

**Présentation &
Programme de la filière**

Préparation Opérationnelle à l'Emploi Individuelle
**Ingénieur de
Production**

Du 06/03/2024 au 31/05/2024

Filière

57 jours

399 heures

Version 2024 1.1

sopra  steria

PRÉSENTATION DE LA FILIÈRE

8458.80 € TTC / participant

57 jours / 399 heures

La durée et le prix correspondent à une inscription en inter-entreprises. Toute demande inter-entreprise fait systématiquement l'objet d'un devis si-mesure devant être approuvé pour acceptation.

Objectifs

- Administrer et exploiter une infrastructure multiplateforme
- Mettre en œuvre et déployer des applications virtualisées
- Mettre en œuvre une administration centralisée et automatisée au travers d'outils spécialisés
- Participation aux projets d'évolutions des infrastructures

Profils recherchés

Demandeurs d'emploi issus d'une filière scientifique

Prérequis

Aucun prérequis n'est obligatoire pour accéder à cette formation

Modalités et délais d'accès

- Les postulants devront passer une série d'entretiens pour intégrer la formation
- Ils seront informés de leur inscription au plus tard 15 jours avant le début de la session

Accessibilités aux personnes en situation de handicap

Les personnes en situation de handicap sont invitées à nous communiquer leurs besoins spécifiques. Nous ferons tout pour les mettre dans les meilleures conditions de suivi de la formation possibles (compensation, accessibilité...) ou pour les réorienter.

Vous pouvez contacter Manuela Janvier au 07 64 01 16 07 ou m.janvier@ajc-ingenierie.fr notre référente handicap

Horaire de formation

9h30 – 13h puis 14h – 17h30

Modalités d'évaluation

- A l'issue de chaque module, le formateur évaluera chacun des participants en fonction des cas pratiques, exercices effectués et/ou QCM.
- La fin de la formation sera consacrée à un projet final reprenant l'ensemble des acquis de la formation. Les apprenants participeront à une soutenance pour présenter leur projet devant un jury et démontrer leurs nouvelles compétences

Attestation/certification

Une attestation de fin de stage sera remise à tous les participants à l'issue de leur parcours

PRÉSENTATION DE LA FILIÈRE

8458.80 € TTC / participant

57 jours / 399 heures

La durée et le prix correspondent à une inscription en inter-entreprises. Toute demande inter-entreprise fait systématiquement l'objet d'un devis si mesure devant être approuvée pour acceptation.

Méthodes mobilisées

Alternance d'exercices, cas pratiques, QCM et de notions théoriques, Projet Fil Rouge.

Les supports de cours seront remis via notre plate-forme de téléchargement Quest et/ou AJC Classroom. AJC met à la disposition de chacun un accès aux logiciels utiles dans le cadre de leur module.

Informations concernant les classes virtuelles

Avec @JC CLASSROOM, les stagiaires profitent des mêmes interactions avec leur formateur et l'équipe pédagogique qu'en présentiel : échanges en visioconférence et par chats. La formation se déroule en connexion continue 7h/7. Le formateur peut vérifier l'avancement du travail et évaluer des stagiaires à l'aide d'exercices et de cas pratiques. Cela lui permet d'apporter un suivi pédagogique et des conseils personnalisés.

Notre équipe technique envoie aux futurs stagiaires les modalités de connexion (accès, identifiants, dates, heures et numéro de la hotline) par mail dès leur inscription. Si les stagiaires rencontrent un problème de connexion, ils peuvent 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-ingenierie.fr)

CONTENU PÉDAGOGIQUE

COMPORTEMENTAL	SAVOIR-ETRE EN ENTREPRISE	2 jours
	GESTION DU TEMPS ET DES PRIORITES	1 jour
FONDAMENTAUX ET LINUX	FONDAMENTAUX RESEAUX	3 jours
	COMMANDES LINUX	2 jours
	ALGORITHMIQUE	2 jours
	SHELL SCRIPT	3 jours
	ADMINISTRATION LINUX NIVEAU 1	5 jours
	ADMINISTRATION LINUX NIVEAU 2	4 jours
PROJET	PROJET 1 ^{ère} PARTIE	1 jour
BDD et SERVEURS APPLICATIONS	BASE DE DONNEES	4 jours
	MONGODB	2 jours
	SERVEURS APPLICATIFS (Apache, Tomcat, Wildfly, ...)	4 jours
	ZABBIX	3 jours
	VMWARE VSPHERE	4 jours
PROJET	PROJET 2 ^{ème} PARTIE	1 jour
FONDAMENTAUX ET LINUX	INTRODUCTION A LA CYBERSECURITE	2 jours
DEVOPS	DEVOPS ET INTEGRATION CONTINUE	3 jours
	DOCKER	3 jours
	ANSIBLE	4 jours
PROJET	PROJET FINAL	4 jours

57 jours



Programmes détaillés

COMPORTEMENTAL

SAVOIR-ETRE EN ENTREPRISE

2 jours / 14 heures

Objectifs

- Adopter la bonne attitude au quotidien, trouver son style pour donner une bonne image de soi
- Comprendre l'impact du comportement verbal et non verbal sur ses relations professionnelles
- Appréhender les codes du savoir-être en entreprise

Contenu du programme

Pourquoi développer son savoir être professionnel ?

- Pourquoi présenter une bonne image chez un employeur ?
- Quelles sont les conséquences pour soi et pour son quotidien ?
- Retour sur son rapport au savoir-être et au savoir-vivre avec les autres

Adopter le bon comportement professionnel en entreprise

- Comprendre l'impact qu'exerce son image sur soi et sur les autres
- Définir si son image contribue ou non à se mettre en valeur
- Apprendre à mettre en avant ses talents, savoir communiquer dans un milieu professionnel
- Quel vocabulaire adopter ?
- Prendre conscience de l'impact de son comportement non verbal dans la communication
- Accepter ses capacités et ses limites, ses défauts

Intégrer les règles incontournables de la vie en entreprise

- Comprendre le fonctionnement, les règles de l'entreprise et usages de chacun
- Mise au point sur la notion de harcèlement et de respect mutuel
- Utiliser les règles de politesse et les expressions appropriées à chaque circonstance
- Savoir gérer les situations embarrassantes, éviter les impairs, les rattraper s'ils se sont produits
- Présenter et recevoir des excuses
- Tutoyer ou vouvoyer ? Adopter la bonne distance

S'intégrer dans une équipe

- Intégrer et comprendre la culture de son entreprise : ses valeurs et ses codes
- Comprendre vite pour s'adapter rapidement : les personnes, les objectifs, les contraintes
- Développer son écoute active et optimiser sa communication

Intégrer les technologies dans l'appréciation du savoir-vivre

- Adopter les bons usages de la messagerie au travail, les codes et les limites
- Mise au point sur l'utilisation du téléphone et autres outils personnels dans un cadre professionnel
- Focus sur les bonnes pratiques métier : à déterminer en amont de la formation

GESTION DU TEMPS ET DES PRIORITES

1 jour / 7 heures

Objectifs

- Acquérir des outils et des méthodes de gestion du temps afin de mettre en place des comportements nouveaux
- Prendre conscience de son comportement
- Reprendre le contrôle de son temps

Contenu du programme

Le temps : un allié de la croissance professionnelle

Connaître les différentes manières de structurer son temps

- Types de personnalités et structuration du temps
- Bilan de ses pratiques actuelles et de l'influence de son environnement
- Prise de conscience individuelle, premier diagnostic et niveaux de motivation de chacun

Savoir faire des choix

- Clarifier sa mission et les tâches qui en découlent
- Fixer et fractionner des objectifs
- Hiérarchiser ses priorités
- Savoir filtrer, sélectionner les véritables urgences

Maîtriser son temps sans subir

- Déterminer et agir sur les "voleurs de temps"
- Mieux renoncer pour mieux choisir

Gérer son temps avec les autres

Savoir dire "non"

- Gérer les interruptions
- Savoir déléguer

Utiliser ses forces positives

- Mieux connaître son capital énergie, ses rythmes de travail
- Contacter ses ressources positives, s'en servir comme multiplicateur d'énergie
- Savoir se concentrer, se motiver, s'arrêter, se relaxer

Intégrer le stress

- Rôle du stress, personnalités sensibles
- Se servir du "bon" stress, se protéger du "mauvais" stress
- Gestion des situations de stress les plus fréquentes ou cas particuliers

Qu'acceptez-vous de changer ?

- Déterminer les points réalistes de son contrat de changement
- Visualiser les résultats, modéliser ceux qui savent gérer leur temps



Programmes détaillés

FONDAMENTAUX ET

LINUX

FONDAMENTAUX RESEAUX

3 jours / 21 heures

Objectifs

- Connaître les différents équipements réseaux
- Comprendre l'importance du protocole TCP/IP dans l'élaboration d'un réseau
- Installer un réseau physique : poste de travail, routeur, commutateur, dns, dhcp
- Appréhender les principaux services et protocoles tcp/ip, udp, arp, http, https

Contenu du programme

Introduction

- Maintenance de la qualité de service
- Classification des réseaux: LAN, WAN
- Paquet
- Circuit
- Câblé
- Sans fil
- Standardisation des communications de données

Développement de réseaux avec les liaisons de données

Information d'encodage

- Bits, octets et paquets
- Avantages de l'encodage numérique

Amélioration de l'efficacité avec le contrôle d'erreurs

- Acheminement de paquets dans les trames
- Détection et correction des erreurs
- Utilisation d'ACK et correction d'erreur par retransmission

Déploiement de médias physiques

Identification des types de médias

- Sélection des types de câbles de cuivre (Cat. 5e ou plus)
- Avantages par rapport à la fibre optique

Utilisation de liaisons sans fil

- Utilisation des bandes et fréquences radio
- Gestion des interférences et bruits

Miser sur Ethernet

FONDAMENTAUX RESEAUX (suite)

3 jours / 21 heures

Contenu du programme

Étude des standards IEEE 802

- Transfert avec des adresses MAC
- 1 Mo/s à 100 Go/s
- Comparaison entre LAN commuté et partagé

Étude détaillée d'Ethernet

- Étude de la commutation Ethernet
- Ajout de QoS à Ethernet
- Commutation de couche 2 et de couche 3

Exploiter le Wi-Fi pour permettre la mobilité

Communication via les ondes radio

- Types de réseaux Wi-Fi: a, b, g et n
- Miser sur le mode infrastructure et la mobilité

Intégration du Wi-Fi

- Vérification de la transmission
- Augmenter le débit et la portée avec 802.11n
- Fournir une QoS pour la voix et le multimédia

Déploiement des points d'accès

- Transfert du trafic via les points d'accès
- Points d'accès bi-bande
- Utilisation de SSID (Service Set Identifiers)

Créer des sous-réseaux avec TCP/IP et des routeurs

TCP/IP: Une suite de protocoles

- Utiliser TCP pour les données, UDP pour la voix et la vidéo
- Maximisation des applications et équipements TCP/IP
- Optimisation du trafic VoIP et de données

Diagrammes de données et adressage IP

- Augmentation de l'efficacité avec des schémas d'adressage
- Interprétation des masques sous-réseaux et préfixes réseaux

Mode opératoire des routeurs

- Relais du trafic avec les tables de routage
- Découverte de chemins avec les protocoles de routage
- Migration des routeurs pour la QoS

FONDAMENTAUX RESEAUX (suite)

3 jours / 21 heures

Contenu du programme

Mise en œuvre de la sécurité

- VPN (réseau privé virtuel)
- Authentification des utilisateurs
- Tunnels chiffrés VPN
- Vérifier l'intégrité et la source des informations

Évaluation des risques et contre-mesures

- Analyse des menaces et besoins en sécurité
- Chiffrement des données
- Tunnels L2 et L3
- Utilisation de certificats et signatures numériques

Sécurité LAN

- Sécurité Wi-Fi: WPA, WPA2, 802.11i, AES
- Isolation des groupes de travail avec les VLAN

Création de réseaux d'entreprise

Utilisation des liaisons Télécoms

- Flux de données en circuits commutés
- Lignes louées E1 et T1

Communication intersites

- Choix des options xDSL
- LAN Extension Services (LES) et Metro-Ethernet

Sélection de services réseau évolutifs

- MPLS (Multi Protocol Label Switching)
- Frame Relay
- Services ISP améliorés et Informatique et services en nuage

COMMANDES LINUX

2 jours / 14 heures

Objectifs

- Les commandes de base, l'utilisation poussée du bash, les grand utilitaires Unix dont les expressions régulières, sed et awk

Contenu du programme

Introduction

- Présentation des shells
- Premières commandes
- Les différents Shell
- Comparaison de sh, bash, ksh et dash

Aide

- L'aide locale
- À savoir
- Freenode
- Usage IRC

Commandes de base

- Accéder au contenu des fichiers
- Commandes de compression, d'impression et de gestion du temps
- Gestion administration
- Commandes composites pipes et redirections

Variables

- Présentation
- Variables utilisateur
- Prompt
- Substitutions

Commandes internes

- Set
- cd, pushd, popd, umask, type, enable
- Historique et Alias
- Commande sur les processus : kill, jobs, wait, ulimit

COMMANDES LINUX (suite)

2 jours / 14 heures

Contenu du programme

Utilitaires

- Commandes cut, tr, uniq, sort, wc, find, grep
- Commandes de transformation : iconv, od, nl, basename, diff
- Commandes utilitaires : xargs, tee, cmp, comm, paste, sed
- Expressions régulières

Awk

- Principe awk
- Critères de sélection
- Instructions de contrôle
- Les Fonctions

Objectifs

- Structurer des programmes selon un algorithme
- Maîtriser les éléments de lexique et de syntaxe d'un langage pour écrire un programme
- Compiler et exécuter un programme

Contenu du programme

Les fondements de la programmation

- Qu'est-ce qu'un programme ? Qu'est-ce qu'un langage ? Les différents paradigmes. Quel langage pour quelle application ?
- Les compilateurs. Les exécutables.
- Les responsabilités d'un programmeur.
- Qu'est-ce qu'un algorithme ?
- Les besoins auxquels répond un algorithme.
- Le concept de pseudo-langage

Genèse d'un premier programme

- Ecriture d'un programme simple : syntaxe et instructions.
- Compilation et exécution du programme.
- Qu'est-ce qu'une librairie ? Son rôle, son usage

Règles de programmation

- Convention de nommage.
- Convention syntaxique.
- Utilisation des commentaires. Pourquoi commenter les développements ?
- Améliorer la lisibilité des programmes : indentation du code, découpage du code...

Les variables

- Qu'est-ce qu'une variable ?
- Pourquoi typer une variable ?
- Les types primitifs : entiers, chaînes de caractères, nombres réels, autres.
- Déclaration, définition et initialisation d'une variable.
- Les constantes.
- Saisie, affichage, affectation, conversion de type.
- Organiser ses données sous forme de tableaux.
- Les types évolués : enregistrement, matrice, arbre

ALGORITHMIE (suite)

2 jours / 14 heures

Contenu du programme

Opérateurs et expressions

- Qu'est-ce qu'un programme ? Qu'est-ce qu'un langage ? Les différents paradigmes. Quel langage pour quelle application ?
- Les compilateurs. Les exécutables.
- Les responsabilités d'un programmeur.
- Qu'est-ce qu'un algorithme ?
- Les besoins auxquels répond un algorithme.
- Le concept de pseudo-langage

Les structures de contrôle

- Ecriture d'un programme simple : syntaxe et instructions.
- Compilation et exécution du programme.
- Qu'est-ce qu'une librairie ? Son rôle, son usage

Les procédures et les fonctions

- Convention de nommage.
- Convention syntaxique.
- Utilisation des commentaires. Pourquoi commenter les développements ?
- Améliorer la lisibilité des programmes : indentation du code, découpage du code

Introduction à la programmation objet

- Les concepts associés à la programmation objet : classe, attribut, méthode, argument.
- Introduction aux bonnes pratiques d'organisation de conception et d'organisation d'un programme

Maintenance, débogage et test des programmes

- Savoir lire et interpréter les différents messages d'erreurs.
- Utiliser un débogueur : exécuter un programme pas à pas, points d'arrêts, inspecter les variables pendant l'exécution

SHELL SCRIPT

3 jours / 21 heures

Objectifs

- Acquérir les compétences pour écrire des scripts en Shell et exploiter les possibilités des filtres Unix/Linux.

Contenu du programme

INTRODUCTION

- Rôle d'un Shell
- Présentation des différents shell sous Unix/Linux
- Types et syntaxes

AIDE

- Les man
- Help
- IRC freenode

PARAMETRAGE DE L'ENVIRONNEMENT

- Options du Shell
- Variables et fichiers d'environnement
- Historique des commandes

UTILISATION DU SHELL EN MODE INTERACTIF

- Énumérer les commandes essentielles par thème
- Substitution de nom de fichiers
- Protection des caractères spéciaux
- Redirections et Tubes de communication
- Regroupement des commandes

BASE DE LA PROGRAMMATION

- Structure d'un script
- Commentaires
- Exécution d'un script
- Débogage d'un script et Code de retour

VARIABLES ET CONSTANTES

- Var
- iables et Constantes
- Tableaux
- E/S de données
- Commandes de substitution
- Pushd et popd

SHELL SCRIPT (suite)

3 jours / 21 heures

Contenu du programme

STRUCTURE DE CONTROLE

- Instructions conditionnelles
- Choix multiples
- Boucles et Sauts inconditionnels

ALIAS ET FONCTIONS

- Alias
- Sous-programme sous forme de script
- Sous-programme sous forme de fonction

ARITHMETIQUES

- Syntaxe
- Commande expr

EXPRESSIONS REGULIERES

- Meta-caractères des expressions régulières
- Utilisation des expressions régulières avecGrep

CHAINE DE CARACTERES

- Manipulation de chaînes de caractères
- Expressions de variables
- Commandes basename et dirname

FILTRE SED

- Principe de fonctionnement
- Commandes de sed
- Utilisation des expressions régulières
- Présentation des sous-expressions

PROCESSEUR DE TEXTE AWK

- Principes de fonctionnement
- Structure d'un programme awk
- Critères
- Variables et les expressions
- Tableaux
- Instructions et Fonctions prédéfinie

ADMINISTRATION LINUX NIVEAU 1

5 jours / 35 heures

Objectifs

- Devenir autonome sur Linux afin de garantir la bonne disponibilité des serveurs
- Pouvoir prendre en charge la responsabilité de l'administration de systèmes Linux
- Savoir intégrer Linux avec les autres systèmes d'exploitation de l'entreprise
- Être en mesure de garantir un premier niveau de sécurité d'une infrastructure Linux

Contenu du programme

Présentation

- L'historique d'Unix et Linux
- Les caractéristiques de Linux, les Unix-Like, les distributions Linux
- Comment administrer le système : le mode texte et les outils d'administration
- La documentation : le man, les autres sources d'informations (Howto, ...)

Installer Linux et ses applications

- Introduction : Linux, les distributions Linux, les sources d'information
- Installer un système de type RedHat et un système de type Debian
- Administrer le système avec sudo sous Debian et RedHat
- Installer des applications sous RedHat : les paquets RPM, le système YUM
- Installer des applications sous Debian : les paquets DEB, le système APT

Administrer le système avec les commandes du mode texte

- Utiliser le Shell, connaître les commandes de base du système (rappels)
- Savoir lire des scripts Shell
- Gérer les utilisateurs : les commandes de gestion des comptes, les droits (rappels)
- Gérer les processus (rappels), gérer les bibliothèques partagées
- Savoir programmer des travaux périodiques
- Savoir organiser les journaux de bords et leur rotation

Gérer l'espace disque

- Comprendre la vision Linux des disques, partitionner des disques (Msdos, GPT)
- Gérer le LVM, gérer le swap
- Gérer les FS (ext2/ext3/ext4, xfs, ...) et les quotas

Gérer l'arrêt et le redémarrage

- Connaître les grandes étapes du démarrage (BIOS, bootloader, kernel, initramfs, init)
- Utiliser le chargeur ("bootloader") Grub
- Gérer le démarrage des services : init SysV, Upstart, systemd ; la notion de runlevel

ADMINISTRATION LINUX NIVEAU 1 (suite)

5 jours / 35 heures

Contenu du programme

Configurer TCP/IP en environnement Linux

- Ajouter un système (Debian, RedHat) dans un réseau IPv4/IPv6
- Connaître les commandes de diagnostics
- Comprendre le fonctionnement des systèmes INETD (inetd, xinetd), les wrappers

Gérer l'environnement graphique

- Connaître le vocabulaire : Serveur X, X11, DISPLAY, DM, Desktop Environment, ...
- Choisir son Desktop (XFCE, Gnome, KDE), configurer un DM (GDM, LightDM, ...)

Les fondamentaux de la sécurité

- Savoir configurer la sécurité de connexion, limiter l'usage des ressources
- Savoir utiliser les commandes SSH d'accès distant sécurisées

Éléments divers

- Savoir gérer l'impression sous Linux (CUPS)
- Connaître le matériel présent, comprendre leur gestion par Linux
- Connaître les rudiments du SQL pour interroger une base de données
- Savoir configurer les locales, le fuseau horaire et l'heure
- Savoir configurer la messagerie d'un compte Linux (retransmission, alias, ...)

ADMINISTRATION LINUX NIVEAU 2

4 jours / 28 heures

Objectifs

- Acquérir un niveau d'expertise plus élevé sur Linux
- Savoir tirer parti simplement de la richesse modulaire de Linux et du monde Open Source
- Comprendre comment organiser et gérer l'espace disque de gros serveurs Linux
- Apprendre à paramétrer finement le système
- Savoir déployer Linux et l'intégrer avec les autres environnements existants

Contenu du programme

Les systèmes de fichiers journalisés, les systèmes de fichiers spéciaux

- Comparaison des systèmes de fichier journalisés
- Gérer des FS ext/xfs : créer, supprimer, configurer, vérifier, réparer, surveiller, ...
- Savoir configurer le montage automatique avec l'autofs

Paramétrage avancé des systèmes de fichiers et des disques

- Panorama des techniques RAID, savoir gérer le RAID logiciel MD
- LVM: les concepts (rappels), les pratiques avancées : Snapshots, Stripping, RAID, ...
- SAN ou NAS ?
- Le vocabulaire du SAN, savoir gérer un SAN iSCSI
- Gérer le swap

Modifier le noyau

- Pourquoi recompiler le noyau ?
- Patcher et compiler le noyau
- Ajouter un module et modifier les paramètres du noyau sans recompiler

La gestion des périphériques

- Comment sont gérés les périphériques sous Linux, les commandes d'informations ?
- L'ajout d'un périphérique, le système UDEV

Le démarrage d'un système Linux

- Le chargeur Grub : installation et paramétrage; les autres chargeurs (Lilo, Syslinux, ...)
- Paramétrer de manière souple le démarrage avec les fichiers initrd/initramfs
- Utiliser un système bootable de dépannage

Garantir l'intégrité des données : les sauvegardes

- Panorama des outils de sauvegarde, les sauvegardes incrémentales
- Le couteau suisse de la sauvegarde : la commande tar ; les autres commandes

ADMINISTRATION LINUX NIVEAU 2 (suite)

4 jours / 28 heures

Contenu du programme

Améliorer les performances du système

- Panorama des outils de diagnostic, les méthodes pour obtenir un système performant
- La surveillance des performances, les outils de l'expert : vmstat, sar, dstat, collectd, iostat, ...

Le dépannage

- Une approche méthodique du dépannage
- Les outils de l'expert : strace, lsof, tcpdump/wireshark, ...
- Se prémunir des problèmes

La maîtrise du réseau

- Organisation d'un réseau TCP/IP (rappels)
- Les configurations de base et avancées du réseau
- Savoir dépanner les problèmes réseaux

Points divers

- L'installation d'un logiciel à partir des sources
- La communication avec les utilisateurs (messagerie, wall / etc / issue / etc / motd, ...)



Programmes détaillés

PROJET

PROJET 1^{ère} PARTIE

1 jour / 7 heures

Objectifs

- Réaliser un projet en groupe reprenant les éléments techniques vues dans les fondamentaux et l'infrastructure Linux

Programmes détaillés

BASE DE DONNEES ET SERVEURS D'APPLICATIONS

BASE DE DONNEES

4 jours / 28 heures

Objectifs

- Comprendre les modèles conceptuels et physiques des bases relationnelles et non relationnelles
- Mettre en place des requêtes SQL avec jointures, regroupement et sous-requêtes
- Connaître le LMD, le LDD et les traitements stockés
- Connaître les fonctionnalités de base d'exploitation des différentes BDD du marché

Contenu du programme

Introduction aux Base de Données

- Historique
- Qu'est-ce qu'une base de données ?
- Différents types pour différents usages (hiérarchique, réseau, relationnelle, orientée objet, orientée cloud, NoSQL, graphe, etc ...)
- Panorama des outils du marché

Introduction à Merise

Les principes de base Merise

- Les cycles de la démarche Merise
- La technique de la démarche

Le MCD

- Entité
- Association

Les compléments du modèle

- Les formes normales
- Nouveautés Merise 2
- Le MLD, le MPD
- Une table exemple
- Formalisme
- Passage du MCD au MLD
- Le reverse-engineering
- Les outils de modélisation et de génération

BASE DE DONNEES (suite)

4 jours / 28 heures

Contenu du programme

Le langage SQL

- Introduction
- Les requêtes simples
 - . Connaître la syntaxe de l'ordre SELECT
 - . Connaître les prédicats simples
 - . Savoir écrire des requêtes sur une seule table
 - . Comprendre ce qu'est une jointure
 - . Savoir écrire des requêtes sur plusieurs tables
 - . Connaître les fonctions utilisées dans les requêtes
- Les requêtes avec regroupement
 - . Comprendre le regroupement
 - . Savoir écrire une requête utilisant le regroupement
- Les requêtes ensemblistes
 - . Connaître les opérateurs ensemblistes
 - . Savoir écrire une requête utilisant des opérateurs ensemblistes
- Les requêtes imbriquées
 - . Connaître les opérateurs utilisés pour les sous requêtes
 - . Savoir écrire des requêtes imbriquées
- Le langage de manipulation de données
 - . Savoir insérer une ligne dans une table
 - . Savoir modifier une ligne dans une table
 - . Savoir supprimer une ligne dans une table
- Le langage de définition de données
 - . Syntaxe de création des différents types de tables
 - . Définition de contraintes (clé primaire, clé étrangère, ...)
- Concepts procéduraux
 - . Procédures et fonction stockées
 - . Triggers

Focus sur quelques bases relationnelles : ORACLE et PostgreSQL

- Architecture d'une base de données et outils de gestion, architecture client/serveur, architecture web, etc
- Notions de sauvegarde, réplication, clustering, restauration
- Lecture de journaux de transaction
- Troubleshooting typique sur les bases de données
- Si le temps le permet, les bases MySQL et Microsoft SQL Server seront abordées

Objectifs

- Comprendre le fonctionnement de MongoDB
- Comprendre comment installer, configurer et administrer un serveur MongoDB
- Être capable de créer des requêtes d'interrogation
- Disposer des compétences nécessaires pour mettre en œuvre la réplication avec MongoDB

Contenu du programme

Introduction

- Présentation MongoDB, historique du projet, les versions
- Structure des données : notions de documents, de collections
- Le format BSON (Binary JSON), comparaison avec JSON
- Fonctionnalités de MongoDB
- Interfaces disponibles

Installation et configuration

- Plates-formes supportées
- Packages nécessaires, scripts de lancement
- Travaux pratiques : installation, lancement du service mongod
- Tests de connexion

Interpréteur

- Présentation du Shell Mongo
- Initialisation et premières requêtes
- Opérations CRUD : Create, Read, Update, Delete
- Importation, exportation de données
- Travaux pratiques : la méthode find, critères de requêtes, les types, les curseurs, ...

Sécurité

- Mise en œuvre de l'authentification dans MongoDB
- Paramètres de configuration auth et keyFile
- Gestion des rôles
- Étude de la collection system.users
- Les outils de modélisation et de génération

MONGODB (suite)

2 jours / 14 heures

Contenu du programme

Le sharding

- Définition, principe de fonctionnement
- Exemples de mise en oeuvre du sharding, configuration et administration
- Réplication : principe des replica sets et mise en oeuvre
- Mécanisme de fail-over automatique
- Partitionnement des données avec le sharding
- Optimisation : gestion des connexions, ajout de serveurs, équilibrage

Exploitation

- Gestion des opérations, analyse, points de blocage
- Mise en oeuvre de mongotop et mongostat
- Gestion des index, chargement des données en mémoire
- Analyse des logs

Administration

- Supervision : gestion de la mémoire, analyse des performances, tuning
- Sauvegardes d'un serveur, de cluster
- Travaux pratiques avec mongodump

SERVEURS APPLICATIFS (Apache, Tomcat, Wildfly, ...)

4 jours / 28 heures

Objectifs

- Installer et configurer un serveur Tomcat
- Déployer des applications JavaEE
- Mettre un place un pool de connexions vers une base de données
- Optimiser les performances d'un serveur Tomcat
- Sécuriser un serveur Tomcat et ses ressources
- Installer un serveur d'application JBoss
- Maîtriser les modes de déploiement des composants war , jar et ear
- Configurer un domaine avec plusieurs hosts
- Superviser l'activité du serveur d'application JBoss

Contenu du programme

Contexte des applications JavaEE

- Les projets Apache. Les distributions de Tomcat
- JavaEE, Servlet, JSP et composants : architecture

Installation

- Installation et configuration de la JVM.
- Installation du serveur. Les options.
- Les éléments de Tomcat : répertoires, moteur, connecteurs

Configuration

- Configurations : composants, utilisateurs, déploiement, contrôle d'accès, application Web.
- Outils de configuration. Automatisation avec Ant
- Filtrage de requêtes (les "Valves"), de logging . Single Sign On. Les Realms
- Configuration de la persistance de sessions. Les ressources JNDI : JDBC et Mail.□
- Analyse du chargement des classes Java.

Les connecteurs

- HTTP 1.0 et 1.1 et connecteur Coyote.□
- Utilisation de Tomcat avec un Reverse Proxy.
- Intégration avec IIS et Apache.
- Intégration via mod_proxy (HTTP et AJP)
- Intégration via mod_jk
- Optimisation. Répartition de la charge.
- Connecteurs JDBC : alternative et configuration.

SERVEURS APPLICATIFS (Apache, Tomcat, Wildfly, ...) (suite)

4 jours / 28 heures

Contenu du programme

Sécurité

- Principes. Sécurisation du système de fichiers. Les permissions. Java Security Manager et Tomcat
- Les "Realms " : message Digest, rôles, fichiers,
- Le protocole SSL

Journalisation

- JULI, configuration logging.properties
- Projet Log4J. Sorties : console, fichier, syslog
- Logger. Niveaux de log. Lecture du journal
- Configuration par XML ou Java.
- Remplacer JULI par Log4J.

Performances et tests

- Configuration "standalone " ou avec serveur
- JVM, usage de la mémoire.
- Mise à l'échelle.
- Tests de charge avec JMeter
- Charge du serveur et charge de l'application.
- Le monitoring JMX.

Introduction à JBOSS

- Principes de l'Open Source.
- L'Open Source professionnel avec RedHat
- Jboss EAP vs WildFly , serveurs certifiés Java EE
- L'architecture de WildFly

Base de l'administration JBOSS

- Principe des configurations.
- Déploiement d'applications Java EE (ear), d'applications Web (war) et de composants EJB (jar)
- Déploiement de Driver JDBC 4, de DataSources et de modules partagés.

Configuration en mode autonome

- Description des modes autonome et domaine.
- Les configurations autonomes (default, full, ha, full ha).
- Les outils de configuration : console Web, jboss cli.
- Modes de déploiement : autonome, partiel....

SERVEURS APPLICATIFS (Apache, Tomcat, Wildfly, ...) (suite)

4 jours / 28 heures

Contenu du programme

Suivi du serveur

- Les outils d'inspection intégrés.
- Outils de supervision JMX : jconsole , jvisualvm
- Paramétrage des logs d'accès Web.
- Exploitation des logs du serveur.
- Intégration de logs applicatifs avec Log4J et SLF4J

Administration d'un domaine

- Organisation d'un domaine.
- Services administratifs d'un domaine : Host Controller Slave ou Master, Process Controller.
- Configuration de plusieurs hosts de domaine.
- Déploiement d'applications en domaine avec la console web ou jboss cli

Optimisation des performances

- Réglage VM : profil client/serveur, mémoire heap ,
- Réglage des pools (EJB, DataSource, threads)
- Optimiser le chargement de classes : stratégie de chargement de modules, dépendances entre modules.
- Introduction à la haute disponibilité

Sécurité

- Principes de sécurisation du serveur.
- Gestion des accès aux outils d'administration.

Objectifs

- Connaître et comprendre le fonctionnement de Zabbix
- Savoir installer, configurer et administrer la solution
- Être en mesure de mettre en œuvre la supervision dans un environnement hétérogène

Contenu du programme

Présentation Zabbix

- Historique du produit, version, licence
- Systèmes supportés
- Les fonctionnalités de Zabbix : supervision réseau et serveurs (état des services, charge processeur, disques, ...)
- Système de configuration d'alertes
- Interface de supervision
- Reporting et visualisation des données collectées par zabbix
- Gestion des ressources (capacity planning)

Architecture

- Quelques définitions : notion d'hôte, de groupe d'hôtes, item, trigger, event, action, escalation, media, notification, template
- Les composants de l'architecture : Zabbix server, agent, proxy
- Java gateway pour la supervision JMX

Installation

- Configuration requise.
- Plates-formes supportées
- Site de référence

Configuration

- Gestion des hôtes et groupes d'hôtes, des items, triggers, event
- Gestion des notifications sur événements
- Visualisation
- Création de templates
- Les utilisateurs : configuration, groupes d'utilisateurs, droits d'accès

Applications types

- Supervision de services web, de machines virtuelles
- Auto-découverte des éléments réseau

Supervision distribuée

- Principe des proxy Zabbix
- Mise en œuvre d'une architecture distribuée

Objectifs

- Installer et configurer les composants de l'infrastructure virtuelle VMware vSphere (ESXi, vCenter)
- Configurer et gérer le réseau virtuel sous vSphere
- Créer, déployer et utiliser des machines virtuelles
- Sécuriser l'accès à l'infrastructure VMware
- Gérer les mises à jour de l'infrastructure virtuelle

Contenu du programme

L'infrastructure virtuelle

- Concepts de virtualisation des serveurs, des réseaux et du stockage.
- Les avantages de la virtualisation.
- Vue d'ensemble de VMware vSphere 7.
- Architecture de vSphere 7.
- Les composants de vSphere 7.

Le serveur VMware vCenter 7

- La Platform Services Controller.
- L'Appliance vCenter 7.
- Gestion des licences.
- Gestion des inventaires vSphere.

Gestion du réseau : vNetwork Standard Switch

- Les switches virtuels standards.
- Configuration réseau avancée (sécurité, gestion du trafic, gestion du teaming).

Gestion du stockage sous vSphere : vStorage

- Les différents protocoles de stockage.
- Le stockage SAN Fibre Channel.
- Le stockage SAN iSCSI.
- Le stockage NFS.
- Création et gestion des Datastores.

Création, déploiement et administration des machines virtuelles (VM)

- Éléments composant une VM.
- Création d'une nouvelle VM, clonage, snapshots, suppression et migration à froid.
- Création et gestion des templates.
- Gestion du matériel virtuel : Thin provisioning, VMDirectPath.
- Raw Device Mapping.

VMWARE VSPHERE (suite)

4 jours / 28 heures

Contenu du programme

Sécurité d'accès à l'infrastructure

- Configurer et administrer le pare-feu dans ESXi.
- Utiliser le Lockdown Mode.
- Intégrer ESXi à l'Active Directory.
- Sécurisation des accès: rôles et permissions.

Surveillance et gestion de l'utilisation des ressources

- Gestion des ressources d'une VM (Shares, limites, réservations).
- Les pools des ressources.
- Technologies d'optimisation de l'utilisation des processeurs et de la mémoire.
- Monitoring des performances.
- Surveillance de l'infrastructure avec les alarmes.



Programmes détaillés

PROJET

PROJET 2^{ème} PARTIE

1 jour / 7 heures

Objectifs

- Réaliser un projet en groupe reprenant les éléments techniques vues dans les bases de données et serveurs d'applications



Programmes détaillés

FONDAMENTAUX ET

LINUX

INTRODUCTION A LA CYBERSECURITE

2 jours / 14 heures

Objectifs

- Identifier les menaces informatiques liées à mon métier
- Sécuriser son matériel informatique et son poste de travail
- Acquérir les bons réflexes pour assurer la sécurité des données
- Comprendre la charte informatique de mon entreprise
- Comprendre et appliquer la PSSI

Contenu du programme

Introduction

- Pourquoi se sensibiliser à la cybersécurité ?
- Les notions clés à connaître
- Comprendre le cadre légal et réglementaire
- Les acteurs de la cybersécurité et leur rôle
- Les guides de l'ANSSI

Les principales menaces

- Les objectifs et acteurs des principales menaces
- Les impacts
- Les risques
- Quelques types d'attaques
- La fréquence des cyberattaques

Les mots de passe

- Quelques chiffres...
- Un bon mot de passe doit...
- Les bonnes habitudes à adopter
- Les gestionnaires de mots de passe
- Les e-mails
- L'arnaque la plus fréquente par email
- Quelques exemples d'hameçonnage
- Reconnaître une tentative d'hameçonnage
- Petit test !
- Les bons réflexes

A photograph of a man and a woman in a professional setting. The man, on the left, is bald, wearing glasses and a white button-down shirt, and is holding a tablet. The woman, on the right, has long brown hair and is wearing a light-colored button-down shirt and dark pants. They are both smiling and looking at the tablet together. The image is overlaid with a large purple triangle on the left side.

Programmes détaillés **DEVOPS**

DEVOPS ET INTEGRATION CONTINUE

3 jours / 21 heures

Objectifs

- Acquérir la terminologie, la structure, les outils et les concepts de base de la démarche DevOps
- Présentation des principes du Continuous Delivery

Contenu du programme

DEVOPS

- Historique du mouvement DevOps, origines et influences
 - . Qu'est-ce que DevOps ?
 - . Planète DevOps : tendances et mouvements émergents
- 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
 - . Ansible
 - . Terraform

DEVOPS ET INTEGRATION CONTINUE (suite)

3 jours / 21 heures

Contenu du programme

CI

- Les tests dans le processus de construction
 - . Intérêts de l'automatisation des tests
 - . Concepts de « l'extreme Programming »
 - . Principes des « Test Driven Development »
 - . Présentations des familles de tests
 - . Tests unitaires, tests d'intégrations et tests fonctionnels
 - . Exemple de test d'une application Web avec Selenium
- Qualité de code et traitement des résultats
 - . Apports de la mise en place d'outils de métriques
 - . Les principaux outils de métriques Java: de la génération de la documentation à la couverture de code en passant par la détection de bugs
 - . Exemple du site Maven
 - . Installation et configuration de Sonar
 - . Mise en œuvre d'une Qualimétrie avec Sonar pour un projet Maven et un projet Ant/Ivy
 - . Publication des résultats dans le serveur Hudson/Jenkins à travers ses plugins
 - . Les bonnes pratiques d'utilisation
- Communication
 - . Indicateurs de visibilité du résultat construit
 - . Statut, Santé et tendance
 - . Savoir réagir
 - . Communication avec des outils externes
 - . Notification par email et flux RSS
- Concepts et enjeux
 - . Les problématiques de la livraison logicielle (Déploiement manuel, Déploiement dans un environnement différent, configuration manuelle, etc)
 - . Qui est concerné?
 - . Définition du processus et l'amélioration de la chaîne de valeur
 - . Les bénéfices
 - . Mesurer son niveau de maturité et se fixer des objets par palier
- La gestion de configuration
 - . L'usage d'un outil de gestion de source
 - . La gestion des dépendances (internes et externes)
 - . La gestion de la configuration logicielle
 - . La gestion des spécificités des différents environnements

DEVOPS ET INTEGRATION CONTINUE (suite)

3 jours / 21 heures

Contenu du programme

De CI vers CD

- La gestion de configuration
 - . L'usage d'un outil de gestion de source
 - . La gestion des dépendances (internes et externes)
 - . La gestion de la configuration logicielle
 - . La gestion des spécificités des différents environnements
- «Continuous Integration» première étape du «Continuous Delivery»
 - . Rappel des principes et enjeux de l'intégration continue
 - . Liaison entre l'outil de gestion de sources, le build, les tests, le déploiement et la livraison
 - . Rappel des meilleures pratiques (TDD, Pre tested commit, etc)
- Continuous Delivery & Continuous Deployment
 - . Les différences impacts (versionning, marking des nouvelles versions, etc)
 - . Les cas d'usage et exemples concrets
- Chaîne (Pipeline) de déploiement
 - . Les principales étapes
 - . Mise en œuvre
 - . Le rôle d'un serveur d'intégration continue comme orchestrateur de déploiement
 - . Les critères de validation
 - . Illustration avec Jenkins
- Déploiement et Livraison Logicielle
 - . Illustration de la mise en œuvre d'un projet avec Maven et GIT

Objectifs

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

Contenu du programme

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.

ANSIBLE (suite)

4 jours / 28 heures

Contenu du programme

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

- 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

Contenu du programme

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 maquettage.
- La ligne de commande et l'environnement.
- Travaux pratiques

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

- 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

DOCKER (suite)

3 jours / 21 heures

Contenu du programme

Utilisation Docker Compose.

- Création d'un fichier yml de configuration.
- Déployer plusieurs conteneurs simultanément.
- Lier tous les conteneurs de l'application.
- Travaux pratiques

Interfaces d'administration

- L'API Docker et les Webservices.
- Interface d'administration en mode Web.
- Docker Registry : 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 ?

Orchestration et clusterisation

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



Programmes détaillés

PROJET

PROJET FINAL

4 jours / 28 heures

Objectifs

- Permettre aux participants de mettre en œuvre tout ce qu'ils ont appris au cours des sessions de formations précédentes.
- Savoir développer une architecture en couche à forte valeur ajoutée en privilégiant les interfaces.
- Apprendre à gérer les risques d'un projet et faire des choix de conception adaptés au problème.
- Apprendre à effectuer des tests de validation.
- Réaliser un ou plusieurs rédactionnels de suivi de projet.

Contenu du programme

Déroulement du module

- Préparation à la communication orale et écrite pour la soutenance du projet
- Les stagiaires travaillent en toute autonomie, en groupe. 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 groupe 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.



A VOTRE DISPOSITION POUR DISCUTER DU FUTUR

AJC INGENIERIE

6, rue Rougemont
75009 Paris
01 75 43 86 72
www.unjourunjob.com
www.ajc-ingenierie.fr

