

DEVOPS INTEGRATION CONTINUE

Filière





**PROGRAMME
DE LA FILIERE**

OBJECTIFS

- Mettre en place une plateforme d'intégration continue(PIC)
- Savoir utiliser des conteneurs
- Gérer des dépôts

Méthodes pédagogiques. Pour l'ensemble des stagiaires, le cours intégrera les suivantes :

- Alternance d'exercices, cas pratiques, QCM et de notions théoriques
- Evaluations

Moyens pédagogiques

- AJC met à la disposition de chaque stagiaire un accès à notre plateforme à distance ainsi qu'éventuellement les logiciels utiles dans le cadre de chaque module
- Les supports de cours seront remis via notre la plate-forme de téléchargement Quest et/ou AJC Classroom

METHODES ET MOYENS PEDAGOGIQUES

Informations concernant les classes virtuelles

- Pour les formations en classe virtuelle, avec @JC CLASSROOM, vous profiterez des mêmes possibilités et interactions avec votre formateur que lors d'une formation présentielle : votre formation se déroulera en connexion continue 7h/7.
- Vous pourrez échanger directement avec le formateur et l'équipe pédagogique à travers notre système de visioconférence, mais aussi grâce aux forums et chats présents dans @JC CLASSROOM.
- Votre formateur sera à même de vérifier l'avancement de votre travail et de vous évaluer à l'aide d'exercices et de cas pratiques. Cela lui permettra de vous apporter un suivi pédagogique et des conseils personnalisés pendant toute la durée de la formation.
- Notre équipe technique vous enverra les modalités de connexion (accès, identifiants, dates, heures et numéro de la hotline) par mail dès votre inscription.
- Si vous rencontrez un problème de connexion, vous pourrez joindre à tout moment (avant ou même pendant la formation) notre hotline assistance technique au 01 82 83 72 41 ou par mail (hotline@ajc-formation.fr)

PRE-REQUIS

- Des notions de Développement seraient un plus

PARTICIPANTS

- Consultants, Ingénieurs, Développeurs, Administrateurs....

LIEU

Distanciel

CERTIFICATION / ATTESTATION

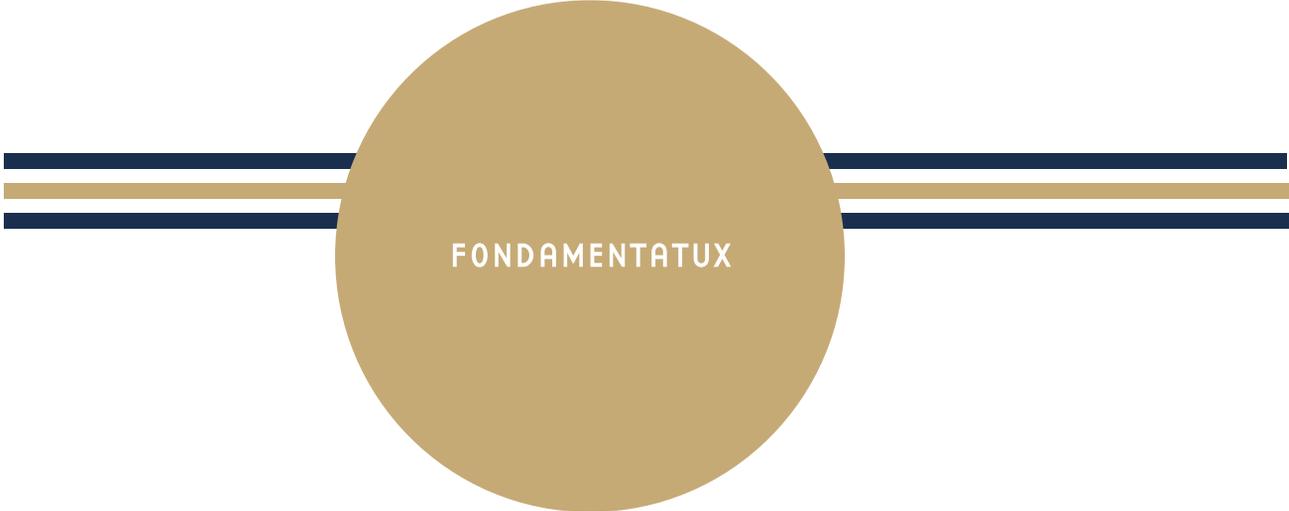
Attestation de formation

Programme - Contenu pédagogique

FONDAMENTAUX	FONDAMENTAUX UNIX	2 jour
DEVOPS ET AUTOMATISATION	DEVOPS	2 jours
	L'INGENIERIE DEVOPS SUR AMAZON WEB SERVICES	3 jours
	ANSIBLE	2 jours
INTEGRATION CONTINUE ET OUTILS	GESTION DES SOURCES AVEC GIT	1 jour
	GIT LAB-CI	1 jour
	DOCKER	2 jours
	JENKINS	2 jours
	AGILE TDD	1 jour
PROJET	PROJET FINAL	2 jours



PROGRAMMES
DÉTAILLÉS



FONDAMENTATUX



FONDAMENTAUX UNIX

PROGRAMME DU MODULE

Utilisation de la console et SSH

La gestion des paquets (yum/dnf, apt/aptitude, zypper)

**Les différents éditeurs (vi, vim, nano, pico, mc)
Configuration du réseau**

Gestion des services avec systemctl

Manipulation des fichiers et des répertoires (cp, mv, rm, etc)

Commandes de base, pipelining et redirecteurs

Gestion des utilisateurs et des droits

Les processus

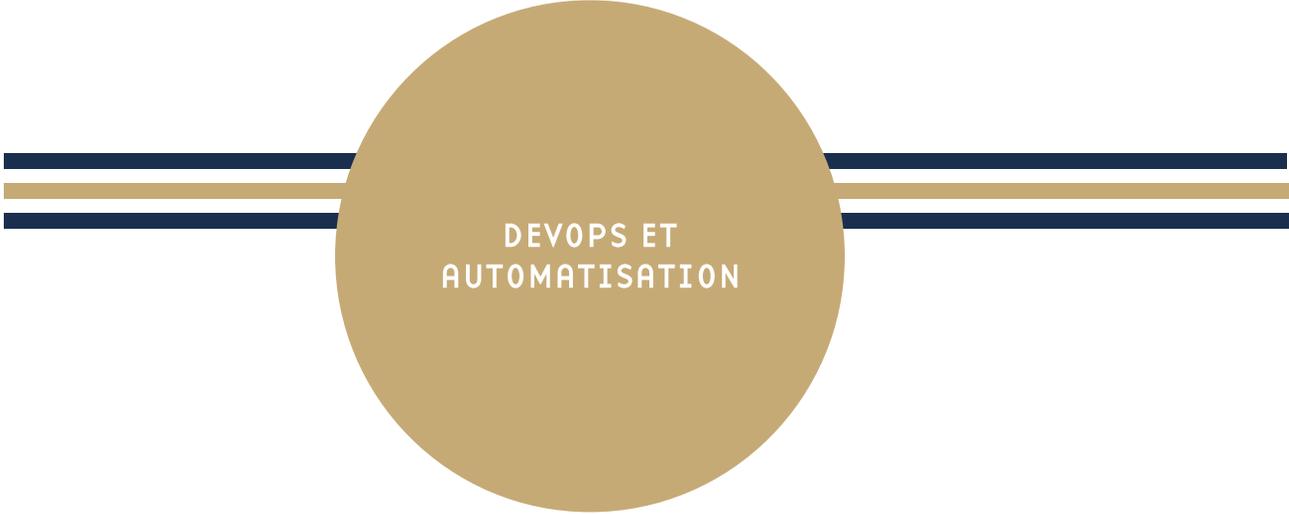
Sauvegarde et compression

Les journaux d'activités avec rsyslog

La planification de tâches avec Cron

OBJECTIFS

- Prise en main des éditeurs, des commandes de base et des tâches principales



DEVOPS ET
AUTOMATISATION

DEVOPS

2 jours,
14 heures



DISTANCIEL

PROGRAMME DU MODULE

Historique du mouvement DevOps, origines et influences

- Qu'est-ce que DevOps ?
- Planète DevOps : tendances et mouvements émergents
- Ansible
- Terraform
- Kubernetes
- ...

La conduite du changement

- DevOps, comment placer le curseur entre Dev et Ops ?
- Vers une convergence des métiers : changements organisationnels / culturels / technologiques
- Alignement des Dev aux réalités des Ops : rendre son application "prête pour la production"
- Alignement des Ops aux enjeux des Dev : intégration de la plateforme de production à l'usine logicielle

Processus et étapes éligibles à des fonctionnements DevOps

- Usine logicielle
- Livraison
- Déploiement
- Exploitation
- Troubleshooting

Bonnes pratiques pour un développement industriel

- Performance
- Sécurité
- Exploitabilité

Le rôle et choix des outils dans les organisations DevOps

- Puppet
- Chef

OBJECTIFS

- Acquérir la terminologie, la structure, les outils et les concepts de base de la démarche DevOps
- Identifier les problèmes liés à la communication inter-équipes
- Éliminer l'opposition entre les objectifs d'équipe
- Faire participer progressivement les développeurs aux opérations de production
- Industrialiser les déploiements applicatifs et industrialiser les opérations de gestion de l'infrastructure
- Connaître les différents outils DevOps du marché



L'INGENIERIE DEVOPS SUR AMAZON WEB SERVICES

PROGRAMME DU MODULE

Introduction

- En quoi consiste le DevOps?
- Pratiques et concepts principaux de la méthodologie

Conception

- Conception et sécurité
- Mise en place d'une infrastructure sur AWS avec un ou plusieurs projets Devops
- CloudFormation et gestion des applications

Infrastructure en tant que code

- Conception et sécurité
- CloudFormation et gestion des applications

Intégration

- Intégration continue dans le cloud.
- Livraison continue sur AWS
- Utilisation de Git
- Déploiements d'applications

Les différentes technologies de déploiement

- Choix de la technologie (AWS CodeDeploy, AWS OpsWork...)
- Analyse du scénario

Mise en œuvre des composants AWS

- Les possibilités des services EC2, S3, SQS, RDS et de leur modèle de programmation.
- Comment les applications Java peuvent-elles utiliser ces ressources ?

- Composants Amazon AWS avancés.
- Les services Virtual Private Cloud, Beanstalk et leur modèle de programmation.
- Exercice
Création de ressources virtuelles avec les services EC2, S3, SQS, RDS. Déploiement d'une application Java.

Industrialisation et pratiques DevOps

- Besoin de traçabilité, de répétabilité. Infrastructure as Code (IaC).
- Bonnes pratiques et outils disponibles.
- Définition et utilité des architectures Cloud avec des instances basées sur les rôles.
- Mécanisme Cloud-Init et notion de UserData.
- Exercice
Création de scripts et exécution "tracée et contrôlée".

OBJECTIFS

- Utiliser les pratiques et concepts principaux de la méthodologie DevOps
- Concevoir et mettre en place une infrastructure sur AWS, qui prend en charge un ou plusieurs projets de développement du DevOps
- Configurer Git sur AWS et découvrir l'éventail d'options permettant de mettre en place un environnement d'intégration continue sur AWS
- Utiliser les principes essentiels de l'intégration et du déploiement continu

ANSIBLE

2 jours,
14 heures



DISTANCIEL

PROGRAMME DU MODULE

Positionnement de Ansible

- Ansible et DevOps.
- Devops & IaC (Infrastructure as Code), le code source de l'infrastructure.
- Outils Puppet, Chef, Saltstack... Ansible.
- Fonctionnement d'Ansible.
- Architecture : inventaire, modules, playbooks, tasks, rôles.

Installation et configuration

- Installation et prise en main de l'outil.
- Les commandes de base d'Ansible.
- Configuration des noeuds : clés ssh, escalade de privilèges sudo.
- Le fichier de configuration.
- L'inventaire : création et utilisation.

Travaux pratiques

- Installation d'Ansible et configuration de plusieurs noeuds clients, création de l'inventaire et utilisation des premières commandes.

Les commandes Ad-Hoc

- Parallélisme et commandes Shell.
- Transferts de fichiers.
- Les packages avec yum, apt.
- Les utilisateurs et les groupes.
- Les services.

Travaux pratiques

- Utilisation des différentes commandes Ad-hoc en parallèle sur plusieurs noeuds.

Les playbooks

- Introduction aux playbooks.
- Définition des tasks, plays.
- Syntaxe Yaml.
- Variables, modules et tâches.
- Exécution d'un playbook.

- Test d'un playbook en dry-run.
- Exécution step by step, saut de tâches.
- Gestion des erreurs.

Travaux pratiques

- Écriture d'un playbook simple composé de plusieurs tâches.

Ecrire du code modulaire

- Notifications et Handlers.
- Les rôles et les includes. Les tags.
- Les modules de la communauté.
- Ansible-galaxy : partager son code.

Travaux pratiques

- Exploration de la galaxie Ansible, téléchargement et utilisation de modules, ajout de tags dans un playbook.

Les playbooks

- Les variables. Les templates et les filtres.
- Structures de contrôle : Conditions, Boucles et Blocks.
- Les prompts. Les facts.
- La rédaction de playbooks.

Travaux pratiques

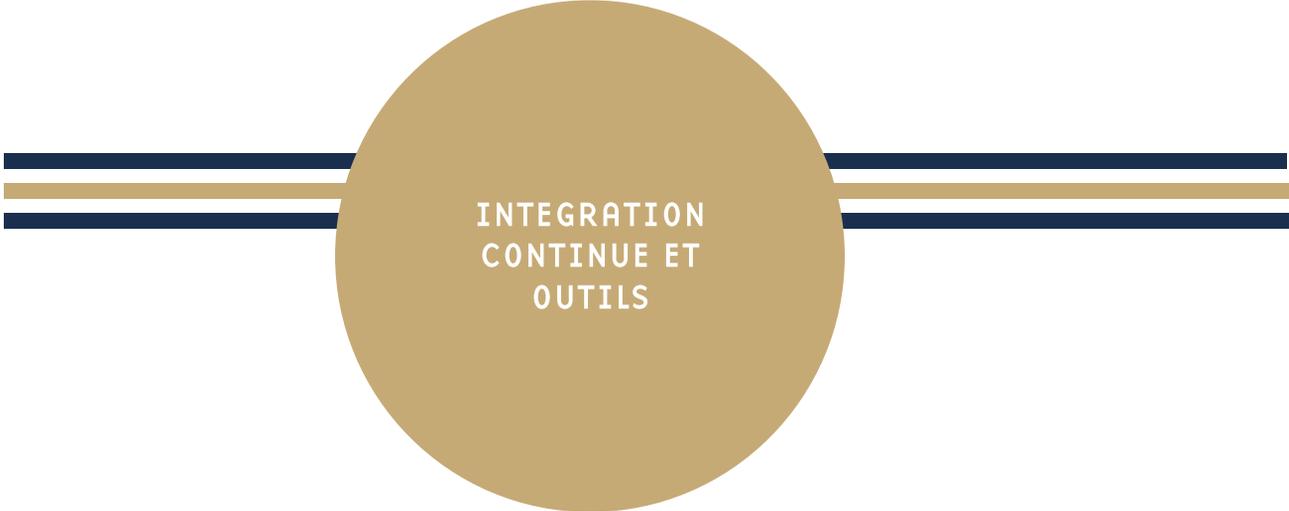
- Écrire un playbook complet pour le déploiement d'un service Apache sur plusieurs noeuds.

Commandes avancées

- Vault : chiffrement de données.
- Les lookups.
- Développer ses propres modules, déboguer un module, les plugins.
- Créer ses propres filtres.
- Ansible et Ansible Tower.

OBJECTIFS

- Rédiger des playbooks Ansible pour orchestrer des opérations au sein de leur parc



INTEGRATION
CONTINUE ET
OUTILS

GESTION DES SOURCES AVEC GIT

PROGRAMME DU MODULE

Le contrôle de version

- Pourquoi versionner son code source ?
- Les concepts de base du contrôle de version

Le principe DVCS (Distributed version Control)

- Qu'apporte la décentralisation ?
- Principe de fonctionnement
- Branche, dépôt, merge, rebase et tous les concepts DVCS
- Le contrôle de version

Utilisation au jour le jour

- Créer/cloner un dépôt
- Consulter l'état de l'arbre de travail
- Visualiser les modifications
- Enregistrer les modifications
- Parcourir l'historique des révisions
- Retrouver l'auteur d'une modifications
- Les concepts de base du contrôle de version

Gestion de dépôt et branches

- Créer une branche
- Passer de branche en branche avec les merges ou le rebase
- Mettre à jour un dépôt
- Exporter son dépôt
- Les dépôts distants

1 jour,
7 heures



DISTANCIEL

OBJECTIFS

- Comprendre les principes DVCS
- Apprendre à gérer son code source avec Git
- Apprendre à collaborer avec les dépôts Git
- Savoir manipuler les outils annexes à Git

GITLAB-CI

PROGRAMME DU MODULE

Introduction à GIT

Introduction à GitLab CI/CD

Pipeline

GitLab CI/CD pour Docker

Revue des Apps

Auto Devops

GitLab CI pour GitLab Pages

Exemples

Intégrations

Configuration spécifique

1 jour,
7 heures



DISTANCIEL

OBJECTIFS

- S'initier à GITLAB-CI



DOCKER

PROGRAMME DU MODULE

De la virtualisation à Docker

- Les différents types de virtualisation.
- La conteneurisation : LXC, namespaces, control-groups.
- L'évolution de Dotcloud à Docker.
- Le positionnement de Docker.
- Docker vs virtualisation.

Présentation de Docker

- L'architecture de Docker.
- Disponibilité et installation de Docker sur différentes plateformes (Windows, Mac et Linux).
- Création d'une machine virtuelle pour maquetage.
- La ligne de commande et l'environnement.
- Travaux pratiques
Créer une machine virtuelle pour réaliser un maquetage.

Mise en œuvre en ligne de commande

- Mise en place d'un premier conteneur.
- Le Docker hub : ressources centralisées.
- Mise en commun de stockage interconteneur.
- Mise en commun de port TCP interconteneur.
- Publication de ports réseau.
- Le mode interactif.
- Travaux pratiques
Configurer un conteneur en ligne de commande.

Création de conteneur personnalisé

- Produire l'image de l'état d'un conteneur.
- Qu'est-ce qu'un fichier DockerFile ?
- Automatiser la création d'une image.
- Mise en œuvre d'un conteneur.
- Conteneur hébergeant plusieurs services : supervisor.

- Travaux pratiques
Créer un conteneur personnalisé.

Mettre en œuvre une application multiconteneur

- Utilisation Docker Compose.
- Création d'un fichier yml de configuration.
- Déployer plusieurs conteneurs simultanément.
- Lier tous les conteneurs de l'application.
- Travaux pratiques
Mettre en œuvre une application multiconteneur.

Interfaces d'administration

- L'API Docker et les Webservices.
- Interface d'administration en mode Web.
- Docker Registry : construire et utiliser son propre hub.
- Exercice
Construire et utiliser son propre hub.

Administrer des conteneurs en production

- Automatiser le démarrage des conteneurs au boot.
- Gérer les ressources affectées aux conteneurs.
- Gestion des logs des conteneurs.
- Sauvegardes : quels outils et quelle stratégie ?
- Travaux pratiques
Administrer les conteneurs.

Orchestration et clusterisation

- Présentation de Docker Machine.
- Présentation de l'orchestrateur Swarm.
- Déploiement d'applications.

OBJECTIFS

- Comprendre le positionnement de Docker et des conteneurs
- Manipuler l'interface en ligne de commande de Docker pour créer des conteneurs
- Mettre en œuvre et déployer des applications dans des conteneurs
- Administrer des conteneurs

JENKINS

2 jours,
14 heures



DISTANCIEL

PROGRAMME DU MODULE

Intégration continue

- Définition, principes
- Notions de génie logiciel
- Best practices d'intégration continue
- La chaîne de fabrication logicielle

Utilisation de Jenkins

- Concepts, définitions
- Présentation de Jenkins comme serveur de build
- Archétype de projet
- Déclencheurs de build
- Résultat du build
- Workspace
- Visite guidée de l'IHM
- Jenkins dans l'IDE

Utilisation de Jenkins en cluster

- Configuration des « esclaves »
- Modes de démarrage Unix, Windows
- Répartition des jobs entre "esclaves".
- Bonnes pratiques de déploiement

Administration de Jenkins

- Configuration des vues Jenkins
- Considérations multiplateformes
- Visite guidée de la JENKINS_HOME
- Monitorer et sauvegarder Jenkins
- Scripts Jenkins en langage Groovy
- Utiliser la ligne de commande d'administration

Cas pratiques

- Création de builds et déploiements

OBJECTIFS

- Maîtriser les principes d'intégration continue
- Comprendre les processus et méthodes
- Comprendre la notion de build
- Automatiser la production logicielle
- Déployer Jenkins sur les projets



AGILE TDD

PROGRAMME DU MODULE

Les méthodes Agiles

- Présentation des méthodologies Agiles
- Approche en cascade vs approche itérative
- Le manifeste Agile
- La définition de « Terminer »
- Ce qu'il faut mettre en place pour itérer sur un projet
- La méthode SCRUM

Initiation au TDD

- Présentation du TDD
- Tests traditionnels et TDD
- Impact sur la documentation
- Impact des tests sur la conception
- Le développement orienté comportement
- Coûts et bénéfices

Mise en place du TDD dans une équipe

- Signaux externes de qualité d'un produit
- Résistances fréquemment rencontrées
- Mise en place d'un processus vertueux
- Rétrospectives
- Mythes et incompréhension

Présentation des outils de test

- Les outils xUnit
- Les Matchers Hamcrest
- Mock/fake/stub/dummy

- Différence entre simulacre et bouchons
- Présentation de l'outil easy-mock
- Bonnes pratiques

Re-factorisation

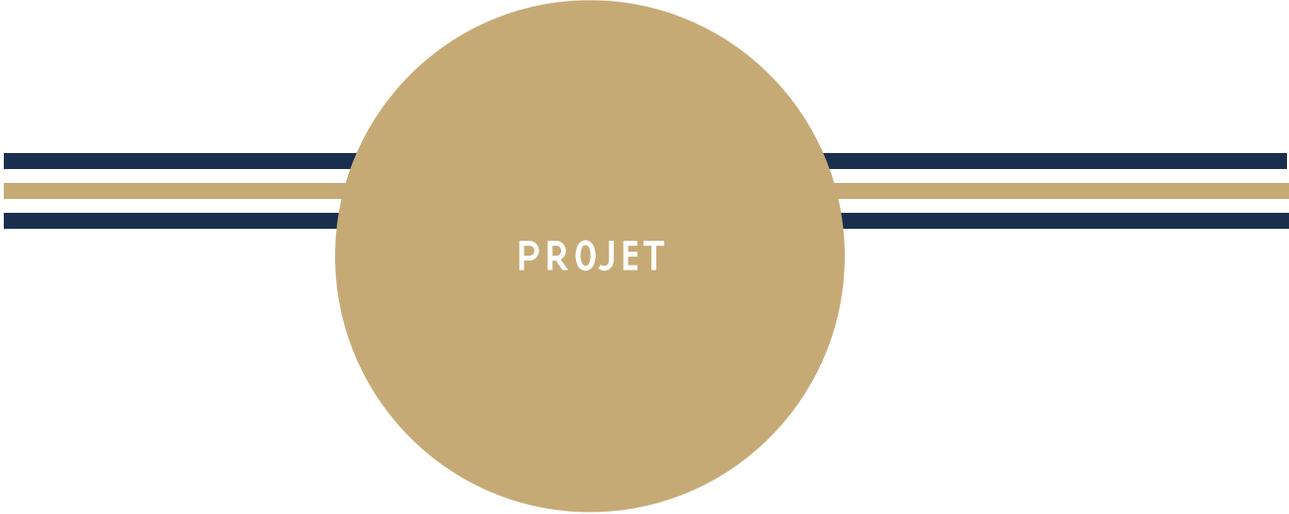
- Principes
- Niveaux de re-factorisation
- Quand doit-on re-factoriser ?
- Évaluer la qualité du code par des métriques
- La couverture de tests
- Les règles d'un design simple

Aller plus loin

- TDD et interfaces utilisateurs
- Spécificité de TDD et le développement d'une base de données
- L'intégration continue
- Exemple d'usine logicielle
- Les tests de charges

OBJECTIFS

- Apprendre à orienter son développement en fonction des futurs tests unitaires, partie indispensable à la mise en production d'une application



PROJET



PROJET FINAL

PROGRAMME DU MODULE

Déroulement du module

- Les stagiaires travaillent en toute autonomie, en binôme. Ils sont libres d'effectuer les choix adaptés, de développer les parties dont ils jugent avoir le plus besoin et d'apporter leurs propres solutions aux problèmes posés.
- Le formateur encadre les stagiaires par sa présence et répond aux questions. Il intervient pour épauler un binôme en difficulté ou pour faire le point à l'ensemble du groupe sur des notions non acquises. Il peut être amené à approfondir ou compléter certaines connaissances.

OBJECTIFS

- Mettre en application les acquis de la formation en complétant les mini projets réalisés dans tout le cursus

NOUS CONTACTER

AJC FORMATION
01 81 51 64 85
formonsnous@ajc-formation.fr
6 rue ROUGEMONT
75009 PARIS



www.ajc-formation.fr
www.ajc-classroom.fr

