la livraison continue, et le monitoring continu.
- Assurez-vous que le fichier **Dockerfile**, situé sur GitHub, se trouve dans votre propre branche. La réalisation du pipeline est un travail individuel, et il est interdit d'utiliser le même code que tous les membres de l'équipe.
- Il est impératif que le pipeline soit optimisé, avec un code clair, des étapes correctement ordonnées, et en évitant de lancer des étapes en double.
- Vous avez la possibilité de lancer simultanément les deux outils, Sonar et Nexus, en utilisant docker-compose et docker-volume, mais ce n'est pas obligatoire.

- La note attribuée à l'excellence sera décernée à ceux qui présentent des éléments non exigés par l'enseignant, comme l'ajout de nouveaux outils, de l'originalité, un envoi de courrier électronique personnalisé précisant la source d'une erreur dans le pipeline, la création d'un livrable Angular (bien que la création d'un pipeline pour la partie Angular ne soit pas obligatoire), des tests de l'application directement depuis l'interface du projet Angular, la surveillance de l'application Backend, etc.