

PLAN DE COURS

☒ **Saint-Jérôme**

☐ **Mont-Laurier**

☐ **Mont-Tremblant**

420-4D4-JR
Numéro du cours

Déploiement de serveurs Internet
Titre du cours

Hiver 2026
Session

1-3-2
Pondération

420-3C3-JR
Préalable(s)

Auteur(e)s ou professeur(e)s utilisant ce plan de cours :

Boudreault, Alain
Nom et prénom

6514
Poste

aboudrea@cstj.qc.ca
Courriel

D-125e
Bureau

Ce plan de cours contient les informations suivantes :

- Présentation du cours
- Description de la cible du cours
- Description de l'évaluation synthèse du cours incluant sa pondération et ses critères
- Démarche d'enseignement et d'apprentissage
- Outils pédagogiques utilisés
- Démarche d'évaluation
 - Formative
 - Sommative (type, pondération, moment)
- Calendrier du cours
- Activités obligatoires pour l'étudiant (s'il y a lieu)
- Modalités de participation au cours
- Règles départementales d'évaluation des apprentissages (s'il y a lieu)
- Référence à la PIEA :
 - Modalités de reprise en cas d'absence à une évaluation
 - Pénalité pour les retards dans la remise des travaux (s'il y a lieu)
 - Modalités d'évaluation de la langue
- Médiagraphie
- Matériel requis et frais (s'il y a lieu)
- Modalités de diffusion des disponibilités de l'enseignant
- Référence à la Politique institutionnelle sur la fraude, le plagiat et la tricherie par les étudiants

La coordination départementale recommande l'approbation de ce plan de cours au nom de l'assemblée départementale

Informatique
Nom du département

Cliquez ici
Coordination départementale (Nom)

Cliquez ici pour entrer
une date.
Date

Note aux étudiants :

Nous vous suggérons de conserver ce plan de cours afin de vous éviter des frais, si ultérieurement vous aviez à demander au collège une impression ou un envoi électronique de ce plan de cours.

Notez qu'il est primordial pour votre réussite, d'être présent physiquement et intellectuellement en classe. Vous êtes invité à informer votre enseignant en cas d'absence.

PRÉSENTATION DU COURS

Famille : Soutien aux utilisateurs et réseau

Ce cours permettra à l'étudiant, à partir de l'approche DevOps, de réaliser, de documenter et d'automatiser les cycles d'édition, de production et de distribution d'une application en utilisant des outils d'intégration et de déploiement continus (CI/CD).

DESCRIPTION DE LA CIBLE DU COURS

À la fin de ce cours, l'étudiant sera en mesure d'utiliser un outil de type orchestration de conteneurs, permettant d'automatiser les parties du développement logiciel liées aux builds, aux tests et au déploiement, et facilitant l'intégration et le déploiement continus (outils CI/CD).

DESCRIPTION DE L'ÉVALUATION SYNTHÈSE DU COURS

PONDÉRATION : 50 %

Objets d'évaluation

La capacité de l'étudiant à utiliser un outil de type orchestration de conteneurs, permettant d'automatiser les parties du développement logiciel liées aux builds, aux tests et au déploiement, et facilitant l'intégration et le déploiement continus (outils CI/CD).

Contexte de réalisation

Volet A

Pondération :

40/40

Tâche exigée de l'élève :

Travail pratique individuel - À partir d'un devis de réalisation fourni par l'enseignant, réaliser, documenter et automatiser les cycles d'édition, de production et de distribution d'une application à l'aide d'un outil d'orchestration de conteneurs.

Durée :

Entre 12 et 16 périodes (3 à 4 semaines) en classe et travail personnel

Matériel permis lors de l'évaluation :

Toute documentation permise

Critères d'évaluation :

Pondération

Rédaction correcte des manifestes de déploiement

25 %

Fonctionnement correcte du système déployé

25 %

Construction fonctionnelle de conteneurs personnalisés

10 %

Utilisation correcte d'une structure YAML

5 %

Utilisation correcte des Pods

5 %

Utilisation correcte des ConfigMaps

5 %

Utilisation correcte des secrets

5 %

Programmation judicieuse des ensembles de réplicas

5 %

Configuration adéquate des services réseaux et des équilibreurs des charges

5 %

Configuration judicieuse et consommation adéquate des volumes persistants

5 %

Détermination correcte de l'information à rédiger

5 %

DÉMARCHE D'ENSEIGNEMENT ET D'APPRENTISSAGE

- | | |
|--|---|
| <input checked="" type="checkbox"/> Exposé | <input type="checkbox"/> Présentation |
| <input checked="" type="checkbox"/> Démonstration | <input checked="" type="checkbox"/> Travail pratique en atelier |
| <input type="checkbox"/> Étude de cas | <input type="checkbox"/> Travail d'équipe |
| <input checked="" type="checkbox"/> Laboratoire dirigé | <input checked="" type="checkbox"/> Résolution de problème |
| <input checked="" type="checkbox"/> Lecture | <input type="checkbox"/> Discussion de groupe |
| <input type="checkbox"/> Mise en situation | <input type="checkbox"/> Autres |

OUTILS PÉDAGOGIQUES UTILISÉS

INFORMATION ET DOCUMENTATION	CANAL DE COMMUNICATION
Plan de cours	Léa
Énoncé d'évaluation et notes d'évaluation	Léa et Teams
Remise des évaluation	GitHub
Communications individuelles	Mio et Teams
Communications de groupe	Mio
Notes de cours, exercices, documentation	Teams et GitHub

DÉMARCHE D'ÉVALUATION

FORMATIVE

L'évaluation formative fait partie intégrante de l'apprentissage. Elle s'inscrit dans un processus continu qui informe l'étudiant sur ses apprentissages vis-à-vis de la compétence à atteindre. C'est pour cette raison qu'aucune note n'est comptabilisée. À titre d'exemple, elle peut prendre différentes formes dont : un jeu questionnaire, une simulation d'examen, un débat en classe, des réseaux de concepts, un plan de rédaction, etc.

SOMMATIVE

Évaluation	Type	%	Distribution de l'énoncé	Remise
TP1 – Docker	Travail pratique individuel	30	Semaine 4	Semaine 8
Examen de fin de session	En classe, 4 périodes	30	Semaine 15	Semaine 15
Épreuve synthèse	Travail pratique individuel	40	Semaine 11	Semaine 15

CALENDRIER DU COURS

Semaine	Contenus	Évaluations sommatives
01	<p>Docker – Introduction</p> <ul style="list-style-type: none"> • Installation : Docker sur un serveur Linux, mention de Docker Desktop • Utilisation de l'environnement de développement <ul style="list-style-type: none"> ◦ Utilisation de VS Code, module ssh de VS Code et de Git ◦ Emplacement de l'environnement sur le disque et accès fichiers hors de l'environnement • Introduction aux conteneurs <ul style="list-style-type: none"> ◦ Fonctionnement de conteneurs ◦ Différences et avantages par rapport aux machines virtuelles ◦ Utilisation de Docker Desktop ◦ Utilisation de Docker Hub, d'images et de différentes versions (tags) • Commandes de base (docker) <ul style="list-style-type: none"> ◦ Affichage des conteneurs (ps, ps -a) ◦ Création d'un conteneur (run -d --name nom -p interne:externe) ◦ Gestion d'un conteneur (start, restart, stop, rm) ◦ Gestion des images (images, rmi, system prune -a) ◦ Débogage d'un conteneur (logs, exec -it, cp) 	
02	<p>Docker – Docker Compose</p> <ul style="list-style-type: none"> • YAML <ul style="list-style-type: none"> ◦ Comparaison et conversion entre JSON et YAML ◦ Syntaxe de base (indentation, commentaires, valeurs, listes) • Fichier Docker Compose <ul style="list-style-type: none"> ◦ Création d'une architecture de dossiers par conteneur ◦ Création de conteneurs multiples (services, image, ports, environment) ◦ Persistance des données à l'aide de volumes (volumes) <ul style="list-style-type: none"> ▪ Fonctionnement des volumes des conteneurs (temporaire, mounts) ▪ Ajout de fichiers ou dossiers locaux (bind mount) ▪ Ajout de volumes persistants pour le contenu des conteneurs (volume mount) ▪ Gestion des volumes Docker (docker volume) ◦ Réseaux entre conteneurs (networks) <ul style="list-style-type: none"> ▪ Fonctionnement des réseaux et adressages entre conteneurs ▪ Ajout de réseaux privés entre conteneurs • Commande Docker Compose (docker compose) <ul style="list-style-type: none"> ◦ Création et exécution de conteneurs (up, stop, start, restart, down) ◦ Débogage multi-conteneurs (ps -a, logs, exec -u root -it bash) 	

03	<p>Docker – Dockerfile</p> <ul style="list-style-type: none"> • Dockerfile <ul style="list-style-type: none"> ◦ Syntaxe d'un Dockerfile (<code>#</code>, <code>FROM</code>, <code>WORKDIR</code>, <code>COPY</code>, <code>RUN</code>, <code>EXPOSE</code>, <code>CMD</code>) ◦ Fonctionnement de la construction des images à l'aide de la cache ◦ Construction d'une seule image (<code>docker build --no-cache</code>) • Construction d'un Docker Compose avec conteneurs personnalisés <ul style="list-style-type: none"> ◦ Syntaxe Docker Compose (<code>build</code>, <code>context</code>, <code>target</code>) ◦ Nomenclature et organisation des Dockerfile (ex. <code>Dockerfile.frontend</code>, <code>Dockerfile.backend</code>) ◦ Construction des images (<code>docker compose up -d --build --no-cache</code>) • Construction d'une application multi-conteneur <ul style="list-style-type: none"> ◦ Utilisation de conteneurs de bases de données ◦ Utilisation de conteneurs Linux <code>alpine</code> et du gestionnaire d'installation <code>apk</code> • Utilisation de fichiers pour la personnalisation des variables d'environnement <ul style="list-style-type: none"> ◦ Syntaxe d'un fichier d'environnement <ul style="list-style-type: none"> ▪ Déclaration de variable dans le fichier <code>CLE=valeur</code> ▪ Utilisation dans Docker Compose <code>\${CLE}</code> ◦ Utilisation du fichier d'environnement par défaut (<code>.env</code>) ◦ Utilisation de fichiers d'environnement multiples (<code>--env-file</code>) 	
04	<p>Docker – Dépôt d'image</p> <ul style="list-style-type: none"> • Notions de dépôts d'images Docker <ul style="list-style-type: none"> ◦ Notions de registre (<code>registry</code>) ◦ Notions de dépôt d'images publiques et privés (<code>repository</code>) ◦ Notions de gestion des versions (<code>tags</code>) • Utilisation d'un dépôt privé pour les images (https://hub.docker.com) <ul style="list-style-type: none"> ◦ <code>docker login</code> ◦ <code>docker tag</code> ◦ <code>docker push</code> ◦ <code>docker pull</code> • Notions de sécurité et vulnérabilité des images (<code>docker scout</code>) 	Énoncé du TP1 – Docker
05	Travail sur le TP1	

06	<p>Kubernetes – Déploiement local</p> <ul style="list-style-type: none"> • Installation : Kubernetes sur un serveur Linux (Un seul nœud et plusieurs nœuds), mention de Docker Desktop • Introduction à Kubernetes (K8s) <ul style="list-style-type: none"> ○ Notions d’outils d’orchestration de conteneur ○ Notions d’architecture Kubernetes <ul style="list-style-type: none"> ▪ Architecture <ul style="list-style-type: none"> • Amas (cluster), nœud (node), Pod (conteneur), image ▪ Control plane (master) <ul style="list-style-type: none"> • API server, Controller manager, Scheduler, etcd ▪ Data plane (ouvrier) <ul style="list-style-type: none"> • Kubelet, proxy • Création des ressources à partir de manifestes (fichiers YAML) <ul style="list-style-type: none"> ○ Syntaxe commune <ul style="list-style-type: none"> ▪ kind, name, spec, selector, --- ○ deployment <ul style="list-style-type: none"> ▪ name, image, port, replicas ○ service <ul style="list-style-type: none"> ▪ protocol, port, targetPort, type (ClusterIP, NodePort) • Commandes de base Kubernetes (kubectl) <ul style="list-style-type: none"> ○ get, describe, apply -f, delete -f, logs, exec -it • Utilisation d’images Docker pour le déploiement dans Kubernetes 	
07	<p>Kubernetes – Déploiement local (suite)</p> <ul style="list-style-type: none"> • Construction d’une application locale multi-conteneurs <ul style="list-style-type: none"> ○ Utilisation de volumes (PersistentVolume, PersistentVolumeClaim) ○ Utilisation de ConfigMap ○ Utilisation de Secret et chiffrement des données (Base64) 	
Semaine de relâche		
08		Remise du TP1 – Docker (30%)

09	<p>Kubernetes – Services infonuagiques, routes et haute disponibilité (high availability)</p> <ul style="list-style-type: none"> • Fournisseurs de services infonuagiques principaux (AWS, Azure, GCP) <ul style="list-style-type: none"> ○ Notions des différents services offerts en ligne <ul style="list-style-type: none"> ▪ Bare metal ▪ Machines virtuelles (Compute Engine) ▪ Conteneurs (Cloud Run, GKE) ▪ Données (Cloud Storage, Persistent Disk) ○ Utilisation d'un estimateur de coûts de service infonuagique • Utilisation de Google Kubernetes Engine (GKE) <ul style="list-style-type: none"> ○ Mise en place d'un cluster <ul style="list-style-type: none"> ▪ 1 cluster, 2 nœuds (n1-standard-2, Preemptible VM) ○ Utilisation de la ligne de commande Google Cloud (gcloud) ○ Utilisation du Google Artifact Registry • Notions d'accès aux conteneurs par routes et adresses publiques <ul style="list-style-type: none"> ○ Utilisation d'un équilibreur de charge (LoadBalancer) ○ Utilisation d'un gestionnaire de routes (Ingress) <ul style="list-style-type: none"> ▪ GCE, nginx (reverse-proxy) ○ Utilisation d'un DNS public (https://www.duckdns.org/) • Notions de haute disponibilité (high availability) <ul style="list-style-type: none"> ○ Utilisation de configurations de haute disponibilité (HorizontalPodAutoscaler) 	
10	<p>Kubernetes – Gestion d'environnement multiples</p> <ul style="list-style-type: none"> • Notions d'environnements multiples (local, dev, prod) • Utilisation de Google Kubernetes Engine (GKE) <ul style="list-style-type: none"> ○ 1 cluster dev, 1 nœud (n1-standard-4, Preemptible VM) ○ 1 cluster prod, 1 nœud (n1-standard-4, Preemptible VM) • Notions d'Infrastructure as Code (IaC) • Utilisation d'un outil de gestion d'Infrastructure as Code (Terraform) <ul style="list-style-type: none"> ○ Installation de Terraform ○ Syntaxe des fichiers Terraform (.tf) ○ Création de configurations pour plusieurs environnements (local, dev, prod) 	

11	<p>Kubernetes – Configuration de déploiements par environnement</p> <ul style="list-style-type: none"> Utilisation d'un outil de gestion de configurations de déploiements (Helm) <ul style="list-style-type: none"> Installation de Helm Utilisation de la ligne de commande Helm (repo, install, upgrade, uninstall) Syntaxe et modification de chartes existantes Création de chartes selon différents environnements (local, dev, prod) 	Énoncé de l'Épreuve synthèse
12	Travail sur l'Épreuve synthèse (40%)	
13	Travail sur l'Épreuve synthèse (40%)	
14	<p>Kubernetes – DevOps et automatisation</p> <ul style="list-style-type: none"> Notions de DevOps <ul style="list-style-type: none"> Notions de Constant Integration (CI) et Continuous Delivery (CD) Mise à jour d'une application en production <ul style="list-style-type: none"> Tests unitaires, tests d'intégration tests, déploiement Automatisation avec GitHub Actions <ul style="list-style-type: none"> Syntaxe et utilisation des GitHub Actions Actions existantes du Marketplace Automatisation de la création des images et déploiement sur Google Cloud Notions de sécurité <ul style="list-style-type: none"> Permissions des comptes et compte d'accès utilitaire pour GitHub Gestion des secrets de façon sécuritaire Notions de sauvegarde des données <ul style="list-style-type: none"> Utilisation de Persistent Disk pour le backup des données Gestion et restauration des points de sauvegarde Documentation du déploiement d'une application (Markdown) 	
15	Travail sur l'Épreuve synthèse (50%)	Remise de l'Épreuve synthèse (40%)
Période d'évaluation communes		

Veuillez noter que des changements pourraient être apportés à cette grille en cours de session !

ACTIVITÉS OBLIGATOIRES POUR L'ÉTUDIANT (S'IL Y A LIEU)

Aucune activité spécifique en dehors des périodes de classes.

MODALITÉS DE PARTICIPATION AU COURS

La classe est un lieu d'apprentissage privilégié. Par sa présence et sa participation active en classe aux activités pédagogiques proposées, l'étudiant contribue à sa propre réussite. C'est pourquoi l'étudiant a la responsabilité d'assister à ses cours et de prendre les mesures nécessaires pour reprendre les apprentissages faits en classe en cas d'absence à un cours. Aucun retard ne sera toléré : dès que le cours est commencé et que la porte du laboratoire est fermée, l'élève devra attendre la pause ou le prochain cours pour réintégrer la classe. Les périodes de disponibilité de l'enseignant ne seront pas consacrées à des périodes de rattrapage en cas de retard ou d'absence au cours.

Au Cégep de Saint-Jérôme, la réussite de nos étudiantes et de nos étudiants nous tient à cœur. C'est pourquoi on te propose différentes mesures d'aide adaptées à tes besoins, afin que tu atteignes les objectifs que tu t'es fixés. Les mesures et les outils sont disponibles sur le portail *Je gère ma réussite* accessible via le code QR suivant :



Pour le bon déroulement de certaines activités d'apprentissage, il peut être nécessaire que l'étudiant ait préalablement accompli une tâche spécifique (devoir, lecture, questionnaire, etc.). L'enseignant peut **exclure** momentanément de la classe l'étudiant qui n'a pas réalisé ladite tâche et inviter ce dernier à l'accomplir à l'extérieur. L'étudiant réintègre la classe lorsqu'il fait la démonstration qu'il s'est suffisamment préparé pour participer à l'activité en question et recevoir la rétroaction formative associée à celle-ci. L'étudiant qui refuse de participer à l'activité et qui ne désire pas de rétroaction est libre de réintégrer la classe **quand l'enseignant juge** que les activités d'enseignement et d'apprentissage reliées à cette tâche spécifique sont terminées.

Afin de préserver un climat de classe propice aux apprentissages et respectueux des personnes, les dispositions suivantes appuyées par le Règlement sur le code de vie au Collège (Règlement 14) s'appliquent :

- L'usage du cellulaire est interdit en classe, sauf si cet usage se fait dans le cadre d'activités d'apprentissage encadrées par l'enseignant ;
- L'utilisation du matériel informatique ou de communication de manière inappropriée ou à des fins autres que la participation au cours est interdite ;
- Tout comportement qui entrave le déroulement du cours ou qui nuit au climat de classe est interdit.

L'enseignant peut exclure du laboratoire sur le champ quiconque contrevient à l'une ou l'autre de ces dispositions (Règlement sur le code de vie au Collège, art. 6.2). Par ailleurs, l'intégralité du Règlement sur le code de vie au Collège s'appliquent dans les cours d'informatique. L'élève a la responsabilité d'en prendre connaissance et de s'y conformer.

La règle du Collège au sujet de la nourriture dans les laboratoires informatiques est simple : c'est interdit. Toutefois, dans les laboratoires informatiques, les enseignants du département d'informatique tolèrent les breuvages s'ils sont dans un contenant hermétique seulement.

RÈGLES DÉPARTEMENTALES D'ÉVALUATION DES APPRENTISSAGES (S'IL Y A LIEU)

Le département d'informatique applique fidèlement la politique institutionnelle d'évaluation des apprentissages (PIEA).

MODALITÉS DE REPRISE EN CAS D'ABSENCE À UNE ÉVALUATION — PIEA ARTICLE 5.7

L'étudiant qui s'absente à une évaluation peut obtenir une reprise selon des modalités équivalentes dans la mesure où la raison de cette absence est **justifiée et jugée valable par son enseignant**. Son enseignant pourrait d'ailleurs lui demander de fournir une pièce justificative avant d'autoriser la reprise.

En cas d'absence pour une raison imprévue, notamment : maladie, décès d'un proche, accident ou raisons familiales, l'étudiant doit aviser l'enseignant dans un délai de **24 heures**. Sans raison valable, aucune reprise ne sera permise et la note zéro sera attribuée pour l'évaluation.

Dans le cas d'une absence prévisible (comparution à la Cour, rendez-vous avec un spécialiste médical, Alliance Sport-Études et Cheminots), l'étudiant doit aviser son enseignant au moins **5 jours ouvrables avant** la tenue de l'évaluation afin de planifier avec lui des modalités de reprise. Si l'étudiant n'informe pas son enseignant avant la tenue de l'évaluation, aucune reprise ne sera permise et la note zéro sera attribuée pour l'évaluation.

PÉNALITÉ POUR LES RETARDS DANS LA REMISE DES TRAVAUX (S'IL Y A LIEU) — PIEA ARTICLE 5.6

Toute évaluation sommative remise en retard fera l'objet d'une pénalité de 10 % de la valeur de l'évaluation par bloc de 24 heures de retard, incluant les fins de semaine et la semaine de relâche, à moins qu'une entente ne soit conclue entre l'enseignant et l'étudiant.

Les modalités de remise des travaux sont spécifiées dans le plan de cours ou dans les consignes remises par l'enseignant. On entend par modalités de remise : date de remise, moyen utilisé (papier ou électronique), format utilisé (Word, PDF, etc.), lieu de dépôt (classe, casier, bureau, etc.). Un travail qui n'est pas remis selon les modalités exigées pourra être refusé par l'enseignant. Dans un tel cas, les pénalités pour les retards s'appliquent.

Un travail sera refusé s'il est remis par l'étudiant alors que les autres étudiants du même cours ont reçu leur copie corrigée ou le solutionnaire de l'évaluation. Dans un tel cas, la note zéro sera attribuée.

MODALITÉS D'ÉVALUATION DE LA LANGUE — PIEA ARTICLE 5.10

Le développement des compétences linguistiques, pendant les études collégiales, appartient à toutes les disciplines, à tous les services et à toutes les directions. En participant à ce développement, tous agissent comme modèles pour les étudiants. L'acquisition de compétences linguistiques contribue au développement personnel et professionnel des individus et à leur rayonnement dans la société. Elle facilite l'accès à l'emploi et aux études supérieures. Elle est aussi essentielle à l'exercice des responsabilités sociales. En conformité avec ces principes issus de la Politique institutionnelle de valorisation de la langue du Cégep, la qualité du français doit faire l'objet **d'une évaluation dans chacun des cours**, à l'exception des cours portant sur l'apprentissage d'une autre langue.

Un travail dont la langue écrite est jugée inacceptable peut être refusé par l'enseignant. Dans un tel cas, les pénalités pour les retards dans la remise des travaux s'appliquent.

Une pénalité est imposée pour les fautes, jusqu'à concurrence de 10 % de l'évaluation écrite ou orale, et ce, dans la mesure où l'étudiant a accès aux outils de révision linguistique.

MÉDIAGRAPHIE

DOCUMENTATION OBLIGATOIRE

Aucune.

DOCUMENTATION COMPLÉMENTAIRE

Aucune.

MATÉRIEL REQUIS ET FRAIS (S'IL Y A LIEU)

L'étudiant doit avoir en tout temps lors du cours :

- Une clé USB ou autre pour sauvegarder ses travaux
- Un compte sur les services suivants :
 - GitHub
 - Docker Hub
 - Google Cloud

L'étudiant doit avoir à partir de la mi-session :

- Projet Google Cloud avec les droits administrateurs couvrant les frais suivants par mois :
 - GKE
 - 100 heures de cluster
 - 200 heures de Node-time
 - n1-standard-4
 - Preemptible VM
 - Cloud Storage
 - Standard Storage
 - 100 GB

MODALITÉS DE DIFFUSION DES DISPONIBILITÉS DE L'ENSEIGNANT

Les périodes de disponibilités de l'enseignant sont affichées à la porte de son bureau et à l'adresse suivante :

<https://ve2cuy.github.io/4204d4/Documentation/Disponibilites-enseignant-h26.html>

Vous pouvez communiquer avec vos enseignants du département d'informatique par Teams ou par MIO (<https://cstj.omnivox.ca>).

Pour consulter les avis d'absence et de retard des enseignants : portail Omnivox et écran à l'accueil.

POLITIQUE DE FRAUDE ET PLAGIAT

Vous trouverez la politique institutionnelle du CSTJ sur la fraude et le plagiat en cliquant sur le lien suivant: [Politique institutionnelle sur la fraude, le plagiat et la tricherie par les étudiants](#)