

## DEVOPS

Filière





PROGRAMME  
DE LA FILIERE

## OBJECTIFS

- Gérer une plateforme d'intégration continue avec les outils Devops
- Savoir utiliser des conteneurs
- Avoir des notions sur le Cloud et l'automatisation de tâches
- Assimiler l'organisation de Ansible (rôles, tâches, playbooks, modules...)
- Mises en pratique

## METHODES ET MOYENS PEDAGOGIQUES

**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

## PRE-REQUIS

- Des notions systèmes seraient un plus

## PARTICIPANTS

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

## LIEU

Distanciel

## CERTIFICATION / ATTESTATION

Attestation de formation

# Programme - Contenu pédagogique

FONDAMENTAUX ET LANGAGE	UNIX SHELL	3 jours
OUTILS	GESTION DES SOURCES AVEC GIT	2 jours
	DOCKER	3 jours
FONDAMENTAUX ET LANGAGE	DEVOPS	2 jours
CLOUD	L'INGENIERIE DEVOPS SUR AMAZON WEB SERVICES	3 jours
OUTILS	JENKINS	1 jour
	ANSIBLE	4 jours
PROJET	PROJET FINAL	2 jours



---

---

PROGRAMMES  
DÉTAILLÉS



FONDAMENTAUX  
ET LANGAGE



# UNIX SHELL

## PROGRAMME DU MODULE

### UNIX

#### Introduction

- Historique
- Caractéristiques
- Système de fichiers
- Débuter avec le Shell

#### Une Session

- Connexion
- Interface graphique
- L'aide avec Man

#### Les commandes indispensables de l'éditeur Vi

#### Les fichiers et répertoires

- L'arborescence
- Les commandes de gestions de fichiers
- Les commandes de gestions de répertoires
- La commande find

#### Les liens

- Concept
- Création de liens
- Les liens symboliques

#### Les droits

- Les utilisateurs et groupes
- Affichage et modification des droites
- Droits par défaut
- Gestion des groupes
- Les droits avancés

#### Gestion des processus

- Gestion des jobs
- Les commandes de gestion des processu

#### Gestion des packages avec apt et yum

#### Gestion des services

#### Démarrage du système

#### SHELL

- Généralités
- Les jokers
- Protection des caractères spéciaux
- Les redirections
- Les tubes
- Historique des commandes
- Expressions régulières (grep...)
- Les alias
- Les variables d'environnement
- Les filtres
- Les scripts

## OBJECTIFS

- Acquérir la connaissance des commandes fondamentales des systèmes d'exploitation Unix et Linux à travers des exercices modulaires de difficulté progressive
- Devenir autonome pour une première prise en main d'un système
- Apprendre à mettre en place des traitements et procédures BATCH
- Avoir quelques notions d'administration



# DEVOPS

## 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

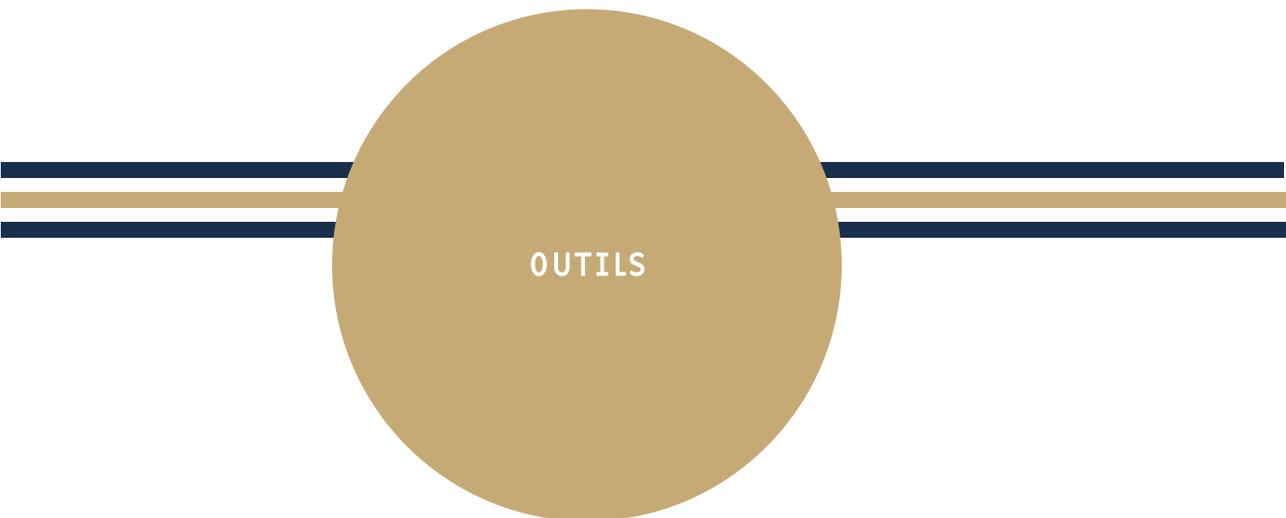
- Performance
- Sécurité
- Exploitabilité

### 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

A decorative graphic consisting of a central brown circle. Three horizontal lines pass through the circle: a dark blue line on top, a brown line in the middle, and another dark blue line on the bottom. The word "OUTILS" is written in white, uppercase letters in the center of the circle.

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

## 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



# 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

## 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



# ANSIBLE

## 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

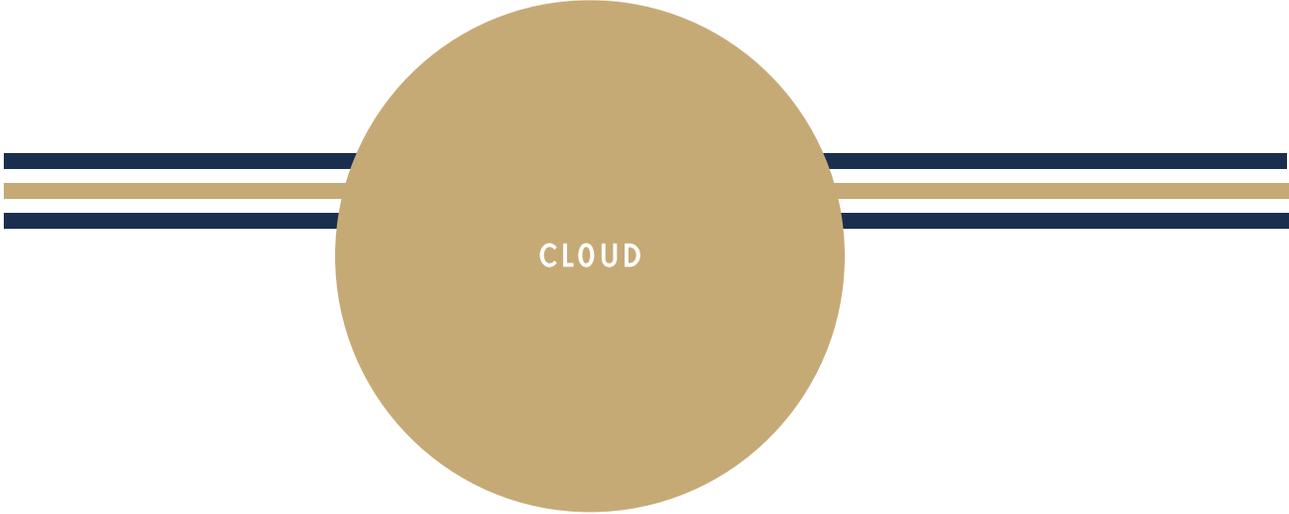
- Écrire un playbook complet pour le déploiement d'un service Apache sur plusieurs noeuds.

### 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



CLOUD



# 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 ?

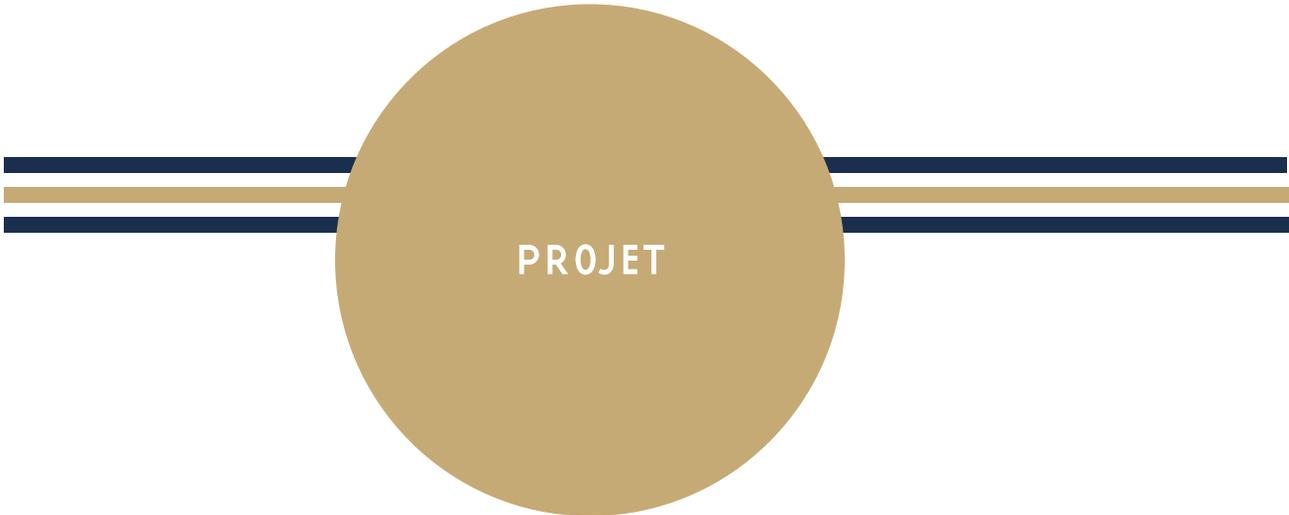
- Composants Amazon AWS avancés.
- Les services Virtual Private Cloud, Beanstalk et leur modèle de programmation.
- Exercice  
*Création de ressources virtuelles avec les services EC2, S3, SQS, RDS. Déploiement d'une application Java.*

### 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



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

6 rue ROUGEMONT  
75009 PARIS



[www.ajc-formation.fr](http://www.ajc-formation.fr)  
[www.unjourunjob.com](http://www.unjourunjob.com)

