

Catalogue des formations 2012

Pythagore F.D.
11, rue du Faubourg Poissonnière
75009 PARIS
Tél : 01 55 33 52 10 – Télécopie : 01 55 33 52 11 – Site : www.pythagore-fd.fr



Zoom sur quelques points forts en 2012 :

« Migration vers IPv6 » :

La norme d'adressage IP v6, ou IPng pour IP Next Generation, existe depuis 1993, et est actuellement largement utilisée en Asie et sur le continent américain. Une des limites d'utilisation du protocole IP v4 est le nombre d'adresses, qui est insuffisant pour l'utilisation actuelle des réseaux IP, et ce phénomène est accentué par l'usage croissant des technologies nomades.

Le dernier bloc d'adresses IP v4 a été alloué au mois de février 2011. La question n'est donc plus de savoir s'il faut engager les migrations, mais de savoir maîtriser la technologie, les outils, les méthodes de migration, de cohabitation entre les deux systèmes d'adressage, ...

Nous proposons des formations « **Migration vers IPv6** » de deux jours pour connaître les caractéristiques d'IPv6 et savoir élaborer une méthode de migration.

« Industrialisation des applications android » :

Avec l'arrivée à maturité des outils de développement pour google android, de nouvelles questions sont à l'ordre du jour : création d'une plate-forme privée de type "android market", télédistribution, gestion de flottes de tablettes ou de smartphones, centralisation des configurations, ...

Nous proposons des formations sur l'industrialisation des applications android, afin d'étudier les aspects essentiels pour la mise en oeuvre d'une gestion industrielle des développements sur android.

« Linux Embarqué » :

Les systèmes embarqués basés sur l'opensource se développent rapidement. Parmi les solutions éprouvées et fiables, figurent aujourd'hui, openWRT, emDebian, openEmbedded, Angstrom, openMoko.

Pour maîtriser la mise en oeuvre de systèmes Linux embarqués, il est nécessaire de savoir installer une chaîne de compilation croisée, compiler un chargeur comme U-Boot, compiler un noyau pour une carte spécifique, optimiser l'empreinte mémoire du noyau, etc ...

Notre stage : « **Systèmes Linux embarqués** » traite l'ensemble de ces sujets avec des travaux pratiques sur des cartes ARM11. Nous pouvons également réaliser des formations sur mesure adaptées à votre matériel.

Les grands classiques :

Nos filières de supervision et d'administration s'étendent avec un stage « **Reporting avec Cacti** » qui vient compléter les stages « **Administration Nagios** »;

les formations « **Administration JBoss** » intègrent la mise en oeuvre de JON (JBoss Operations Network).

En ce qui concerne la virtualisation, nous proposons des formations de présentation des différents outils de virtualisation sur Linux, mais également des stages d'expertise sur Xen et kvm, au cours desquels vous pourrez, par exemple, apprendre à configurer des réseaux virtuels avec VDI.

... et quelques nouveautés à découvrir :

Le serveur NGinx a atteint fin 2011 plus de 8% de parts de marché des serveurs Web. Nous vous proposons une formation de trois jours « **Administration serveur NGinx** » pour apprendre à configurer, exploiter et administrer NGinx, en serveur http, mais également en proxy.

Du côté du cloud, notre stage « **Architecture cloud d'entreprise** » qui aborde tous les points nécessaires à la création d'un cloud privé, avec des travaux pratiques sur Eucalyptus, se complète d'un stage d'une journée : « **Cloud, les technologies et enjeux** » pour avoir une visibilité technique et stratégique sur le cloud.

Nos domaines d'expertise :

De manière plus vaste, nous pouvons étudier toutes vos demandes de formation sur les domaines Unix/Linux, réseaux TCP/IP, développement Java/Jee, langage C, ...

Sur chacun de ces domaines, notre équipe possède une excellente maîtrise : nos formations vont de l'introduction à Linux, à la Sécurité, la Haute Disponibilité, ou le clustering Jboss, ...

Notre centre de formation :

Sur le plan pratique, notre centre de formation est situé 11, rue du Faubourg Poissonnière, à Paris (9^e), à deux pas des Grands Boulevards, à proximité de nombreux restaurants, hôtels, etc ...

Du côté technique, nos salles sont équipées de manière à pouvoir tester et mettre en oeuvre tous les concepts abordés en cours :

- 5 salles de formation,
- 60 postes clients PC quatre ou six coeurs, 4 Go Mo de mémoire, équipés de deux interfaces 100Mbps sur commutateur et une interface Wifi 802.11g+ à 108Mbps
- serveurs AIX , serveurs HP-UX , serveurs Sun UltraSparc;
- pour les formations VoIP, téléphones Linksys SPA901 avec configuration par synthèse vocale, avec support SIP v2, Codecs vocaux: G.711, G.723.1, G.729a;
- Accès réseau internet haut débit sur tous les postes (23 Mo de bande passante)

Nos différentes formules :

Nos formations sont proposées en inter-entreprises, mais également en intra sur site ou dans nos locaux.

Et nous proposons également une solution de formation à distance innovante : les classes virtuelles.

A la différence des outils d'auto-formation en e-learning, les classes virtuelles permettent l'interactivité entre les participants et le formateur.

L'innovation apportée par notre solution est la fourniture d'un environnement de travaux pratiques comme dans un véritable centre de formation : chaque participant dispose d'un poste distant sur lequel il peut réaliser les exercices et travaux pratiques. Le formateur peut également se connecter sur le poste, y effectuer des corrections, ou y déposer des fichiers, etc ...

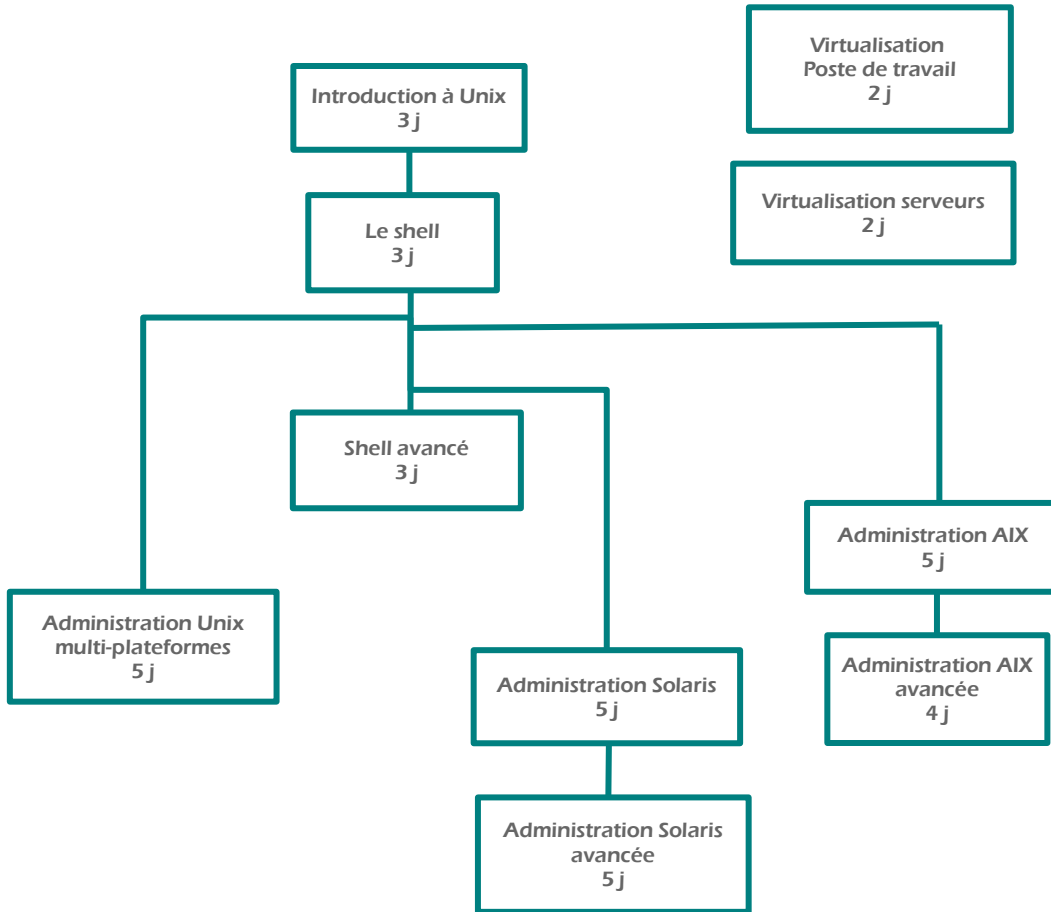
Cette solution, parfaitement adaptée aux formations techniques permet, par exemple, d'organiser des sessions multi-sites pour des entreprises ou organisations dont le personnel est réparti géographiquement sur plusieurs sites.

Les participants à ces stages sont très heureux de pouvoir bénéficier des avantages des formations en centre, tout en évitant les déplacements.

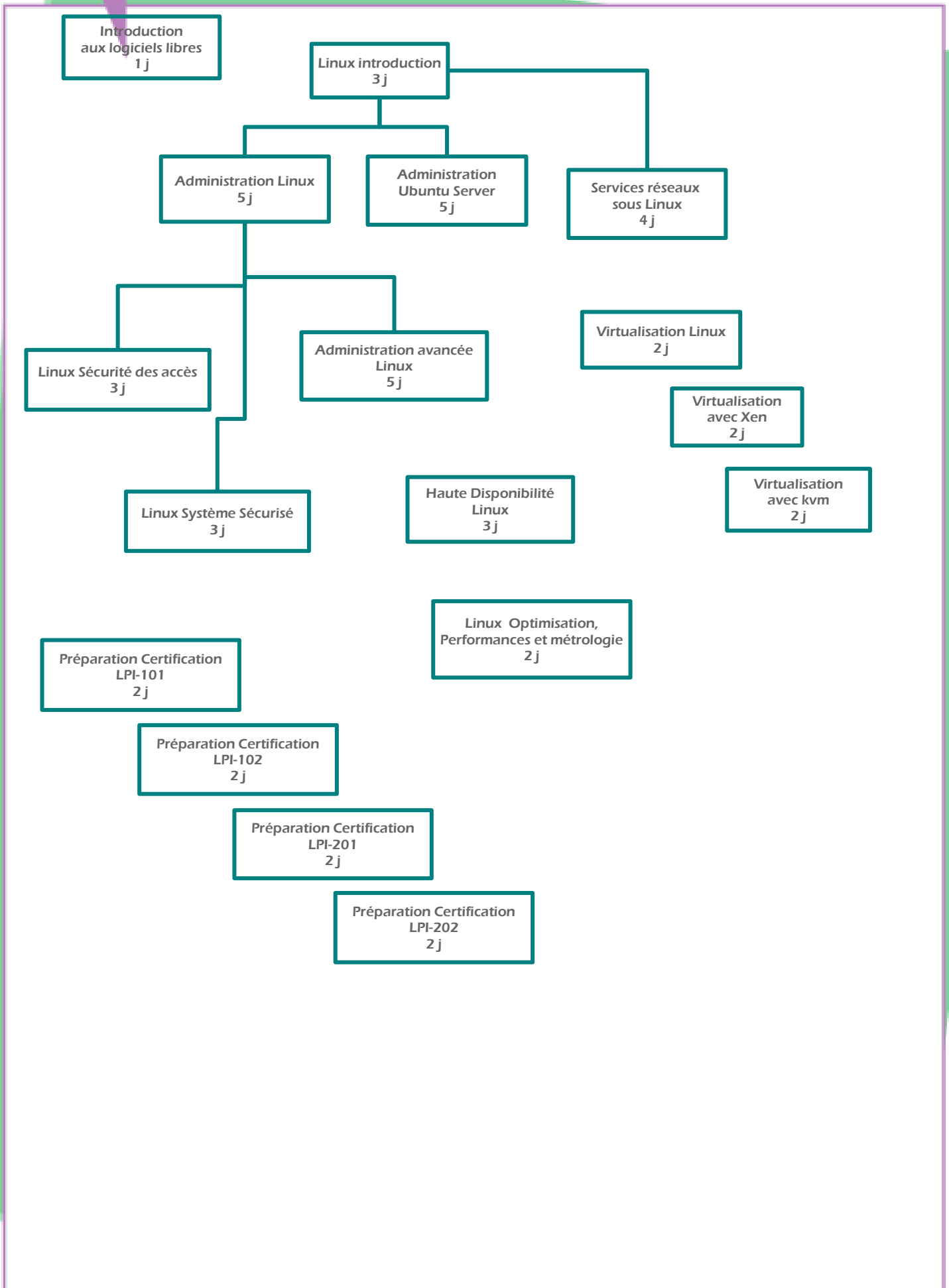
N'hésitez pas à nous contacter pour organiser vos classes virtuelles !

Si vous avez des besoins particuliers dans l'un de nos domaines d'expertise, n'hésitez pas à faire appel à nos services : nous pouvons élaborer des plans de formation et des supports de cours sur mesure.

01 55 33 52 10
www.Pythagore-fd.fr



Filières Systèmes Linux



Linux/unix introduction

Durée: 3 jours
1350 €

du 6 au 8 Février
du 19 au 21 Mars
du 18 au 20 Juin

du 10 au 12 Septembre
du 15 au 17 Octobre
du 26 au 28 Novembre

Public:

Utilisateurs, exploitants de systèmes Unix / Linux.

Objectifs:

Connaître les principes de fonctionnement du système Unix/Linux.

Connaissances préalables nécessaires:

Connaissances générales en informatique.

Programme:

Introduction	Présentation de Linux et Unix, définitions de base, version de noyau, distributions. Méthodes pour obtenir de l'information. Rappel rapide sur l'organisation d'un système. Arrêt/relance du système (shutdown, halt, reboot, sync). Commandes de base. Aide en ligne.
Interface graphique	Présentation : startx. Lancement d'applications, modification des menus, mini-applications d'interfaces Présentation de Gnome, KDE, WindowMaker, XFCE Travaux pratiques : configuration de l'interface graphique, lancement d'applications, modification des menus
Connexion	Principe de la connexion/déconnexion, les commandes en arrière-plan, les redirections (entree standard/sortie standard) Travaux pratiques : lancement de commande en arrière-plan, mise en œuvre des "pipes".
Les fichiers	Les types de fichiers. Chemin d'accès et nom de fichier. Manipulations de fichiers et de répertoires Recherche de fichiers : la commande find
Les filtres	Le mécanisme des tubes Exemples de commandes filtre Grep et expressions régulières
La sécurité	Le fichier des utilisateurs et le fichier des groupes Le mode d'un fichier. Modification des permissions.
Les processus	Gestion de la mémoire et des processus. Caractéristiques d'un processus Processus en arrière-plan. Les travaux batch.

Linux/unix introduction

Commandes shell standard

Commandes classiques : l'aide en ligne avec man,
l'arborescence : pwd, cd, mkdir; rmdir, ls, du, file,
manipulations de fichiers : cp, rm, mv, find, grep, ln, cat, more, ...
gestion des processus : ps, kill, date, who,
commandes d'environnement : tty, id, passwd, lpr, env, .

Travaux pratiques :

mise en œuvre des commandes étudiées.

Ajout de logiciels.

Utilisation de périphériques : lsmod, insmod, lspci

Impression : lpr, configuration.

Montage de cdrom : mount, umount

Configuration de /etc/fstab

Applications

Présentation des applications courantes:
bureautique, dessins, utilisateur WEB, réseau,
et des applications serveurs.

Le Shell

Durée: 2 jours
950 €

du 16 au 17 Février
du 22 au 23 Mars
du 24 au 25 Mai
du 21 au 22 Juin

du 26 au 27 Juillet
du 20 au 21 Septembre
du 25 au 26 Octobre
du 29 au 30 Novembre

Public:

Les développeurs d'applications sur UNIX, analystes d'exploitation, exploitants et administrateurs.

Objectifs:

Maîtriser la programmation en bourne Shell et en Korn Shell.

Connaissances préalables nécessaires:

Connaissance des principes de base du système UNIX.

Programme:

Généralités	Modes d'exécution d'un script Les alias Les métacaractères
Les variables	Les différents types de variables Transmission de paramètres Evaluation en ligne, tableaux Variables spéciales d'environnement
Les entrées/sorties	Lecture Ecriture Travaux pratiques : générer des fichiers de journalisation
Structures de contrôle	Les instructions conditionnelles. Gestion des boucles
Compléments	Les fonctions. Aperçu de AWK, grep et sed

Shell avancé

Durée: 3 jours
1350 €

du 26 au 28 Mars
du 30 Mai au 1er Juin

du 17 au 19 Septembre
du 3 au 5 Décembre

Public:

Les développeurs d'applications sur UNIX, et personnes chargées de la mise en production des applications, les exploitants, les administrateurs.

Objectifs:

Acquérir une parfaite maîtrise du shell. Connaître les outils disponibles et les bonnes pratiques concernant la programmation de scripts shell.

Connaissances préalables nécessaires:

Connaissance des principes de base du système UNIX et des commandes shell.

Programme:

Rappels	Fonctionnement du shell. Portées des variables: locales, globales, environnement.
Variables	Transmission de paramètres. Evaluation en ligne, tableaux. Variables spéciales d'environnement. Exploitation de la variable IFS.
Expressions régulières	Différences entre motifs et expressions régulières. Principe de fonctionnement.Exemples.
Structure	Expressions mathématiques, expressions conditionnelles Structure: tests avancés, boucles, choix multiples. Les instructions test et expr, expr sur les chaînes de caractères, Fonctions, structuration des scripts shell. Exportations. Autoprogrammation. Langages de macros: m4
Exécution	Programmation parallèle, programmation événementielle, gestion des alertes. Gestion des travaux. Exemple: création d'un système de surveillance de processus.
Exploitation	Redirections, interruption d'une procédure. Suivi de consommation. Debugging, tests, optimisation, ... Gestion des répertoires. Simplification des environnements de travail.
Programmation événementielle	Gestion des signaux. Debugging, Tests

Shell avancé

Gestion de données

Exploitation de la commande 'awk':
présentation, fonctionnalités,
syntaxe, variables (NR, FNR, FILENAME, ARGV, FS),
fonctions (length, substr, index, system, close, gsub,)
opérateurs de calcul, de test,
Expressions rationnelles,
Travaux pratiques :
recherches et calculs sur des fichiers. Expressions régulières.
Jointure et comparaison de fichiers: join, diff, diff3, sed

Administration UNIX multi-plateformes

Durée: 5 jours
2045 € HT

du 20 au 24 Février
du 2 au 6 Avril
du 4 au 8 Juin

du 27 au 31 Août
du 8 au 12 Octobre
du 17 au 21 Décembre

Public:

Exploitants et administrateurs de systèmes UNIX.

Objectifs:

Connaître les principes de fonctionnement système UNIX. Configurer et administrer un système (AIX, Solaris, HP-UX, Linux).

Connaissances préalables nécessaires:

Il est demandé aux participants de connaître les bases du système Unix.

Programme:

Introduction	Rappels sur le rôle de l'administrateur.
Installation	Installation du système : les différentes étapes, ajout de logiciel, utilitaires de livraison
Démarrage et arrêt du système	Procédure de boot (/etc/inittab), lancement des processus, montages au boot. Arrêt/relance du système (shutdown, halt, reboot, sync). Commandes de base. Aide en ligne.
Outils graphiques	Outils spécifiques : smit, wsm, sam, admintool, smc. Webmin : présentation, installation, configuration.
Gestion des Utilisateurs	Principes de base Fichier /etc/passwd, les shadow passwd Groupes, droits d'accès, politique d'accès adduser, useradd, passwd Contrôle des accès, contrôle des modifications. Quotas. Fichiers .profile et .cshrc. Choix d'un shell.
Systèmes de fichiers	Définitions : partitions, inodes. Organisation, gestion et maintenance : mkfs, mount/umount. Montage automatique. Visualisation des file system. Maintien de l'intégrité: fsck.
Sauvegardes	Définitions: sauvegardes, archivages, migrations. Utilisation des sauvegardes; périodicité incrémentales, complètes; restaurations Outils sauvegarde/archivage/compression:. tar, dump, restore, dd, cpio, rsync Sauvegarde du système
Programmation de tâches	Besoin, définition Outils : at, cron, utilisation de la crontab.

Administration UNIX multi-plateformes

Services d'impression	Lancement/arrêt des services d'impression. déclaration d'une imprimante. /etc/printcap gestion des files d'attente
Réseau	Les différents paramètres à prendre en compte. Configuration du nom de l'adresse du système Commandes génériques. Visualisation des paramètres locaux. Routage. Fichiers de définitions statiques.
Services de noms	DNS : introduction, configuration du service
NFS	Partage de fichiers, mise en place d'un serveur, configuration client, sécurisation NFS TP: mise en place d'un partage de fichiers entre plusieurs machines
Automonteur	Principe et travaux pratiques avec /home
X11	Environnement graphique. Principe de base. XDM. Contrôle d'accès.
Supervision	Mise en œuvre de rrdtool. Création de graphes.

Administration AIX

Durée: 5 jours
2045 € HT

du 6 au 10 Février
du 21 au 25 Mai

du 10 au 14 Septembre
du 19 au 23 Novembre

Public:

Exploitants et administrateurs de systèmes AIX.

Objectifs:

Savoir configurer, administrer un système AIX.

Connaissances préalables nécessaires:

Connaissance des principes de fonctionnement d'un système unix, et le shell

Programme:

Introduction	Rappel : le rôle de l'administrateur. La norme POSIX. Les systèmes UNIX, le positionnement d'AIX.
Installation du système	Etapes. Installation de logiciels. Suppression. Ajout de logiciels. Recherche d'un package Procédure de démarrage du système, fichiers de démarrage : /etc/inittab Services. Sous systèmes.
Utilisateurs	les fichiers de configuration, les notions d'utilisateurs, de groupes, la gestion des mots de passe. Les rôles, les quotas. Le répertoire /etc/security: /etc/security/passwd
Système de fichiers	les types de systèmes de fichiers : JFS, CDRFS, NFS, UFS montage et démontage de systèmes de fichiers : le fichier /etc/filesystems. ajout/modification des caractéristiques d'un système de fichiers. Prise en charge des fichiers volumineux (>2Go) Auto-Monteur.
Gestion des volumes	Fonctionnalités. Mise en place. les volumes physiques : ajout, liste du contenu, modification les groupes de volume : activation, désactivation (varyon / varyoff); importation, exportation; les groupes de volumes en miroir; les volumes logiques : ajout, retrait, copie. crfs. Redimensionnement.
Périphériques	le répertoire /dev
Impressions	Fonctionnement. Installation. Imprimante réseau. Les commandes d'impression, la gestion des travaux, des files d'attente, d'un serveur d'impression : Ajout d'imprimantes, configuration. Suivi du spooler, /etc/qconfig, lpq/lpr
Pagination	ajout d'un espace de pagination. Modification dynamique d'un espace de pagination. Activation d'un espace de pagination

Administration AIX

Réseau	Les différents paramètres à prendre en compte. Configuration de l'interface. Configuration du nom de l'adresse du système
NFS	partage de fichiers. Configuration d'un serveur NFS, d'un client NFS. Format du fichier d'export. Sécurisation NFS
Sauvegardes/ restaurations	Utilisation des sauvegardes; périodicité. Incrémentales, complètes; restaurations. Méthodes et techniques. Les outils standards : tar, cpio Création d'une image système.mksysb, mkcd Lecture/vérification d'image iso
Programmation de tâches	at : exécution différée, cron : pour le séquençement des travaux (jobs) Présentation des outils d'automatisation du marché
Surveillance	Détection des problèmes, analyse des fichiers de logs.errdemon,alog.
Outils	La gestion de la documentation : HYPER, Administration : SMIT, et par un navigateur : WSM, ODM : principe de fonctionnement

Administration AIX avancée

Durée:4 jours

Prix et dates: nous consulter

Public:

Exploitants et administrateurs de systèmes AIX.

Objectifs:

Savoir optimiser la configuration d'un système AIX, Savoir contrôler les performances du système et connaître les outils de mesure et d'optimisation.

Connaissances préalables nécessaires:

Les principes d'administration d'un système AIX doivent être acquis avant d'assister à ce cours.

Programme:

ODM	Les bases ODM.Administration ODM. Gestion des informations. Classes et objets. Fichiers de configurations. Création d'objets: odm_create_class, odm_add_obj, odmcreate, odmdrop. Création d'une nouvelle base Les commandes en mode ligne: odmdelete, odmadd, odmchange, odmshow,odmget.
Installation AIX	Lancement. Options d'installation. Menu post-install. Fichiers d'installation. SMS/AIX.
Installation réseau	Installation AIX en réseau, principe de NIM Mise en place d'un maître NIM: nim_master_setup, nimconfig. Informations sur l'environnement NIM.Clients NIM.NIM avec wsm. nimol. Mise en œuvre.
Configuration	Matériel installé.Configuration de matériel.Connexions ODM.Informations sur le système Informations sur le noyau.Passer en noyau 64 bits Console système
Disques	comparaison de systèmes de fichiers sur deux machines différentes. Mise en œuvre de rdist et install. distfile. Réparation d'une image de boot corrompue.
Partitionnement physique	Répartition des accès disques: RAID 1, RAID 5. Utilisation des LVM. Création d'une image statique d'un volume physique, remise en place de cette image avec splitvg et joinvg. Le micro-partitionnement avec DLPAR
Partitionnement logique	LPAR : principe. Versions.

Administration AIX avancée

Accès	<p>Le fichier /etc/security/user. Paramètres avancés de chuser , lsuser, mkuser et rmuser . Gestion des mots de passe. Gestion des rôles. Fichier login.cfg. Programme de connexion. Définition d'une séquence SAK.</p> <p>Autorisation de connexion. Masque applicatif. Localisation du compte. Accès à su. Shell sécurisé/base TCB</p>
Performances	<p>La mesure des performances : les temps de réponse.</p> <p>Les commandes de base :sar, vmstat,iostat. Le démon sadc.</p> <p>Présentation des ressources et impact sur les performances : la mémoire, l'espace disque, les accès disques, la cpu</p> <p>Notion de charge machine. Entrées/sorties, disques fixes, threads et mémoire. Mode de fonctionnement: serveur, multi-utilisateur ou station de travail. Restreindre l'utilisation de ressources: le fichier limits, ulimit.</p> <p>Utilisation du Performance Diagnostic Tool. Structure du PDT, installation et activation. Configuration avec pdt_config.</p>
CPU	<p>Commandes de base: time, timex. Mesure de la consommation de programmes: tprof. Mesure de la consommation des threads du noyau: pprof. Mesure par l'accounting: prtacct, acctprc1, acctprc2.</p>
Mémoire	<p>Utilisation de la commande vmstat. Les options -l et -s. Mesure de la mémoire utilisée: svmon.</p> <p>Analyse de la charge mémoire VMM avec schedtune et vmtune.</p>
Disques	<p>Commandes de base: df, du. Mesure avec acctmerge. La commande dodisk et le fichier dacct.</p>
Le diagnostic	<p>Fichiers de logs : syslog, analyse par errpt</p>
Traces	<p>Utilisation de traces systèmes. Création d'un fichier trace. Exploitation avec trcrpt, filters.</p>

Administration Solaris 9 et 10

Durée: 5 jours
2045 €

du 30 Janvier au 3 Février
du 16 au 20 Avril

du 9 au 13 Juillet
du 1er au 5 Octobre

Public:

Exploitants et administrateurs de systèmes Solaris.

Objectifs:

Savoir installer, configurer, administrer un système Solaris. Les participants travailleront sur des serveurs Sparc et des serveurs x86.

Connaissances préalables nécessaires:

Connaissance des principes de fonctionnement d'un système Unix, et le shell

Programme:

Introduction	Rappels : le rôle de l'administrateur. Introduction à Solaris 10. Principaux apports de solaris 10.
Installation	Installation de Solaris a partir d un dvd ou du reseau. Ajout de logiciel : pkgadd Autoconfiguration du système : principe, périphériques
Démarrage/arrêt	Etapes de lancement, procédure de boot Niveaux de boot, fichier /etc/inittab Montages au boot : /etc/vfstab Arrêt du système : halt, reboot, shutdown Configuration matérielle
SMF	Service Management Facility: Présentation, utilisation. Gestion des services.
Utilisateurs	Fichier /etc/passwd Principe des shadow passwords Création de comptes utilisateurs, valeurs par défaut Fichiers profils , choix d'un shell : .cshrc, .profile, gestion des roles (RBAC)
Fichiers	Les inodes. Gestion des partitions. Les différents systèmes de fichiers, le système UFS, le système ZFS Commandes : format, fsck Création d'un système ZFS Le montage du système de fichiers. Montage automatique, mount/umount. Les pseudo-systèmes de fichiers les ACLS sous Solaris
Sauvegardes	Définitions, outils standards : tar, cpio, dd Commandes ufsdump/ufsrestore
Ordonnancement	At, cron, fichier crontab

Administration Solaris 9 et 10

NFS	Network File System Installation d'un serveur de fichiers : configuration serveur, client. Sécurisation.
Impressions	Configuration des services d'impression Installation d'une imprimante
X11	Principe de fonctionnement,ssh forwarding
Introduction à la virtualisation	Zones: présentation, définitions. Création d'une zone minimale. Création d'une zone complète. Utilisation, administration de zones.
Administration au quotidien	Administration graphique. Surveillance des logs, processus, mémoire, fichiers. Syslog, coreadm dumpadm.

Administration avancée Solaris 9 et 10

Durée: 5 jours
2080 €

du 27 Février au 2 Mars
du 21 au 25 Mai

du 27 au 31 Août
du 26 au 30 Novembre

Public:

Administrateurs de systèmes Solaris.

Objectifs:

Savoir installer, configurer, administrer un parc de systèmes Solaris. Gérer une infrastructure Solaris cluster. Les participants travailleront sur des serveurs Sparc et des serveurs x86.

Connaissances préalables nécessaires:

Connaissance des principes de fonctionnement d'un système unix, du shell et de l'administration Solaris de base.

Programme:

Introduction	Rappels : le rôle de l'administrateur.
Installation avancée	Installation en réseau. Jumpstart : principe de fonctionnement et configuration. Travaux pratiques : configuration d'un serveur d'installation automatisée jumpstart.
Virtualisation sous solaris	Rappels sur les différentes techniques de virtualisation. Présentation des solutions sur Solaris : Xen, virtual box et Zones. Les Zones Solaris : principe, zones globales et non-globales. Travaux pratiques : création, configuration et installation d'un système d'exploitation dans une zone. Connexion , et utilisation avec zlogin. Xen : Présentation, la paravirtualisation. Installation de Xen, et configuration des domaines utilisateurs. création d'un domU, paramétrage Gestion des ressources. VirtualBox : principe de fonctionnement Installation sur Solaris, création et configuration de machines virtuelles Démarrage de VirtualBox dans une zone Solaris.
Configuration système avancée	/etc/system modload/modunload, Swap Space, Core Dumps
Filesystems et gestion de volumes	Svm, zfs, fssnap, rappels sur les sauvegardes Le système de fichiers ZFS (ZettaByte File System) Présentation, principe de fonctionnement Création d'un système ZFS , montage, et mise en place d'un montage automatique dans

Administration avancée Solaris 9 et 10

Analyse de performances	<p>Les ressources à contrôler Outils de mesures : vmstat, iostat, mpstat Prise de traces : sar, dtrace Actions : répartition des charges avec l'ordonnanceur FSS (Fair Share Scheduler) et avec l'ordonnanceur TS (TimeSharing Scheduler) Travaux pratiques : mise en place du FSS contrôle des ressources processeurs</p>
NFS	<p>Configuration avancée et analyse de comportement. Etude des fichiers /etc/dfs/dfstab et /etc/dfs/sharetab. Sécurisation : trace des accès NFS, différentes méthodes de protection.</p>
DNS et LDAP	<p>Installation et configuration d'un serveur DNS et d'un serveur LDAP.</p>
Réseau et sécurité	<p>Rappels sur le routage, mise en place d'un firewall, services en mode sécurisé, commandes et fichiers de configuration.</p>

Introduction au logiciel libre

Durée: 1 jour
465 € HT

le 2 Mars
le 27 Avril
le 6 Juillet

le 26 Octobre
le 7 Décembre

Public:

Chefs de projet, architectes et toute personne souhaitant avoir une vision claire du monde Open Source.

Objectifs:

Présenter les caractéristiques et le mode de fonctionnement du logiciel libre, les solutions disponibles, les critères de choix.

Connaissances préalables nécessaires:

Connaissances générales sur les systèmes d'information.

Programme:

Introduction	Le principe du logiciel libre, historique. La notion de licence, la Free Software Foundation Les licences "ouvertes" : GPL, BSD, X11, Les projets OpenSource : GNU, hurd, FreeBSD, ...
La place de l'Opensource aujourd'hui	Quelques chiffres, les projets significatifs. Les enjeux économiques. L'offre de services (éditeurs, SSL). Les sites de références, de documentation. Les apports, les risques, les critères de choix.
Le système GNU/Linux	Origine, historique La place de Linux dans l'OpenSource : définitions : noyau, distributions Solutions disponibles sous Linux : en serveurs (produits réseaux, systèmes), sur postes clients (outils bureautiques, client web, messagerie, ...), dans le monde de l'embarqué (openWRT, android).
Le réseau	La place de l'Open Source dans les réseaux IP : logiciels d'infrastructure, de supervision. Les produits phares : bind (DNS, apache, nagios, squid).
Les outils JEE	Rappel de l'architecture JEE, positionnement des outils opensource : apache, jetty, tomcat, JBoss, jonas, geronimo
Les environnements de développement	Eclipse, Netbeans, Ant Python, Ruby, RubyOnRails, le SDK sur android.
Liens avec les solutions propriétaires	Gestion de postes clients hétérogènes en entreprise : compatibilité OpenOffice , formats de fichier, clients messagerie Environnement de développement : plugins Les limites de l'offre Open Source : les logiciels métiers

Administration Linux

Durée: 5 jours
2045 €

du 13 au 17 Février
du 23 au 27 Avril
du 25 au 29 Juin

du 3 au 7 Septembre
du 22 au 26 Octobre
du 10 au 14 Décembre

Public:

Administrateurs, et toute personne souhaitant maîtriser l'installation, la configuration d'un système Linux.

Objectifs:

Savoir installer, administrer un système Linux. Chaque participant dispose des différentes distributions (Debian, Redhat) et peut, si il le souhaite, tester les travaux pratiques sur le système de son choix.

Connaissances préalables nécessaires:

Des connaissances de base des systèmes unix et/ou linux sont nécessaires, ainsi que du shell.

Programme:

Introduction	Linux et l'opensource : historique, caractéristiques de linux Les distributions, les différences et points communs. Rappel rapide sur l'organisation d'un système. Arrêt/relance du système (shutdown, halt, reboot, sync).
Installation	Les phases d'installation d'un système Linux. Options dans les chargeurs : grub. Les outils d'installation. Gestions de paquets. Les différentes méthodes. RPM, le système RedHat : historique, présentation et fonctionnement de la commande rpm, principales options pour l'installation, l'interrogation, l'affichage du contenu d'un paquet, etc ... Travaux pratiques : requêtes d'interrogation des packages rpm, installation et mise à jour de packages. Le paquetage Debian : fonctionnalités, format et statut des paquetages, les applications de gestion (dpkg, dpkg-deb, dpkg-query, apt, ...) Travaux pratiques avec dpkg : extraction des informations concernant un paquet. Présentation des outils : apt, yum, urpmi.
Environnement graphique	Présentation, gestionnaire de fenêtres. Différentes solutions : gnome, KDE, Windowmaker, xfce.
Outils d'administration	Webmin: Présentation, installation, configuration. Démonstration.

Administration Linux

<p>Systèmes de fichiers</p>	<p>Définitions : inodes, filesystem, partition Organisation, gestion et maintenance : utilisation de la commande mkfs. Principe du montage d'un périphérique. Travaux pratiques : mise en place d'un montage à l'initialisation du système (/etc/fstab) et d'un montage temporaire (commande mount). Exploitation et maintien de l'intégrité des systèmes de fichiers : commandes mkfs, mount, umount, df. Travaux pratiques : comparer le résultat des commandes df et du Test de montage d'un système de fichiers sur un point d'ancrage non vide. Présentation de différents types de systèmes de fichiers : ext3, reiserFs, xfs, jfs. Les autres systèmes de fichiers: fat, vfat, nfs, smb. Partition: création d'images de partitions. Synchronisation de données. Chiffrement des données.</p>
<p>Utilisateurs</p>	<p>Etude des fichiers /etc/passwd, /etc/group, /etc/shadow. Gestion des comptes utilisateurs: useradd, usermod, userdel, passwd, gestion des groupes : groupadd, groupdel, ajout d'utilisateurs, création d'administrateurs de groupes, droits d'accès, politique d'accès. Travaux pratiques : création d'utilisateurs et de groupes, puis vérification de cohérence avec la commande pwck. Contrôle des connexions de root : les objectifs et les méthodes. Travaux pratiques : utilisation de l'outil "john the ripper" pour la recherche de mots de passe. Introduction à PAM : Pluggable Authentication Modules.</p>
<p>Processus</p>	<p>Les processus. Les threads. Gestion des priorités. Utilisation des pseudo-processus /proc: stat, cpuinfo, ...</p>
<p>Sécurité des données</p>	<p>Sauvegardes Outils sauvegarde/archivage/compression: gzip, zip, tar, dd, cpio, dump, restore. Sauvegarde du système, création de CD de secours. Travaux pratiques : sauvegarde par cpio, réalisation d'un archivage par tar. Tests de restauration des données. Synchronisation des données par rsync sur des serveurs distants.</p>

Administration Linux

Impressions	<p>Les services d'impression, démarrage/arrêt des services d'impression.</p> <p>Présentation de CUPS : Common Unix Printing System</p> <p>Définitions : classes d'imprimantes, classes implicites, destination, filtres, backends.</p> <p>Installation d'une imprimante, modification d'un pilote : lpr, cups, printtool, system-config-printer.</p>
Programmation de tâches	<p>Le besoin, l'automatisation des tâches système.</p> <p>Exécution différée avec at.</p> <p>Programmation de tâches avec cron.</p> <p>Etude du fichier crontab.</p> <p>Les produits du marché : openPBS, fcron</p>
Exploitation	<p>Journaux : /var/log/messages</p>
Réseau IP	<p>Les objets à configurer : les interfaces réseaux, les routes, le DNS.</p> <p>Principe de la configuration dynamique ou statique.</p> <p>Configuration, nommage/activation des interfaces réseau, drivers.</p> <p>Etude des fichiers /etc/hosts, /etc/nsswitch, /etc/resolv.conf.</p> <p>Travaux pratiques : création d'une interface réseau, visualisation, configuration de plusieurs adresses IP sur la même interface physique.</p> <p>Ajout d'une route, d'un hôte, d'un serveur DNS, et tests.</p> <p>Présentation des utilitaires ssh, clients windows (Putty, WinSCP)</p>
NFS	<p>Fonctionnalités : partage de fichiers en réseau, avec gestion de la sécurité.</p> <p>Description du fonctionnement client/serveur.</p> <p>Etude du fichier /etc/exports.</p> <p>Travaux pratiques : configuration d'un serveur NFS sur chaque poste, et configuration des clients NFS pour tester les accès.</p>
Intégration système d'information	<p>Samba: Principe.</p> <p>Intégration de SMB au niveau des couches réseaux.</p> <p>Fonctionnalités : partage de répertoires, d'imprimantes, création de comptes, etc</p> <p>Travaux pratiques : Installation et configuration de samba pour le partage de fichiers.</p>

Les services réseaux Linux

Durée: 4 jours
1760 €

du 5 au 8 Mars
du 11 au 14 Juin

du 1er au 4 Octobre
du 3 au 6 Décembre

Public:

Administrateurs systèmes et réseaux.

Objectifs:

Savoir installer, configurer et sécuriser les principaux services réseaux sur Linux.

Connaissances préalables nécessaires:

Des notions de base sur le système Unix ainsi que sur TCP/IP sont souhaitées.

Programme:

Configuration IP	Présentation. Activation du réseau. Interfaces réseau. Routage. Fichiers de configuration.
Outils réseau	Outils de trace Tcpcdump Outils de diagnostic
Les services	Serveurs de configuration dns, dhcp, bootp (présentation de kickstart) Serveurs de fichiers : nfs, ftp, tftp, http Serveurs d'accès : routage, firewall, proxy
DHCP	Définition , principe. Configuration poste client., serveur. Notion de bail. configuration avancée. Redondance DHCP.
DNS	Définition, fonctionnement. Travaux pratiques avec bind : configuration client et serveur.
FTP	File transfer protocol Travaux pratiques : mise en œuvre de ftp: configuration, droits sur les répertoires, gestion des utilisateurs, surveillance, fichiers de logs
NTP	Définition.Fonctionnement. Déclaration d'un point de synchronisation. Configuration d'un serveur d'horloge. Configuration de clients. Architecture. Contrôles d'accès. Implémentations de NTP.
Messagerie	Les protocoles, POP3 et IMAP4 Structure des messages. Présentation de Postfix, installation et configuration. Fichiers master.cf, main.cf

Les services réseaux Linux

Serveur Web	Installation Apache : configuration de base, configuration multi-sites (httpd.conf) Suivi: access_log, error_log Principe des scripts CGI, et des modules.les hôtes virtuels
NFS	présentation, fonctionnement, configuration d'un serveur NFS et des postes clients partage de fichiers Les groupes de confiance, méthodes de protection.
Intégration hétérogène	interconnexion Unix/Windows: samba : configuration et installation accès depuis des clients windows, et Linux en mode texte : smbclient, smb4K : le navigateur samba
Sécurisation des accès réseau	Connexion directe, distante, liste des points d'entrée dans le système. Analyse des fichiers journaux du réseau Vérification de l'intégrité du système à l'aide des outils: tcpdump, sniffit, cop, satan ping, traceroute
Administration distante	Webmin : installation, présentation de l'interface et des fonctionnalités de webmin

Haute disponibilité Linux

Durée: 3 jours
1430 €

du 20 au 22 Février
du 9 au 11 Mai

du 10 au 12 Septembre
du 19 au 21 Novembre

Public:

Administrateurs Linux, ou toute personne souhaitant mettre en œuvre un système Linux avec des contraintes de haute disponibilité.

Objectifs:

Connaître et savoir mettre en œuvre les mécanismes disponibles sur Linux pour offrir un service continu.

Connaissances préalables nécessaires:

Une bonne connaissance d'un système Unix est nécessaire. Des notions d'administration sont souhaitées.

Programme:

Introduction	<p>Le besoin : pourquoi la haute disponibilité, mesure de la disponibilité.</p> <p>Quelques définitions : tolérance aux pannes, fail-over, RAID, Mirroring, redondance, MTBF, etc ...</p> <p>Les acteurs du marché, positionnement de Linux.</p> <p>Présentation de l'architecture LVS.</p> <p>Les solutions de haute disponibilité.</p>
Clustering	<p>Les différentes fonctions de clustering :</p> <p>répartition des accès disques, répartition de la charge CPU, basculement automatique ou programmé sur un autre processeur, exécution simultanée sur plusieurs processeurs.</p>
Adresses réseaux	<p>Principe du basculement d'adresses.</p> <p>Solution avec Fake.</p> <p>Agrégation d'interfaces réseau.</p> <p>Travaux pratiques :</p> <p>mise en place de l'agrégation avec deux cartes réseaux ethernet.</p> <p>Configuration dynamique et configuration statique.</p> <p>Test et vérification dans les fichiers journaux.</p>
Linux Virtual Server	<p>Architecture : pacemaker, ldirector, heartbeat, fake, coda</p> <p>Utilisation de mon pour la détection des services défectueux.</p> <p>Travaux pratiques :</p> <p>Installation, configuration de heartbeat et ldirector</p> <p>Configuration de Pacemaker pour la gestion du cluster.</p>
IPVS	<p>Présentation : IP Virtual Server.</p> <p>Répartition de charge.</p> <p>Contrainte au niveau du noyau.</p> <p>Travaux pratiques :</p> <p>préparation d'un noyau IPVS, configuration passerelle.</p> <p>Mise en place d'un cluster.</p>

Haute disponibilité Linux

Idirectord	<p>Présentation : Linux director daemon. Fonctionnalités. Travaux pratiques : installation et configuration de Idirectord</p>
Applications	<p>Intégration LVS avec Keepalived. Architecture, prérequis du noyau. Travaux pratiques : Installation et configuration keepalived. Gestion de ressources avec Pacemaker. Présentation de la RedHat Cluster Suite. Répartition de requêtes http, gestion des sticky session. Répartition de charges, routage de niveau 7. Présentation des solutions WebSphere, JBoss et Jonas. Travaux pratiques : Mise en œuvre du répartiteur de charge HAProxy en mode HTTP.</p>
Données	<p>Le besoin, les différentes solutions techniques : réplication de données en réseau, ou en local. Exemples de Coda, Logical Volume Manager. Le RAID, RAID logiciel sous Linux : raidtool, mdadm. Mise en œuvre de disques SCSI partagés entre machines Les systèmes de fichiers haute disponibilité : DRDB (Distributed Replicated Block Device) La réplication des données avec PostGreSQL, MySQL</p>

Linux système sécurisé

Durée: 3 jours
1430 €

du 6 au 8 Février
du 2 au 4 Mai

du 27 au 29 Août
du 15 au 17 Octobre

Public:

Toute personne souhaitant mettre en place une sécurité optimale sur un système Linux, et plus particulièrement les administrateurs système et sécurité.

Objectifs:

Savoir configurer les mécanismes de sécurité de Linux.

Connaissances préalables nécessaires:

Une bonne connaissance de l'administration des systèmes Unix/Linux est nécessaire.

Programme:

Introduction

Le besoin, définition du D.I.C.
Les attaques possibles.
Evaluation des risques.
Méthodes de protection.

Gestion utilisateurs

Rappels sur les notions de base de sécurité sur Unix:
modes d'accès, comptes utilisateurs, groupes, utilisateurs génériques de gestion de ressources.
Fichiers /etc/passwd, /etc/group, /etc/shadow.
Codage des mots de passe.
Création, modification, suppression de comptes utilisateurs.
Gestion des groupes :
ajout , retrait d'utilisateurs, création d'administrateurs de groupes.
Affectation d'un mot de passe au groupe.
Vérification de cohérence : pwck.
Connexions du compte root, contrôle de connexions.
Outil de recherche de mots de passe.
Travaux pratiques :
installation et mise en œuvre de l'outil "John the ripper" en mode "single-crack".
Prise de privilèges: sudo, sudoers.

Authentification

pam: gestion des modules d'authentification.
Présentation et exemples d'utilisation.
Principe de base, configuration.
Les modules : différents types de modules (auth, account, session, password).
Notion de pile de modules.
Travaux pratiques :
mise en œuvre de PAM et de quelques modules parmi les plus courants :
access, chroot, cracklib, env, ftp, groups, limits, listfile, mkhomedir, tally, time, unix, wheel

Linux système sécurisé

Sécurisation traitements	<p>Les risques : le déni de service, exemples de virus sur un système Linux.</p> <p>Travaux pratiques : exploitation d'un débordement de pile.</p> <p>Les moyens de détection, la surveillance, les traces : syslog, l'accounting.</p> <p>L'audit de sécurité.</p> <p>Méthodes de protection : démarche sur les systèmes Linux.</p>
Sécurité du noyau	<p>Les différentes approches de sécurisation du noyau.</p> <p>Présentation de GrSecurity et SELinux.</p> <p>Travaux pratiques avec GrSecurity : installation, configuration du noyau, paramétrage du niveau de sécurité.</p> <p>Administration avec grAdm2.</p> <p>Génération d'une politique : learning mode.</p> <p>Mise en place des règles d'ACL.</p> <p>L'ACL GrSec.</p> <p>Restrictions d'accès aux appels systèmes. Masquage de processus.</p> <p>Visibilité du répertoire /proc.</p> <p>Restrictions chroot.</p> <p>SELinux : principe, configuration du noyau, options du noyau.</p> <p>Travaux pratiques : définition d'une politique de sécurité.</p> <p>Installation et activation de la politique de sécurité dans le fichier /etc/selinux/config.</p>
Sécurité des données	<p>Contrôle de la cohérence du système de fichiers : fsck.</p> <p>Procédure de vérification.</p> <p>Sauvegardes : définitions</p> <p>Commandes et outils standards.</p> <p>Utilisation des sauvegardes pour la disponibilité des données.</p> <p>Outils sauvegarde/archivage/compression: gzip, zip, tar, dump, restore, dd, cpio, rsync</p> <p>Service d'urgence pour Linux : en cas de problème au démarrage du système, utilisation d'un système tiers : "systemRescue CD"</p> <p>Travaux pratiques : création de CD de secours.</p>
Sécurité système de fichiers	<p>Sécurité: mise en place des contrôles d'accès</p> <p>ACL : principe des listes de contrôle d'accès POSIX.</p> <p>Travaux pratiques : mise en place des ACL sur xfs</p> <p>Les quotas : principe, mise en place dans le fichier /etc/fstab.</p> <p>La commande edquota pour l'édition, et le paramétrage, et la commande quota pour la visualisation.</p> <p>Travaux pratiques : mise en place des quotas</p>

Linux sécurité des accès

Durée: 3 jours
1430 €

du 15 au 17 Février
du 14 au 16 Mai

du 5 au 7 Septembre
du 12 au 14 Novembre

Public:

Toute personne souhaitant sécuriser les accès à un système Linux

Objectifs:

Savoir configurer les mécanismes de sécurité réseau de Linux.

Connaissances préalables nécessaires:

Une bonne connaissance de l'administration des systèmes Unix/Linux et des réseaux TCP/IP est nécessaire.

Programme:

Introduction	Le besoin, définition du D.I.C. Les attaques possibles. Evaluation des risques. Méthodes de protection.
Les ports de niveaux 5	Rappels sur la notion de port. Les ports UDP et les ports liés au réseau. Exemples de trames.
Outils de captures réseau	Les analyseurs de trames : tcpdump, wireshark. Travaux pratiques : mise en œuvre de tcpdump, options usuelles, et possibilités de filtrage. Installation de Wireshark, capture et analyse de paquets.
Outils de Diagnostic	Scanners de ports, outils d'audit externe, et d'audit interne. Exemples de nmap, hping, sniffit, ...
Audit réseau	Nessus : principe de fonctionnement, installation. Travaux pratiques : réalisation d'un audit réseau avec Nessus.

Linux sécurité des accès

Sécurisation des accès réseau	<p>Protection de services réseaux au travers de xinetd. Les tcp-wrappers: telnet, tftp, snmp, ftp, pop3s, imap4s Les contrôles d'accès : Etude des fichiers /etc/hosts.allow et /etc/hosts.deny Les accès réseaux : sftp, les r-commandes (rlogin, rsh) Sécurisation des transferts de fichiers avec vsftp Présentation d'openSSH. Travaux pratiques : configuration du serveur et du client pour la mise en place d'un tunnel X11 et ssh. Sécurisation http (apache) : lors de l'exécution des processus (directives user et group), portée des balises, restriction d'accès par méthode : balise Limit, LimitExcept, le fichier .htaccess : autorisation ou restriction d'accès. Authentification HTTP. Création d'utilisateurs avec htpasswd.</p>
VPN , tunnels, iptables	<p>Définitions : DMZ, coupe-feux, proxy. VPN et tunnels. Principe de fonctionnement. Présentation des tunnels chiffrés. Travaux pratiques : mise en œuvre de stunnel pour sécuriser une messagerie smtp. Présentation d'openVPN. Travaux pratiques : installation, configuration, tests de connexion, création d'un tunnel sécurisé par clé statique. Certificats : SERV et CLT. Pare-feux : les iptables, le filtrage de paquets, définition d'une politique de sécurité. Travaux pratiques : mise en place des iptables. Traduction d'adresse, traduction de ports. Architecture avec pare-feux et tunneling.</p>
Proxy Squid	<p>Présentation, principe de fonctionnement. Architecture, hiérarchie de serveurs cache. Exemple d'utilisation, systèmes d'exploitation concernés, logiciels complémentaires. Mécanismes de configuration manuelle, automatique. Scripts d'auto-configuration, filtrage suivant DNS, par protocole. Clients en mode texte ,robots. Installation dans le navigateur. Principe et syntaxe des ACL. Optimisation de l'utilisation du serveur. Restriction d'accès par hôte, par réseau, par plage horaire, par jour, par site. Mise en cache des données. Méthodes d'authentification.</p>

Linux : optimisation performances métrologie

Durée: 2 jours
1000 €

du 10 au 11 Avril

du 20 au 21 Septembre

Public:

Administrateurs, et toute personne souhaitant connaître les éléments permettant d'améliorer les performances d'un système Linux.

Objectifs:

Connaître les points du système à mesurer. Comprendre leur impact sur les performances globales du système et savoir les adapter à un mode de fonctionnement (client, serveur, station, base de données, messagerie, ...)

Connaissances préalables nécessaires:

Une bonne connaissance d'un système Linux est nécessaire. Des notions d'administration sont souhaitées.

Programme:

Introduction	Qu'est ce que la gestion des performances?
Mesures	Les éléments à prendre en compte, les points de mesures. TP : création d'un utilitaire d'extraction des informations système.
Systèmes de fichiers	Les différents types de systèmes de fichiers. Les systèmes natifs: ext2, ext3, xfs, Gestion de la fragmentation, pagination Les systèmes émulés: vfat, ntfs Les systèmes distribués: nfs, smb Options: rsize, wsize, timeo, retrans,... TP: outil de mesure des accès.
Processus	Les processus. Les threads. Utilisation de la commande vmstat TP: déclenchement d'une alarme en cas de surcharge système.
Utilisateurs	accounting, quotas, fichiers de logs.
Réseau	Utilisation de la commande netstat Exploitation des éléments statistiques produits
Outils	Utilisation des pseudo-processus /proc: stat, cpuinfo, ... Utilisation des processus système: kswapd, swpctl, syslogd

Administration avancée Linux

Durée: 5 jours
2080 €

du 5 au 9 Mars
du 21 au 25 Mai

du 17 au 21 Septembre
du 12 au 16 Novembre

Public:

Administrateurs, et toute personne souhaitant approfondir l'administration d'un système Linux.

Objectifs:

Savoir installer, administrer, faire évoluer une distribution. Ce cours a lieu sur Linux RedHat, et sur Debian pour la partie "apt". Il est essentiellement basé sur des travaux pratiques.

Connaissances préalables nécessaires:

Connaître les techniques d'administration d'un système unix ou linux.

Programme:

Distribution

Présentation : RedHat Package Manager.

Les distributions qui utilisent les rpm.

Fonctionnalités : sécurité, méta-données, gestion des dépendance.

Détails de la commande rpm, et de ses options.

Travaux pratiques :

mise en œuvre, installation, désinstallation, requêtes documentation.

Construction de RPMs : depuis les sources jusqu'au package.

Description des paquets DEB: fonctionnement apt, dpkg, dselect, debconf.

L'outil apt : principe, les répertoires apt, fichiers release.

Les commandes apt-get, apt-cache.

Les frontaux apt : apt-shell, aptitude, synaptic.

Travaux pratiques :

recherche d'informations sur un paquet, installation d'une mise à jour.

Démarrage/Installation

Analyse du mode de démarrage : grub, Anaconda

Le système kickstart.

Analyse d'une image initrd.

Travaux pratiques :

Modification d'un initrd, ajout de modules.

Création de média d'installation.

Boot sur un périphérique USB depuis un CD.

Systèmes de fichiers journalisés

Exemples de systèmes de fichiers journalisés.

Les types de journalisation.

XFS : fonctionnement, mise en œuvre, administration compatibilité NFS

Ext3, ext4 : caractéristiques et mise en œuvre.

Administration avancée Linux

LVM	<p>Logical Volume Manager. Présentation. Définitions : VFS, EVMS, Volumes physiques, groupes de volumes, volumes logiques, extension logique. Travaux pratiques : mise en place de partitions LVM. Formatage en xfs. Mode d'utilisation des LVM : les snapshots, le redimensionnement, la concaténation de groupes de volumes. Exercice : création de volumes physiques, de groupes de volumes, création de snapshot. Ajout d'un disque, sauvegarde d'une partition, redimensionnement.</p>
RAID	<p>Définitions. Les principaux types de RAID. Le RAID Logiciel sous Linux : présentation, outils d'administration. Travaux pratiques : utilisation des outils mdadm pour créer un système de fichiers RAID. Mise en évidence des reprises sur incidents : simulation de panne, synchronisation des données. Analyse des performances.</p>
Authentification en production	<p>Besoin de mécanismes d'authentification performants et fiables. pam: gestion des modules d'authentification. Principe de base. Travaux pratiques : configuration, mise en œuvre. Les modules : access, chroot, cracklib, etc ... Ldap : Lightweight Directory Access Protocol Les modèles, la conception d'une arborescence. Interface pam/ldap. Travaux pratiques : mise en œuvre avec Openldap et l'automonteur</p>
Performances	<p>Le besoin, les points à surveiller. Les points de mesures : utilisation CPU, occupation des disques, charge réseau, occupation mémoire, etc ... Commandes de suivi des ressources processeurs et mémoire: vmstat, top. Commandes de suivi des ressources réseaux: netstat, ntop, iptraf. Surveillance des ressources disques : df, lsof Gestion de la fragmentation, pagination. Travaux pratiques : analyse des informations de /proc/stat, /proc/cpuinfo et de l'accounting. Les outils : oprofile, dtstat, systat.</p>

Administration avancée Linux

Ressources	Les quotas disques : principe, mise en place. Travaux pratiques : déclaration des quotas dans le fichier <code>/etc/fstab</code> , activation des quotas, exemple de dépassement de limite d'espace disque autorisé.
Noyau	Compilation du noyau : présentation, les différentes phases. Travaux pratiques : téléchargement et décompression des sources configuratin avec <code>make</code> , recompilation.
Périphériques	Périphériques non standards. Installation de modules: <code>modprobe</code> , <code>insmod</code> . Le répertoire <code>hotplug</code> .

Administration Ubuntu Server

Durée: 5 jours
2045 €

du 27 Février au 2 Mars
du 4 au 8 Juin

du 24 au 28 Septembre
du 26 au 30 Novembre

Public:

Administrateurs, et toute personne souhaitant maîtriser l'installation, la configuration d'un système Linux Ubuntu Server

Objectifs:

Savoir installer, administrer un système Ubuntu Server. Savoir mettre en place et configurer les principaux services d'un Ubuntu Server aussi bien graphiquement que manuellement.

Connaissances préalables nécessaires:

Des connaissances de base des systèmes unix et/ou linux sont nécessaires, ainsi que du shell.

Programme:

Introduction	Linux et l'opensource : historique, caractéristiques de linux Historique Ubuntu et Debian La communauté Ubuntu, le positionnement de Canonical.
Installation	Les phases d'installation d'un système Linux. Options dans les chargeurs : grub. Les outils d'installation ubuntu. Gestion des packages , dépôts et archives. Description des paquets .deb, et format. Les commandes dpkg, dpkg-deb, dpkg_query, apt. Présentation des outils : aptitude et synaptic. Gestion des dépôts et du fichier sources.list. Travaux pratiques : dpkg : extraction des informations concernant un paquet, recherche d'un paquet, fabrication d'un paquet. Installation d'une application à partir des sources ou d'une archive.
Gestion des disques	Les types de systèmes de fichiers : ext2/3/4, reiserfs, xfs, msdos/vfat Montage des systèmes de fichiers (rôle du noyau, options, fichier fstab) Construction de systèmes de fichiers, contrôle d'intégrité. Travaux pratiques : vérification du filesystem avec la commande fsck.
Accès distant : OpenSSH	Les principes de base d'openSSH, de OpenSSL, et des clés. Utilisation pour une simple connexion à distance. Travaux pratiques : mise en place d'openSSH pour des transferts de fichiers. Génération de clés. Création et utilisation de tunnels.

Administration Ubuntu Server

LVM	<p>Principe. Organisation du Logical Volume Manager. Gestion du LVM : les volumes physiques et logiques, les snapshots, les partitions. Travaux pratiques : Création de volumes physiques, logiques, ajout d'un disque, export/import d'un disque. Combinaison de groupes de volumes.</p>
RAID	<p>Définitions : les différents niveaux de RAID. Description du raid logiciel sur Linux. Travaux pratiques : Configuration et utilisation des différents raids logiciels.</p>
Utilisateurs	<p>gestion des comptes utilisateurs , groupes, droits d'accès, politique d'accès Contrôle des connexions de root. Etude des fichiers /etc/passwd, /etc/group, /etc/shadow. Gestion des comptes utilisateurs : useradd, usermod, userdel, passwd, gestion des groupes : groupadd, groupdel, ajout d'utilisateurs, création d'administrateurs de groupes, droits d'accès, politique d'accès. Travaux pratiques : création d'utilisateurs et de groupes, puis vérification de cohérence avec la commande pwck. Contrôle des connexions de root : les objectifs et les méthodes. Travaux pratiques : utilisation de l'outil "john the ripper" pour la recherche de mots de passe. Introduction à PAM : Pluggable Authentication Modules.</p>
Processus	<p>Les processus. Les threads. Gestion des priorités. Utilisation des pseudo-processus /proc: stat, cpubinfo, ...</p>
Sauvegardes	<p>Outils sauvegarde/archivage/compression: gzip, zip, tar, dd, cpio Sauvegarde du système, création de CD de secours. Travaux pratiques : sauvegarde par cpio, archivage par tar Commandes de sauvegardes en réseau : rsync, clonezilla. Présentation de l'outil backuppc.</p>
Impressions	<p>Les services d'impression, démarrage/arrêt des services d'impression. Présentation de CUPS : Common Unix Printing system Définitions : classes d'imprimantes, classes implicites, destination, filtres, backends. Travaux pratiques : installation d'une imprimante, modification d'un pilote : utilisation de lpr, cups, printtool, system-config-printer.</p>
Programmation de tâches	<p>Le besoin, l'automatisation des tâches systèmes Exécution différée avec at. Programmation de tâches avec cron. Etude du fichier crontab.</p>

Administration Ubuntu Server

Réseau IP

Les objets à configurer :

les interfaces réseaux, les routes, le DNS.

Principe de la configuration dynamique ou statique.

Configuration, nommage/activation des interfaces réseau, drivers.

Etude des fichiers /etc/hosts, /etc/nsswitch, /etc/resolv.conf.

Travaux pratiques :

création d'une interface réseau, visualisation,

configuration de plusieurs adresses IP sur la même interface physique,

ajout d'une route, d'un hôte, d'un serveur DNS, et tests.

Les utilitaires ssh, clients windows (Putty, WinSCP).

Sécurisation avec netfilter et iptables : principes de base.

Exploitation

Configuration de syslog

Journaux : /var/log/messages

surveillance des logs avec logwatch

Vérification d'intégrité du système de fichiers avec AIDE

Virtualisation Linux

Durée: 2 jours
1020 € HT

du 9 au 10 Février
du 14 au 15 Mai

du 13 au 14 Septembre
du 22 au 23 Novembre

Public:

chefs de projet, administrateurs souhaitant choisir une solution de virtualisation.

Objectifs:

Connaître les différentes solutions de virtualisation sur linux, et leurs caractéristiques.

Connaissances préalables nécessaires:

Une bonne connaissance du système Unix/Linux est nécessaire.

Programme:

Introduction	Objectifs d'un système d'exploitation, gestion de ressources. Partager des ressources entre plusieurs applications, systèmes, ... Notion de virtualisation , quelle granularité (disques, système d'exploitation, machines physiques, ...) Historique : VM (Virtual Machine) , VMWare, UML, Xen , ..
Les différentes techniques possibles :	conteneurs d'application, noyaux secondaires, machines virtuelles, hyperviseur, virtualisation matérielle, ...
Xen	Présentation de l'architecture de virtualisation Xen. Compilation d'un noyau Xen. Gestion des domaines : Création d'un domaine, arrêt d'un domaine. Console d'administration.
VmWare	Installation du produit. Création d'une machine virtuelle. Installation du système d'exploitation. Lancement.
VirtualBox	Principe et caractéristiques du produit. Les différentes Editions. Configuration des machines virtuelles en XML.
Ixc	Linux Containers Objectifs du projet Isolation et contrôle des ressources. Mise en œuvre.
QEMU et kvm	Principe de QEMU et architecture Kernel Based Virtual Machine : positionnement par rapport aux autres systèmes de virtualisation , et par rapport à QEMU Travaux pratiques avec un noyau contenant les modules kvm
Administration avec libvirt	Présentation de l'API libvirt et des fonctionnalités apportées Travaux pratiques : administration de domaines Xen avec libvirt

Virtualisation avec KVM

Durée: 2 jours
1020 €

du 7 au 8 Février
du 26 au 27 Avril

du 12 au 13 Juillet
du 18 au 19 Octobre

Public:

Administrateurs, chefs de projet, et toute personne souhaitant mettre en œuvre la virtualisation avec kvm.

Objectifs:

Comprendre le principe de fonctionnement de kvm, savoir l'installer et l'administrer.

Connaissances préalables nécessaires:

Une bonne connaissance des systèmes d'exploitation est nécessaire.

Programme:

Introduction	<p>Les différentes techniques de virtualisation. Définitions : conteneurs, machines virtuelles, hyperviseurs, virtualisation matérielle. Présentation de kvm : Kernel-based Virtual Machine. Principe et architecture : module intégré dans le noyau Linux, base QEMU. Positionnement par rapport aux autres solutions de virtualisation. Prérequis matériels et logiciels.</p>
Présentation QEMU	<p>Deux modes de fonctionnement : code compilé pour un processeur, émulation d'une machine physique.</p>
Installation de kvm	<p>Deux configurations possibles : depuis un noyau Linux de version supérieure à 2.6.25 et contenant les modules kvm ou avec recompilation du noyau. Optimisation, gestion de la mémoire.</p>
Mise en œuvre	<p>Création, installation et exécution d'une image. Les images préconfigurées. Réalisation de snapshots. Configuration du réseau.</p>
Migration d'images	<p>Le besoin. Sauvegarde/chargement de machines virtuelles : à l'arrêt ou en fonctionnement Limites par rapport aux processeurs Prérequis. Mise en œuvre : la commande migrate Paramètres (bande passante) Migration vers un fichier : sauvegarde puis restauration</p>
Administration	<p>Les outils de gestion de machines virtuelles kvm : UVMM, virsh, virt-manager. Travaux pratiques avec libvirt</p>

Virtualisation avec KVM

Réseaux virtuels avec VDE

Présentation de VDE : Virtuel Distributed Ethernet
Installation, configuration : création des interfaces "tap"
Travaux pratiques :
connexion de deux réseaux via un routeur virtuel,
création des interfaces TAP et des switches VDE.
Configuration des switches avec Unixterm.

Virtualisation avec Xen

Durée: 2 jours
1020 €

du 9 au 10 Février
du 14 au 15 Mai

du 13 au 14 Septembre
du 22 au 23 Novembre

Public:

Toute personne souhaitant mettre en place la virtualisation avec Xen, administrateurs, exploitants.

Objectifs:

Comprendre les principes de la solution de virtualisation Xen. Savoir configurer et installer Xen.

Connaissances préalables nécessaires:

Une bonne connaissance de l'administration des systèmes d'exploitation est nécessaire.

Programme:

Xen Présentation	<p>Introduction aux solutions de virtualisation Xen. Notion d'hyperviseur. Les différents types d'hyperviseurs. La paravirtualisation. Prérequis matériel pour l'utilisation de Xen. Systèmes d'exploitation supportés. Présentation de l'architecture Xen : l'hyperviseur et les systèmes hôtes. Les fonctionnalités disponibles : migration d'un domaine, gestion des ressources, clustering. Contraintes de sécurité.</p>
Installation	<p>Travaux pratiques d'installation à partir des packages RPMs ou des binaires debian. Installation de xen et du noyau dom0. Personnalisation d'un noyau : compilation du dom0 pour refléter la configuration de la machine hôte. Configuration et démarrage. Paramétrage du lanceur : grub. Création d'un domaine. Arrêt d'un domaine.</p>
Domaines utilisateurs	<p>Gestion des systèmes invités : le service xend La commande d'administration xm (xm create, xm liste, ...) l'accès à xend par l'interface web. Travaux pratiques : ajout de systèmes invités avec xm create. Utilisation d'un fichier de description de machines virtuelles. Configuration de domU. Arrêt et démarrage de systèmes invités avec xm shutdown et xm reboot.</p>
Supports de stockage	<p>Déclaration des espaces de stockage accessibles au domU : périphériques blocs, partitions physiques, ou volumes logiques.</p>

Virtualisation avec Xen

Systemes paravirtualisés	Etude des paramètres de démarrage des domU. Travaux pratiques : création manuelle d'images. Utilisatoin de debootstrap, et de rpmstrap.
Administration	Les outils d'administration : Xend, Xm Etude détaillé de la commande xm. Mise en œuvre de la console d'administration : configuration de domaines, du réseau.
Mise en production	La gestion des LVM,des processeurs Sauvegarde et restauration de domaines Gestion des ressources : CPu, mémoire, réseau et stockage

Virtualisation postes de travail

Durée: 2 jours
1020 €

du 22 au 23 Février
du 18 au 19 Avril
du 13 au 14 Juin

du 5 au 6 Septembre
du 21 au 22 Novembre

Public:

Chefs de projet, architectes, administrateurs

Objectifs:

Connaître les différentes solutions de virtualisation des postes de travail, et leurs caractéristiques.

Connaissances préalables nécessaires:

Aucune connaissance préalable n'est nécessaire.

Programme:

Introduction

Objectifs d'un système d'exploitation, gestion de ressources.
Partager des ressources entre plusieurs applications, systèmes, ...
Notion de virtualisation , quelle granularité (disques, système d'exploitation, machines physiques, ...)
Historique : VM (Virtual Machine) , VMWare, UML, Xen , ...
Utilisation de machines virtuelles sur un poste de travail, VDI (Virtual Desktop Infrastructure).

Apports

Centralisation des postes de travail.
Cohabitation de plusieurs systèmes d'exploitation, environnement de tests.
Simplification des architectures, contrôle du parc.

Accès distants

Déport d'écrans avec virtualisation centralisée.
Présentation de NX, vmware view/thin app, vnc, RDP/rdesktop, dotRiver, remoteFX, LTSP.
Comparaison des différentes approches.

Accès Web

Solutions applicatives hébergées: officeweb, ajax, flash, JEE, Silverlight

Virtualisation locale

Environnement multiples:
présentation de vmware workstation, Virtual Box, VirtualPC, Xen, qemu.
Evaluation des apports de chaque solution en fonction des besoins.

Virtualisation serveurs

Durée: 2 jours
1020 €

du 20 au 21 Février
du 16 au 17 Avril
du 11 au 12 Juin

du 3 au 4 Septembre
du 19 au 20 Novembre

Public:

Chefs de projet, architectes, administrateurs

Objectifs:

Connaître les différentes solutions de virtualisation de serveurs. Comprendre les enjeux de l'administration centralisée et les solutions de gestion du marché.

Connaissances préalables nécessaires:

Aucune connaissance préalable n'est nécessaire.

Programme:

Technologies

Objectifs d'un système d'exploitation, gestion de ressources.
Partager des ressources entre plusieurs applications, systèmes, ...
Notion de virtualisation , quelle granularité (disques, système d'exploitation, machines physiques, ...)
Historique : VM (Virtual Machine) , VMWare, qemu, KVM, Xen , ..
Fonctionnement d'une machine virtuelle.
Description des ordonnanceurs.
Architecture.
Spécificités de la virtualisation de serveurs.
Nécessité d'un noyau autonome.

Apports

Mutualisation des ressources et de l'administration.
Regroupement géographique des centres serveurs, isolation applicative, création d'environnements de tests.

Solutions

Présentation des différentes solutions et leurs principes et architectures, points forts et points faibles:
xVM Server et xVM Ops Center sur Sun, micro-partitionnement avec DLpar sur AIX, et VMWare, Xen, KVM en tant que socle de virtualisation, OpenVZ, Vserver pour l'isolation applicative.
Comparaison avec les solutions fondées sur un système d'exploitation applicatifs comme VirtualBox ou VirtualPC.

Administration

Problèmes posés par la virtualisation de grappes.
Gestion du volume.
Gestion des erreurs.
Sauvegarde et restauration.
Outils: ganeti, virtualmin, Virtual Machine Manager, convirt.
Sécurité.
Les moyens employés par les différents produits: console d'administration, HMC, ...
Repositionnement des compétences: système, réseau et stockage.

Virtualisation serveurs

Socles de virtualisation

Xen: Présentation de l'architecture de virtualisation Xen.
Compilation d'un noyau Xen.
Création d'un domaine.
Arrêt d'un domaine.
Console d'administration.
Introduction à virsh, xm.
VmWare ESx: Présentation de la gamme.
Caractéristiques de ESX.
Comparaison avec Xen.
Installation du produit. Création d'une machine virtuelle.
Installation du système d'exploitation.
Micro partitionnement sur AIX et KVM/qemu sur Linux: mise en œuvre.
Administration des socles.
Administration des machines virtuelles

Cloisonnement

Approche VServer, openVZ sous Linux et zones Solaris: partitionnement des ressources.
Comparaison par rapport à une virtualisation complète.
Organisation de l'administration et de la sécurité.

Préparation certification LPI101

Durée: 2 jours
845 €

du 12 au 13 Mars

du 10 au 11 Septembre

Public:

Administrateurs Linux ayant une expérience pratique de l'administration, ou ayant suivi les formations sur les bases d'administration Linux.

Objectifs:

Préparer les participants au passage de l'examen LPI101 permettant d'obtenir la certification LPI de niveau 1 (LPIC-1). Cette certification couvre les compétences de base d'un professionnel de Linux qui sont communes aux principales distributions Linux.

Connaissances préalables nécessaires:

Connaissance de l'architecture des systèmes Linux, des commandes de bases Unix, Linux, des procédures d'installation, des systèmes de fichiers.

Programme:

Architecture système	Déterminer et configurer les paramètres du matériel. Démarrage du système. Changer de niveau d'exécution et arrêter ou redémarrer le système.
Installation de Linux et gestion de paquetages	Concevoir un schéma de partitionnement du disque dur Installation d'un gestionnaire de démarrage Gestion des bibliothèques partagées Utilisation du gestionnaire de paquetages Debian et des gestionnaires de paquetages RPM et YUM
Commandes GNU et Unix	Travail en ligne de commande Traitement de flux de type texte par des filtres Effectuer une gestion de base sur les fichiers Utilisation des flux, des tubes (pipes) et des redirections Création, surveillance et destruction de processus Modification des priorités des processus Recherche dans des fichiers texte avec les expressions rationnelles Edition de fichiers texte avec vi
Disques	Système de fichiers Linux, Arborescence de fichiers standard. Création de partitions et systèmes de fichiers Maintenir l'intégrité des systèmes de fichiers Contrôle du montage et du démontage des systèmes de fichiers Gestion des quotas de disque Gérer les permissions et les propriétaires des fichiers Créer et changer les liens symboliques et physiques sur les fichiers Recherche de fichiers et placement des fichiers aux endroits adéquats

Préparation certification LPI102

Durée: 2 jours
845 €

du 14 au 15 Mars

du 12 au 13 Septembre

Public:

Administrateurs Linux ayant une expérience pratique de l'administration, ou ayant suivi les formations sur les bases d'administration Linux.

Objectifs:

Préparer les participants au passage de l'examen LPI102 permettant d'obtenir la certification LPI de niveau 1 (LPIC-1). Cette certification couvre les compétences de base d'un professionnel de Linux qui sont communes aux principales distributions Linux.

Connaissances préalables nécessaires:

Connaissance du shell, de la programmation de scripts, de l'interface graphique des systèmes Linux, des tâches de base d'administration, des notions sur la configuration du réseau.

Programme:

Shells, scripts et gestion de données	<ul style="list-style-type: none"> Personnalisation et utilisation de l'environnement du shell Personnalisation ou écriture des scripts simples Gestion de données avec le langage SQL
Interface utilisateur et environnement graphique	<ul style="list-style-type: none"> Installation et configuration de X11 Configurer un gestionnaire d'affichage (Display Manager) Accessibilité
Tâches d'administration	<ul style="list-style-type: none"> Gestion des comptes utilisateurs et des groupes ainsi que des fichiers systèmes concernés Automatisation des tâches d'administration par la planification Paramètres régionaux et langues
Services système essentiels	<ul style="list-style-type: none"> Gestion de l'horloge système Journaux systèmes Configuration de base de l'agent de transfert de courrier (MTA) Gestion de l'impression et des imprimantes
Notions élémentaires sur les réseaux	<ul style="list-style-type: none"> Notions élémentaires sur les protocoles Internet Configuration réseau de base Résolution de problèmes réseau Configuration DNS sur les postes de travail
Sécurité	<ul style="list-style-type: none"> Tâches d'administration de sécurité Configuration de la sécurité du système Sécurisation des données avec le chiffrement

Préparation certification LPI201

Durée: 2 jours
845 €

du 19 au 20 Mars

du 24 au 25 Septembre

Public:

Administrateurs Linux ayant une expérience pratique de l'administration, ou ayant suivi les formations sur les bases d'administration Linux et sur les services réseau sous Linux.

Objectifs:

Préparer les participants au passage de l'examen LPI201 permettant d'obtenir la certification LPI de niveau 2 (LPIC-2). Cette certification couvre les compétences de base d'un professionnel de Linux qui sont communes aux principales distributions Linux.

Connaissances préalables nécessaires:

Disposer d'une certification LPIC-1. Savoir compiler le noyau Linux, personnaliser et maintenir le système, configurer le réseau et les serveurs de noms, sécuriser un réseau hétérogène (Windows/Linux)

Programme:

Noyau Linux	Composants du noyau. Compilation, mise à jour d'un noyau. Personnalisation, construction et installation d'un noyau et des modules. Gestion/interrogation des modules.
Démarrage du système	Personnalisation du démarrage système et séquence de boot. Restauration du système.
Systèmes de fichiers et périphériques	Intervention sur le système de fichiers Linux. Maintenance du système de fichiers. Création et configuration des options du système de fichiers. Gestion de périphérique udev.
Administration avancée de périphérique de sauvegarde	Configuration RAID. Ajustement des accès au périphérique de sauvegarde. Gestionnaire de volumes logiques.
Configuration réseau	Configuration élémentaire, et avancée d'un réseau et dépannage. Identification et correction des problèmes réseau. Notification des utilisateurs concernant les problèmes relatifs au système.
Maintenance système	Construction et installation de programmes à partir du code source. Opérations de sauvegarde.
Serveur de nom de domaine	Configuration élémentaire d'un DNS. Création et maintenance de zones DNS. Sécurisation d'un serveur DNS.

Préparation certification LPI202

Durée: 2 jours
845 €

du 21 au 22 Mars

du 26 au 27 Septembre

Public:

Administrateurs Linux ayant une expérience pratique de l'administration, ou ayant suivi les formations sur les bases d'administration Linux et sur les services réseau sous Linux.

Objectifs:

Préparer les participants au passage de l'examen LPI201 permettant d'obtenir la certification LPI de niveau 2 (LPIC-2). Cette certification couvre les compétences de base d'un professionnel de Linux qui sont communes aux principales distributions Linux.

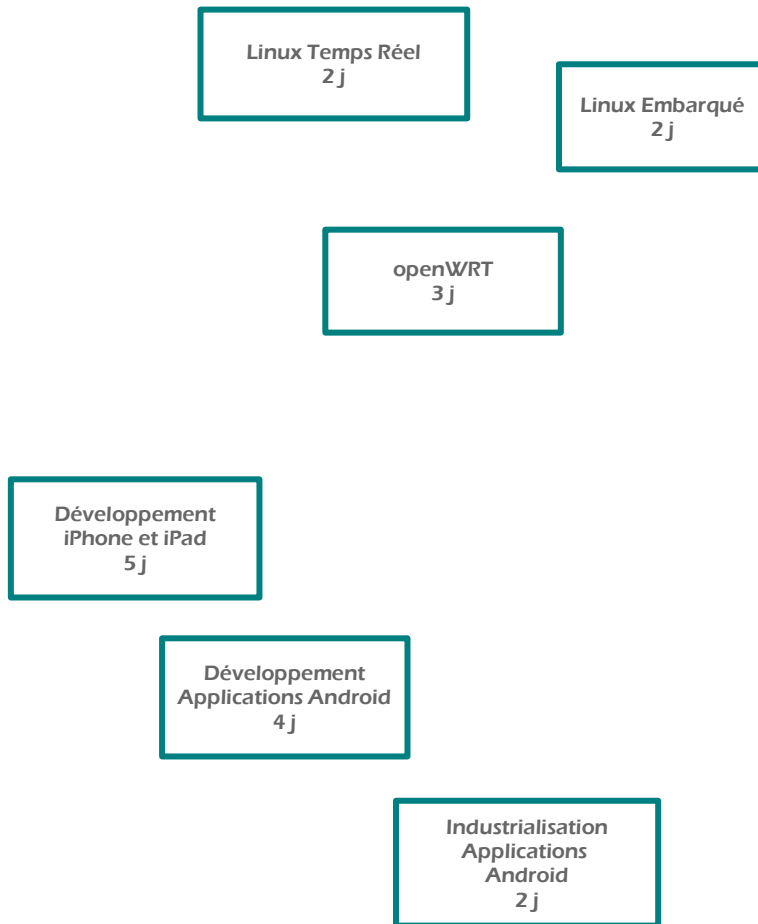
Connaissances préalables nécessaires:

Disposer d'une certification LPIC-1. Savoir administrer les services web, gérer le partage de fichiers, les clients réseau, les services de mail, la sécurité du système, et effectuer les dépannages.

Programme:

Services web	Mise en place et maintenance d'un serveur web. Mise en place d'un serveur mandataire.
Partage de fichiers.	Configuration du serveur Samba, d'un serveur NFS.
Gestion des clients réseau	Configuration DHCP. Authentification PAM. Usage de client LDAP.
Services e-mail	Utilisation des serveurs e-mail. Gestion de la livraison locale et distante des e-mail.
Sécurité système	Configuration d'un routeur. Sécurisation des serveurs FTP. Shell sécurisé (SSH). TCP Wrapper. Tâches de sécurité.
Dépannage	Identification des étapes de démarrage et dépannage des chargeurs de démarrage. Dépannage général. Dépannage des ressources système, et des configurations liées à l'environnement.

Filières Systèmes embarqués et mobilité



Linux temps réel

Durée: 2 jours

Prix et dates: nous consulter

Public:

Toute personne intéressée par les systèmes temps réel sur Linux

Objectifs:

Connaître les principes de fonctionnement et savoir installer un système Linux temps réel.

Connaissances préalables nécessaires:

Bonnes connaissances du système Linux.

Programme:

Introduction	Définition d'un système temps réel. Les services offerts. Solutions Temps Réel sur Linux. Architectures et caractéristiques. Critères de choix.
Définitions	Interruptions. Horloge, Entrées/Sorties, automate, séquenceur. Contexte d'exécution. Interruptions d'horloge. Alarmes. Timer cycliques et non cycliques. Automate à états. Travaux pratiques : mise en place d'un mini-moniteur réparti.
Ordonnancement	Notion de processus, de ressource, de thread. Analyse et découpage des tâches dans RT Linux. Modification de l'ordonnanceur. Concurrence et coopération. Stratégies d'ordonnancement Impact sur la gestion de contexte. Mesure de la charge. Travaux pratiques: Mise en évidence du fonctionnement de l'ordonnanceur de RT Linux
Communication inter-processus	Communication synchrone entre tâches, verrous. Section critique. Sémaphores. Communications entre le noyau RT et le noyau Linux. Utilisation de mémoire partagée ou de canaux fifos.
Entrées/Sorties	Mappées, accès direct. Pré traitement d'E/S. Alarmes sur seuil. Accès direct aux entrées/sorties de bas niveau. E/S série Temps réel. Structure des pilotes de périphériques. Travaux pratiques: mise en place d'une communication inter-processus sur lien série
Interruptions	Emulation des interruptions Linux. Partage des interruptions
Horloges	Horloges matérielles et logicielles. Alarme POSIX.

Systèmes Linux embarqués

Durée: 2 jours
1030 € HT

du 23 au 24 Février
du 2 au 3 Mai

du 10 au 11 Juillet
du 18 au 19 Octobre

Public:

Tout développeur ou ingénieur système souhaitant mettre en œuvre des systèmes Linux embarqués.

Objectifs:

Connaître les principes des systèmes embarqués Linux et les différentes solutions disponibles, savoir mettre en œuvre un système Linux embarqué.

Connaissances préalables nécessaires:

Bonnes connaissances des systèmes Linux.

Programme:

Les systèmes embarqués	Définitions. Contraintes. L'offre du marché. Les distributions Linux embarquées openWRT, emDebian, openEmbedded, Angstrom, openMoko. Architectures et caractéristiques. Critères de choix: coût.
Construction	Les différentes étapes pour construire un système embarqué. Installation d'une chaîne de compilation croisée. Chargeur, noyau et système racine. Les outils disponibles: buildroot, openWRT, plugins eclipse, ... Plateformes de test : qemu, cartes de développement ARM (6410, Zoom/OMAP, WRT54)
Matériel	Processeurs. Mémoire Nand/Nor. Stockage SD. Ports USB. Port JTag. Travaux pratiques : utilisation du bus JTag.
Chargeur	Présentation. Travaux pratiques : compilation d'un chargeur U-Boot. Mise en place sur une carte de développement. Découpage de la Nand en partitions. mtd.
Noyau	Compilation d'un noyau. Travaux pratiques : compilation d'un noyau pour carte 6410 (ARMv11). Optimisation de l'empreinte mémoire. Gestion de l'initrd.
Système racine	Organisation du système de fichiers: linux, rootfs, nvram, rootfs_data. Systèmes de fichiers embarqués. Présentation de jffs2, yaffs2. Autres types de systèmes de fichiers. Services embarqués: dropbear, dnsmasq, busybox. Compilation d'une busybox. Présentation, ajouts de fonctionnalités, limitations.

Systèmes Linux embarqués

Développement

Ajout de modules au noyau.

Travaux pratiques : Portage de netsnmp.

Compilation d'une MIB spécifique au matériel et intégration dans la distribution embarquée.

Mise en œuvre dans un environnement multi-plateformes.

Centralisation des journaux syslog. Supervision par Nagios.

Travaux pratiques : Portage de haproxy. Mise en œuvre sur de la répartition de charge HTTP. Frontal à un cluster JBoss.

Visualisation des performances.

Durée: 3 jours

Prix et dates: nous consulter

Public:

Intégrateurs de systèmes openWRT.

Objectifs:

Connaître les principes de fonctionnement et savoir installer un système openWrt Linux. Les travaux pratiques seront réalisés sur des plateformes WRT54 et NSLU2.

Connaissances préalables nécessaires:

Bonnes connaissances du système Linux.

Programme:

Embarqué	Définitions. Contraintes. L'offre du marché. Les distributions Linux embarquées DD-WRT, openWRT. Architectures et caractéristiques. Critères de choix: coût.
openWRT	Présentation. Installation d'une chaîne de compilation croisée. Flashage du micro logiciel. Système de récupération. Introduction à mtd, uci et nvram. Buzybox. Présentation, ajouts de fonctionnalités, limitations.
Configuration	Fichiers de configuration. Paramètres système. Gestion du temps: ntp. Services réseau: dropbear, dnsmasq. Gestion des services dhcp et sshd. Gestion des clés ssh. Modification de la configuration avec uci. Mode maintenance. Récupération après erreur de configuration.
Réseau	Gestion du réseau et des interfaces. Configuration réseau avec une adresse par port. Présentation du coupe-feux. Fichiers de configuration. Gestion du Wifi.
Partitions	Organisation du système de fichiers: linux, rootfs, nvram, rootfs_data. Gestion des partitions jffs2. Autres types de systèmes de fichiers. Le CFE: accès, gestion, configuration. L'utilitaire nvserial.
Applications	Mise en place d'un dépôt de paquets. TP: Portage de netsnmp. Compilation d'une MIB spécifique au matériel et intégration dans la distribution embarquée. Mise en œuvre dans un environnement multi-plateformes. Centralisation des journaux syslog. Supervision par Nagios.
Développement	Création d'un paquet applicatif. Ajout d'un paquet à partir des sources. Portage d'un paquet "lourd" sur openWrt. TP: écriture d'un mini-serveur HTTP embarqué.

openWrt

Déploiement

TP: Portage de haproxy. Mise en œuvre sur de la répartition de charge HTTP.
Frontal à un cluster JBoss. Visualisation des performances.

Développement d'applications Android

Durée: 4 jours
1750 €

du 27 Février au 1er Mars
du 23 au 26 Avril
du 2 au 5 Juillet

du 22 au 25 Octobre
du 3 au 6 Décembre

Public:

Développeurs et intégrateurs d'applications sous Android.

Objectifs:

Connaître les principes de fonctionnement et savoir développer des applications sous Android

Connaissances préalables nécessaires:

Connaissance d'un langage de programmation objet.

Programme:

Introduction	Présentation du système d'exploitation android v1, v2 et v3. Architecture. La couche noyau Linux. Bibliothèques de base. Applicatifs java.
Applicatif	Les principales applications existantes. Installation/désinstallation d'une application. Accès distant à la mémoire flash. Arborescence des fichiers.
Développement	Présentation du SDK, installation. Utilitaires: émulateurs, simulateur de carte. Développement d'une application de base. Transfert sur un matériel physique. Présentation des appels en call-back. Structure générale des applications. Les quatre modèles d'applications: Activity, Services, Broadcast receivers, Content receivers. Cycle de vie des composants.
Interface utilisateur	Définitions. Présentation des layout. Récupération du contexte applicatif. Gestion des menus. Boîtes de dialogue. Thèmes. Notifications (Toast, Status Bar, Dialog). Ecriture d'une application calculatrice.
Graphique et multimédia	Développement 2D et 3D. Les APIs. Utilisation du MediaPlayer. Visualisation de vidéos. Enregistrement de sons. Mise en œuvre du MediaRecorder.
API	Accès réseau, accès au système de fichiers. Mise en œuvre des APIs réseau. Capteurs internes. Ecriture d'un code exploitant les capteurs de position.

Développement d'applications Android

Interactions d'applications

Principe de réutilisation d'extraits applicatifs.
Ecriture d'une application exploitant GoogleMaps.
Déclaration des possibilités applicatives: les intents filters.

Industrialisation Android

Durée: 2 jours
930 €

du 29 au 30 Mars
du 14 au 15 Juin

du 11 au 12 Octobre
du 20 au 21 Décembre

Public:

Développeurs, chefs de projet souhaitant déployer des applications android à grande échelle, et gérer des flottes de mobiles android.

Objectifs:

Connaître les techniques de gestion centralisée de mobiles android, de télédistribution.

Connaissances préalables nécessaires:

Connaissance du développement d'applications android.

Programme:

Présentation	Rappels sur l'architecture android : framework, library, runtime, kernel. Structure du système. L'outil adb : fonctionnalités, l'accès au système avec le shell.
Compilation	Gestion centralisée des applications Scripts de compilation avec ant. Compilation en masse. Gestion des droits. Compilation native : le besoin, accès direct au système, ou depuis une application java avec JNI. Accès aux informations systèmes en mode pull. Décompilation, protection de code.
Intégration	Gestion des certificats. Types de signatures. Gestion des clés: clés internes par défaut, création de clés personnalisées. Scripts signataires.
Déploiement	Création d'une plateforme privée de type "AndroidMarket". Télédistribution, mode push, mode pull. Gestion de flotte. Centralisation des informations systèmes des mobiles. Configuration réseau : support d'IPv6.
Système	Recompilation locale d'Android à partir des sources. Présentation de l'interaction Linux/Java. La JVM embarquée: Dalvik VM. Le principe des classes.dex

Developpement pour iPhone et iPad

Durée: 5 jours
2150 € HT

du 13 au 17 Février
du 2 au 6 Avril

du 17 au 21 Septembre
du 19 au 23 Novembre

Public:

Développeurs sur mobiles iPhone

Objectifs:

Connaître les possibilités du développement sur les devices iOS (iPhone, iPod Touch, iPad), maîtriser le langage Objective-C, l'architecture Cocoa-Touch et les APIs disponibles.

Connaissances préalables nécessaires:

Expérience de la programmation, de préférence orientée objet. Une connaissance minimale du langage C est requise.

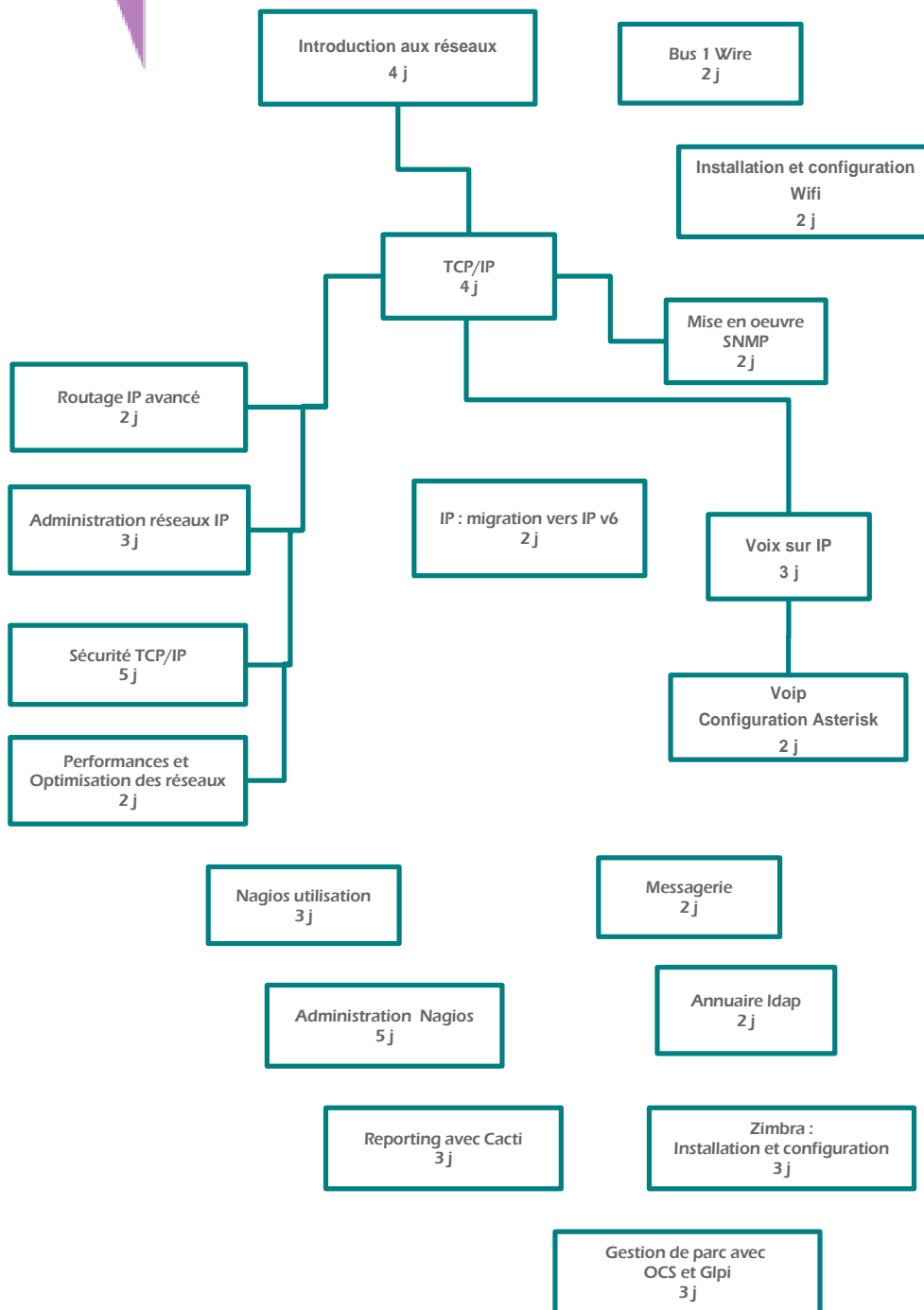
Programme:

Première application	Création de l'interface utilisateur Connexion des objets d'interface Notions d'objets et de messages Installation de l'application dans le simulateur d'iPhone Installation dans l'iPhone
Objective-C	La programmation orientée objet Caractéristiques des langages statiques et dynamiques Présentation du langage Objective-C
Outils de développement	Les grandes phases du projet : architecture, condng, documentation, testing, debugging, optimisation, publication Présentation d'Xcode et des outils utilisés pour chaque phase. Présentation de l'App Store et des modalités de publication.
Texte, délégation	Création d'un éditeur de texte. Retour sur la notion de délégation. Utilisation des notifications. Enrichissement des objets d'interface.
Architecture des objets	Le modèle MVC (model, view, controller) Présentation des ViewControllers standards Utilisation des transitions
Les TableViews	Création d'une application basée sur une TableView. Ajout d'une vue de détails. Utilisation des familles de polices.
Géolocalisation	Présentation des outils : GPS, triangulation, magnétomètre. Création d'une application utilisant la géolocalisation. Ajout d'une carte dynamique.
Application multilingue	Préparation d'une application pour l'international. Gestion du texte, des ressources. Utilisation des formats locaux (nombres, dates, monnaies, calendriers)

Developpement pour iPhone et iPad

Gestion des images	Utilisation de l'appareil photo et de la caméra. Stockage des images dans la bibliothèque. Stockage sous forme de fichier JPG ou PNG.
Le Carnet d'adresses	Création d'une application utilisant le carnet d'adresses Retour sur les ViewControllers et les vues de détail.
La WebView	Création d'un navigateur web. Utilisation des fonctions standards de la WebView.
Son et vidéo	Présentation des différentes API. Lecture d'un son court, d'un fichier audio Enregistrement audio. Lecture vidéo en plein écran, et dans une SubView
L'accéléromètre	Création d'une application utilisant l'accéléromètre.
Core Graphics	Présentation. Examen de l'API. Exemples de dessins en 2D.
Core Animation	Présentation des différentes API d'animation. Les transitions. Déplacement des objets. Transformations 2D (changement de taille, ..), et 3D (rotation dans l'espace, mise en perspective, ...)
Utilisation de la documentation	Présentation de la documentation en ligne, intégrée. Interaction avec le code source dans Xcode. Navigation rapide dans la documentation. Mise à jour. Outils tierces parties. Livres et références, sites utiles.

Filières Réseaux et TCP/IP



Introduction aux réseaux

Durée: 4 jours
1860 €

du 16 au 19 Janvier
du 12 au 15 Mars
du 29 Mai au 1er Juin

du 27 au 30 Août
du 15 au 18 Octobre
du 10 au 13 Décembre

Public:

Toute personne souhaitant acquérir des connaissances générales sur les réseaux, et plus particulièrement sur la mise en œuvre d'un réseau et les outils nécessaires à son exploitation.

Objectifs:

Comprendre les composants fonctionnels d'un réseau informatique. Analyser les possibilités d'interconnexion entre les différents réseaux. Connaître l'état de l'art de la conception, de la gestion et du suivi de réseaux hétérogènes.

Connaissances préalables nécessaires:

Aucune connaissance préalable n'est requise pour suivre ce cours

Programme:

Introduction	Le besoin de communication Quelques définitions. Les couches ISO. Normalisation et standards (ISO, IEEE, IETF, ATM Forum, ...)
Architectures de base	Topologies filaires, topologies sans fils Réseau maillé, Doublement de lignes, sécurisation
Supports physiques	Evolutions technologiques et mutation des réseaux. Acteurs du marché: opérateurs, fournisseurs, intégrateurs, distributeurs. Câblage : topologies et architectures. Usage des locaux techniques. Brassage. Radio : le besoin, les limites, l'état du marché.
Transmissions	Pourquoi et comment transmettre les informations ? Des transmissions série, parallèle ou hertzienne aux protocoles. Concepts de base et terminologie. Composants des réseaux (produits CISCO, 3COM, ...).
Technologies	Présentation rapide Ethernet, Giga Ethernet, Token-Ring, FDDI, Frame Relay, RNIS, ATM Les Ethernet: du 10M au 10G. Les normes 802.3ab et 802.3ae
Réseaux sans fils Wlan	HiperLAN IEEE 802.11
Normes Wifi	Présentation Points forts, points faibles Architecture des réseaux Wifi : 802.11, exemple d'ESS, le monde ad hoc, OLSR le matériel, interopérabilité
Utilisation du Wifi	Points d'accès Modes de fonctionnement, mode répéteur, Mode pont Alignement d'antennes, Supervision de réseaux

Introduction aux réseaux

TCP/IP	Définitions, adressage Exemple d'application Le protocole IP, la trame IP, TCP, UDP
Outils réseau	Outils de trace, tcpdump, outils de diagnostic actifs/passifs, analyseurs de flux, ...
Interconnexion de réseau et routage	Technologies, commutation Routage IP Fragmentation , VLAN Outils de gestion du routage Plan d'adressage OoS
IPV6	Besoin, fonctionnalité La trame IPV6, adressage
Sécurisation	VPN et tunnels: Objectif, fonctionnement DMZ et Pare-feux: Définition, serveur Proxy, fonctionnement pare-feux et tunneling Filtrage: les iptables, politique par défaut, état des connexions, traduction d'adresses, traduction de ports, connexion à internet
Voix sur IP	Commutation de paquets Avantages de la voix sur IP Les protocoles : H323, SIP Introduction RTP : définition et applications, RTP et Nat Utilisation du registrar SIP avec Asterisk Création des comptes téléphones, du dialplan, vérification et tests Enregistrements SRV : serveurs DNS et Asterisk Transport de données Bande passante et qualité de service (QoS)
Evolutions	L'adressage IP, la sécurité, les réseaux de stockage.

TCP/IP : protocoles et mise en œuvre

Durée: 4 jours
1620 €

du 13 au 16 Février
du 10 au 13 Avril
du 11 au 14 Juin

du 8 au 11 Octobre
du 3 au 6 Décembre

Public:

Toute personne souhaitant mettre en œuvre TCP/IP et les outils nécessaires à son exploitation.

Objectifs:

Maîtrisez les fonctionnalités du protocole TCP/IP, sa position par rapport aux autres protocoles. Savoir configurer un routeur et les différents composants d'un réseau local. Savoir mettre en œuvre les aspects fonctionnels et les services applicatifs.

Connaissances préalables nécessaires:

Connaissances de base sur les réseaux et les systèmes d'exploitation.

Programme:

Introduction	Définitions : IP, TCP. Historique. IP dans le modèle ISO.
IP	Le protocole IP: trame, adressage, principes de routage. Configuration des adresses et des masques réseaux. Accès à la couche réseau sur différents systèmes d'exploitation. Configuration de l'interface réseau.
Routage	Interconnexion de réseaux, répéteurs, les ponts. La commutation. Routeurs et passerelles. Définition d'une topologie. Principe de routage, algorithmes. Configuration des routeurs et des postes clients. Visualisation des chemins utilisés via traceroute. Routage dynamique: RIP, OSPF.
TCP/UDP	Les protocoles UDP/TCP: mode non connecté/connecté. Connexion virtuelle. Les ports TCP bien-connus (well known ports)
Applications	Les services du niveau application: telnet, ftp, ssh, scp, traceroute, ping (connexion, transfert de fichiers, contrôle), modèle client-serveur. Serveurs de noms : DNS (Domain Name System). Principe, traitement des requêtes. Mise en œuvre : configuration client, serveur, accès inversé. SNMP (Simple Network Management Protocol) : fonctionnalités, apports SNMP V2.
IPv6	Adressage actuel, attribution des adresses. Le travail de l'IETF (BradnerMankin). Plan d'adressage sur 128bits. Agrégateurs : découpage TLA/NLA/SLA/IID. Intégration des Regional Registries Fonctionnement : Surcharge d'entêtes. Structures des trames. Les nouveaux mécanismes: fragmentation: MTU universelle, DHCPv6, dynamic DNS, renumérotation simplifiée d'un plan d'adressage
Sécurité	Ipssec (IP Security Protocol) TP de mise en œuvre

Administration réseaux IP

Durée: 3 jours
1410 €

du 27 au 29 Février
du 2 au 4 Mai

du 3 au 5 Septembre
du 10 au 12 Décembre

Public:

Les administrateurs réseaux et tout exploitant de réseaux TCP/IP.

Objectifs:

Savoir configurer, tester, surveiller un réseau TCP/IP. Gérer les montées de version.

Connaissances préalables nécessaires:

Notions sur le câblage, transmissions et les bases de TCP/IP.

Programme:

Adressage IP	Utilisation de l'outil "ping" pour vérifier l'existence d'une machine. Création d'un outil de recherche automatique des machines actives sur le réseau.
RIP	Routage IP dynamique. Mise en œuvre de RIP sur différents routeurs, visualisation par traceroute.
DNS	Définition.Fonctionnement. Configuration. Mise en place de plusieurs serveurs DNS primaire et secondaires. Interconnexion.
DHCP	Mise en place d'un serveur DHCP: partage d'une plage d'adresses. Activation de la RFC 1542.
NTP	Définition.Fonctionnement. Déclaration d'un point de synchronisation. Configuration d'un serveur d'horloge. Configuration de clients. Architecture. Contrôles d'accès.
bootp, tftp	Configuration à distance Surveillance du réseau
Analyseurs de trames	Mise en place d'un outil de surveillance des paquets transitant sur un réseau (tcpdump, sniffit).
MRTG	Surveillance de flux, de charge.
SNMP	Rappels du principe de fonctionnement. Principe des MIB. Surveillance des différentes ressources d'un poste. Exécution de processus distants.
Extensions SNMP	ASN-1, modification de MIB
Administration Web	webmanagement avec Webmin. Installation/mise en route. Présentation de l'interface. Configuration de clusters.
QoS	régulation du flux en fonction du protocole TP: mise en œuvre de CBO puis de HTB.

Routage IP avancé

Durée: 2 jours

Prix et dates: nous consulter

Public:

Les administrateurs réseaux et tout exploitant de réseaux TCP/IP.

Objectifs:

Savoir configurer un routeur et les différents composants d'un réseau local.

Connaissances préalables nécessaires:

Il est demandé aux participants de connaître les bases de TCP/IP.

Programme:

Introduction	Routage, partage, commutation Commutation de niveau 3 Pontage sous Linux. LARTC, ebtables, brctl
Rappels sur IP	Plan d'adressage IP notion de classes les masques de sous-réseaux
Routage	Définition d'une topologie. Principe de routage. Configuration des routeurs et des postes clients. Visualisation des chemins utilisés via traceroute.
Routage dynamique	RIP, OSPF TP: mise en évidence des transferts RIP TP: mise en œuvre d'OSPF. Présentation du routage des systèmes autonomes: IGRP BGP, BGP-4 Routage OSPF sous Linux et sous IOS Cisco.

IP: Migration vers IPv6

Durée: 2 jours
935 €

du 8 au 9 Mars
du 18 au 19 Juin

du 20 au 21 Septembre
du 15 au 16 Novembre

Public:

Toute personne souhaitant migrer vers l'adressage IPv6

Objectifs:

Connaître les caractéristiques d'IPv6, et savoir élaborer les méthodes de migration.

Connaissances préalables nécessaires:

Connaissances de bases sur les réseaux et les systèmes d'exploitation.

Programme:

Rappels sur la version 4

Le protocole IP: trame, adressage, principes de routage.
Problèmes d'IPv4.

IPv6

Structure des trames.
Les nouveaux mécanismes: fragmentation: MTU universelle, any cast, renumérotation simplifiée d'un plan d'adressage.
Plan d'adressage.
Adressage actuel, attribution des adresses.
Agrégateurs : découpage TLA/NLA/SLA/IID.
Intégration des Regional Registries
Entêtes: Mobilité (entête 135), Shim6, sécurité (mise en œuvre de l'entête calipso), confidentialité et entête d'authentification.
Problème des entêtes nœud-par-nœud.
Entêtes spécifiques: Fragmentation, Destination

Fonctionnement du multi-cast

Les groupes prédéfinis. Ajout d'un groupe, inscription.
Utilisation du multi-cast dans l'autoconfiguration.

Commandes de base et outils réseau.

Utilisation des outils de base en IPv6: wireshark, tcpdump, ping6, traceroute6, ifconfig, nmap, wget, iptraf, netstat, ip6tables, ...

Produits

Supports natifs sur les produits d'infrastructure: messagerie (postfix/dovecot), connexions (ssh), SNMP, NFS, ldap, proxies, ...
Supports sur les produits métiers:
Web (apache, firefox, IE), tomcat, JBoss, WebSphere

ICMPv6

Auto-configuration. Découverte des voisins (NDP), découverte des routeurs: fonctionnement, activation, activation partielle, désactivation.
Mise en place de radvd. Analyse des trames de découvertes.

Routage

TP de mise en œuvre du routage IPv6 en mode statique.
Activation du mode automatique, visualisation des tables de routage obtenues.

IP: Migration vers IPv6

Gestion des adresses	Mode sans état, avec état. DHCPv6: Présentation. Mise en œuvre d'un serveur dhcpv6. Cohabitation avec IPv4. Attribution statique d'adresses. Gestion du DUID. Stateless Address Autoconfiguration (SAA): Utilisation de radvd en complément. Relais DHCPv6. Cycle de vie des adresses. Adressage aléatoire. Migration d'opérateurs. Choix de l'adresse client.
DNSv6	Mise en œuvre d'un DNS v4/v6. Les différentes implémentations. DNS dynamiques.
Migration v4/v6	Les différentes approches: double pile, encapsulation statique, encapsulation dynamique. Impacts de la suppression du NAT. Utilisation du cycle de vie des adresses. Les différents tunnels. Mise en œuvre d'un tunnel 6sur4. Présentation de l'encapsulation v6 dans v4 et l'extension Teredo pour les réseaux à translation d'adresses Comparaison des différentes approches: isanat/Teredo Choix d'un tunnel broker.
Cas concret	Exemple de migration d'une infrastructure complète. La sécurité: IPsec, les pare feux et les filtres. ip6tables, routeurs et répartiteurs de charge, haproxy. Organisation de la migration. Impacts de la mise en œuvre de plusieurs dhcpv6 et de multiples agents RA. Problème de boucles de tunnels.

Messagerie

Durée: 2 jours

Prix et dates: nous consulter

Public:

Administrateurs réseaux.

Objectifs:

Savoir installer, configurer et administrer une messagerie sous Unix.

Connaissances préalables nécessaires:

Il est demandé aux participants de connaître les notions de base sur les réseaux TCP/IP.

Programme:

TCP/IP	Mode de fonctionnement: Adressage IP, nommage DNS (serveur de noms) TP: écriture d'un plan d'adressage et mise en œuvre
DNS	Fonctionnement, configuration du service.
Concepts de messagerie	Terminologie : SMTP, POP3, IMAP4, MTA, MDA, MUA... Le routage de messages. Anatomie d'un message, les champs d'entête
Architecture distribuée	Les clients, modes d'accès au courrier les protocoles : POP, IMAP, principes de fonctionnement. Etude du protocole POP3. Les extensions SMTP. Gestion des pièces jointes. TP : mise en place d'un système complet de messagerie, configuration d'un serveur SMTP (Postfix)
Marché	Les produits du marché Présentation des serveurs sendmail, Postfix.
Exploitation	Définitions d'alias. Traitements à l'arrivée (procmail, formail). Anti-spam, clamAV, p3scan. Sécurisation, chiffrement, authentification.

Sécurité TCP/IP

Durée: 5 jours

Prix et dates: nous consulter

Public:

Toute personne souhaitant maîtriser la sécurité sur TCP/IP, et plus particulièrement les administrateurs et les architectes réseaux.

Objectifs:

Savoir mettre en œuvre les mécanismes de sécurité, analyser les traces, configurer les systèmes de protection, concevoir une architecture de réseau fiable.

Connaissances préalables nécessaires:

Il est demandé aux participants de connaître les bases de TCP/IP.

Programme:

Introduction	Analyse des risques. Exemples avec l'étude des flux: tcpdump, sniffit, TP: visualisation des mots de passe transitant par le réseau.
Contrôle des accès système	Protection de services réseaux: telnet, tftp, snmp, ftp, ... Le 'tcp wrapper' Verrouillage des accès physiques à distance. Connexions sécurisée: SSH (configuration, connexion automatique), ssl, sftp, scp, tunneling X11 Contrôle de la messagerie: clamAV, p3scan, pop3s, imap4s Gestion des accès: Radius
Architecture de sécurité	: Coupe feux: DMZ, Proxy. Pose de filtres sur un routeur. TP: mise en place d'un proxy ftp iptables, PAT, stunnel, VPN (openvpn, freeSWAN), VLAN Les apports d'IPsec. TP: mise en place d'une architecture openVPN
Sécurisation échanges	Chiffrage des données, mécanisme des certificats
Surveillance	Le protocole SNMP; la surveillance d'applications TP: écriture d'un analyseur de topologie Contrôle des flux. Analyseurs de trames: ethereal, tcpdump, analyse de failles: nessus

Installation et Configuration WIFI

Durée: 2 jours
915 €

du 1er au 2 Mars
du 21 au 22 Juin

du 20 au 21 Septembre
du 8 au 9 Novembre

Public:

Les ingénieurs réseaux souhaitant aborder les technologies sans fil, les chefs de projet et toute personne souhaitant comprendre les mécanismes du wifi.

Objectifs:

Connaître les principes de fonctionnement du wifi, les conditions d'utilisation, et savoir élaborer une architecture de réseau à base de wifi.

Connaissances préalables nécessaires:

De bonnes connaissances réseau sont souhaitées.

Programme:

Les réseaux sans fil	le besoin avantages, inconvénients par rapport aux réseaux fixes les différentes catégories de réseaux sans fil
technologie WIFI	la norme 802.11, lien avec Ethernet, l'architecture, le mode ad hoc, les performances, Les fréquences utilisées.
Mise en pratique	les équipements, les outils disponibles sur le marché Configuration d'un réseau WIFI : les points d'accès, mise en œuvre en mode répéteur, mode pont l'alignement d'antennes
Sécurité	les problèmes le WEP, les insuffisances chiffrement RC4, Mise en place de WPA et de EAP-TLS Comment fiabiliser : Radius, IPsec, VPN
Utilisation du Wifi avec Linux	Les pilotes Wireless Tools. Utilisation de hostapd pour créer un point d'accès.
Le roaming	Présentation de la norme 802.11f, la gestion des utilisateurs itinérants

Bus One-wire

Durée: 2 jours
925 €

du 1er au 2 Mars
du 24 au 25 Mai

du 13 au 14 Septembre
du 8 au 9 Novembre

Public:

Toute personne intéressée par l'intégration d'un système 1 wire pour la commande à distance.

Objectifs:

Connaître les principes de fonctionnement du bus 1 wire et comprendre sa mise en œuvre.

Connaissances préalables nécessaires:

Aucune connaissance préalable n'est requise pour suivre ce cours.

Programme:

Introduction	Présentation de quelques technologies réseaux: Ethernet généraliste, CAN, FlexRay et Most en industriel, X10, I2C et 1wire en domotique.
Bus 1wire	Spécifications. Mode de fonctionnement. Mode vampire, mode trois fils. Présentation des produits: identifiant, capteur de température, interrupteurs, jauges, mémoires, horloges, convertisseurs analogiques, potentiomètres numériques.
Mise en œuvre	Ecriture d'un pilote 1wire. Connexion et configuration du pilote matériel DS2480. Solution 9097/9490. Connexion des capteurs. Mise en œuvre dans une application de supervision de capteurs: tout ou rien, jauge, température. Mesure des vitesses de transfert.
Embarqué	Portage du système 1wire sur une plateforme embarquée openWrt. Extension du 1wire sur des réseaux étendus avec le coupleur 2409.

Mise en œuvre du protocole SNMP

Durée: 2 jours
935 €

du 6 au 7 Mars
du 20 au 21 Juin

du 18 au 19 Septembre
du 13 au 14 Novembre

Public:

Les administrateurs réseau, et toute personne souhaitant mettre en place un système de supervision par SNMP.

Objectifs:

Comprendre le mécanisme de fonctionnement de SNMP, connaître les outils et produits permettant une utilisation efficace de SNMP dans la supervision du réseau.

Connaissances préalables nécessaires:

Il est demandé aux participants de connaître les bases de TCP/IP.

Programme:

Définitions supervision	Objectifs, méthodes, déterminer les objets à superviser, granularité des tests, techniques : prélèvements par SNMP, commandes de vérifications, outils spécifiques de supervision.
Le protocole SNMP	Simple Network Management Protocol Définitions d'objets à superviser, spécifications : RFC 1213. Historique : depuis SNMP v1, jusqu'aux apports de SNMP v3 (contrôle d'accès, chiffrement, ..) Schéma de principe : les requêtes get/set, les agents SNMP.
Fonctionnement	Le principe des MIB. La hiérarchie SNMP. Les zones privées. Exemples avec http et ftp. Détail d'une MIB. Fonctionnalités : Exemples : surveillance des différentes ressources d'un poste, exécution de processus distants
Mise en pratique	Commandes d'interrogation des agents SNMP : snmpget, snmpwalk, Notions de communauté et d'Oid (Object Identifier). Configuration d'un agent snmp sous Linux. Exécution de l'agent comme un service. Interrogations simples : description des cartes réseaux du poste client, affichage de la table de routage, ...
Outils d'interrogation	Graphiques : PTKMib, Mib Browser, MIB Smithy, Automatisation des requêtes avec net-snmp et scli (en mode commande).
Les alertes	Création d'un serveur d'alertes avec snmptradd. Définition des conditions d'alertes pour chaque objet.
Sécurité	Authentification Protection du contenu

Mise en œuvre du protocole SNMP

L'usage de SNMP sur le marché

Les produits d'analyse, les MIBs développées par les constructeurs.

Développement

Développement d'une MIB. Présentation des produits de développement.

Description de la structure en ASN-1.

Travaux pratiques :

conversion en C et compilation dans l'agent SNMP,

ajout d'OID surveillant la température du processeur,

ajout d'OID surveillant le nombre de threads d'un serveur JEE.

Voix sur IP

Durée: 4 jours
1840 €

du 5 au 8 Mars
du 18 au 21 Juin

du 17 au 20 Septembre
du 5 au 8 Novembre

Public:

Les administrateurs réseaux, et toute personne souhaitant mettre en place un système de communication audio sur IP.

Objectifs:

Comprendre le mécanisme de fonctionnement de la voix sur IP, connaître les outils et produits permettant une utilisation efficace de la voix sur IP dans l'entreprise.

Connaissances préalables nécessaires:

Il est demandé aux participants de connaître les bases de TCP/IP.

Programme:

Présentation	<p>Définitions: RTC, le réseau téléphonique commuté; La commutation de paquets avec IP. Les limites du réseau RTC. Avantages et inconvénients de la voix sur IP Les chaînes de transport en RTC, et en VoIP. La modulation PCM64/ Codec G711.</p>
Architectures et protocoles	<p>Les différents composants d'un réseau VoIP : routeurs, passerelle, terminaux. Les principaux protocoles : H323, SIP (Session Initiation Protocol). Fonctionnalités apportées par chaque protocole. Comparaison. Etude du protocole SIP : signalisation, et gestion des paramètres de session. Principe du Registrar SIP. Adressage: affectation d'un numéro à une adresse IP. Les requêtes les plus courantes. Les proxy SIP : exemple de déploiement SIP entre deux réseaux d'entreprises. Le protocole de transport RTP : Real Time Protocol</p>
Qualité	<p>Qualité de la voix. Compression. Détection des défauts. Les différents codecs pour l'encodage de la voix. Bande passante utilisée. Calibrage. Utilisation de RSVP (Resource Reservation Protocol), MPLS (Multiprotocol Label Switching) ou TOS. Mesure du trafic. Utilisation des iptables. Gestion de la qualité de service et répartition voix/données.</p>
Mise en pratique	<p>Configuration de téléphones SIP depuis un navigateur : choix du codec, renseignement de l'adresse SIP, indication d'un serveur DNS, ... Installation et configuration d'un registrar Asterisk. Création des comptes des téléphones et du dialplan (plan de numérotation).</p>

Voix sur IP

Les enregistrements SRV	<p>Communication entre des réseaux différents. Utilisation des serveurs DNS. Travaux pratiques : Création de deux domaines DNS, mise à jour des enregistrements SRV pour la localisation des services sip/udp. Configuration des softphones, et tests. Mise en évidence des requêtes DNS par les traces réseau.</p>
Plan de numérotation	<p>Travaux pratiques avec le dialplan Asterisk : /etc/asterisk/extensions.conf. Notions de contexte, extensions, priorités, et applications. Configuration d'appels stateful et stateless.</p>
La sécurité	<p>Critères de sécurité : Disponibilité, Intégrité, Confidentialité. Mise en évidence des risques pour chaque critère. Contraintes légales concernant les appels d'urgence. Travaux pratiques : à partir d'un analyseur réseau (wireshark), capture de trames VoIP sur le réseau, relecture des échanges téléphoniques, saturation d'un serveur, déni de service. Solutions techniques : traitement des appels d'urgence par Asterisk, chiffrement du protocole SIP, chiffrement des flux RTP avec RTPS, mise en place de tunnels, solutions haute disponibilité pour les serveurs.</p>
Déploiement de masse	<p>Le provisioning : automatisation de la configuration des terminaux téléphoniques. Avantages : gestion d'un nombre important de téléphones (logiciels ou matériels). Principe : distribution d'adresses IP dynamiques par un serveur BOOTP/DHCP, mise à disposition des fichiers de configuration pour chaque téléphone. Travaux pratiques : configuration du provisioning pour des terminaux SIP (Linksys de Cisco).</p>

Voip - Configuration Asterisk

Durée: 2 jours
935 €

les 15 et 16 Mars
les 28 et 29 Juin

les 27 et 28 Septembre
les 15 et 16 Novembre

Public:

Les administrateurs réseaux, et toute personne souhaitant utiliser le logiciel Asterisk en téléphonie IP.

Objectifs:

Comprendre le fonctionnement d'Asterisk, et savoir paramétrer le logiciel afin d'utiliser l'ensemble des fonctionnalités proposées.

Connaissances préalables nécessaires:

Il est demandé aux participants de connaître les principes de la Voix sur IP

Programme:

Asterisk introduction	Présentation, fonctionnalités. Installation et configuration de base.
Utilisation du registrar SIP	Notion de registrar Implémentation : création des comptes téléphones et du "dialplan"
Enregistrements SRV	Communications entre domaines Création de DNS et configuration Asterisk Mise en œuvre et tests
Plan de numérotation	Définitions : contextes, applications Configuration /etc/asterisk/extensions.conf : syntaxe générale, contextes, extensions, priorités. Configuration des Applications : Dial(), Record(), CDR (Call Details Record) Syntaxe détaillée : les opérateurs, expressions, motifs, branchements conditionnels, fonctions et macros.

Performances et optimisation des réseaux

Durée: 2 jours

Prix et dates: nous consulter

Public:

Consultants réseaux, responsables techniques, responsables réseaux, architectes, intégrateurs, chefs de projet possédant des connaissances générales en réseaux.

Objectifs:

Identifier les besoins et les critères de performances des flux transmis sur les réseaux IP. Evaluer l'impact des applications sur les performances d'un réseau. Utiliser des techniques d'estimation et de mesures pour valider les performances. Concevoir des réseaux évolutifs et optimisés.

Connaissances préalables nécessaires:

Notions sur le câblage, transmissions et les bases de IP.

Programme:

Mesure de la charge ponctuelle

Présentation des outils de base de surveillance des paquets: tcpdump, sniffit, wireshark.
iptraf, snmp, ntop
TP: mise en œuvre d'un script de détection de surcharge sur un brin.

Mesure de la charge entre deux points

Calcul de la bande passante disponible. Mesure de base avec ping. Mesure du taux de perte de paquets, du délai de transit et valeur de la gigue avec iperf.
Présentation des produits Argus, et de l'architecture de collecte nfdump/nfsen.

Outils de graphes

MRTG, Cricket, Cacti. Comparaison.

Solutions

Problèmes liés à l'architecture: pose de liens inter-routeurs et de dérivation.
Amélioration du routage: RIP, OSPF, découpage en systèmes autonomes. Gestion des VLANS. Duplication de port sur un commutateur.
Gestion de la qualité de service.
régulation du flux en fonction du protocole
TP: mise en œuvre des tc avec CBO puis HTB sur une interface.

Annuaire LDAP

Durée: 2 jours
905 €

du 1er au 2 Mars
du 9 au 10 Mai

du 6 au 7 Septembre
du 13 au 14 Décembre

Public:

Administrateurs réseaux, intégrateurs d'application souhaitant configurer un annuaire ldap.

Objectifs:

Comprendre le mécanisme de ldap, et savoir mettre en place un service d'annuaire. Les travaux pratiques ont lieu avec Openldap.

Connaissances préalables nécessaires:

Il est demandé aux participants de connaître les bases de TCP/IP.

Programme:

Introduction	le besoin, historique. Définitions.
Procotole LDAP	Lightweight Directory Access Protocol Principe de fonctionnement. Les modèles, la conception d'une arborescence : construction, importation de schema
Mise en œuvre	Travaux pratiques avec OpenLdap : installation, configuration du serveur. Les backends openldap. Définition d'index pour l'optimisation de la recherche dans la base.
Hiérarchie ldap	Construction de la hiérarchie : distinguished name, relative distinguished name. Le format ldif. Utilisation : commandes de recherche dans l'annuaire (search, compare, add, modify, delete, rename, ...) Travaux pratiques : création de fichiers ldif, ajout à l'annuaire avec la commande ldapadd, vérification avec la commande ldapsearch.
Schéma ldap	Définitions : attributs, objets Format du schéma. Mécanisme d'héritage des attributs. Organisation d'un schéma. Travaux pratiques : intégration d'un schéma extérieur.

Annuaire LDAP

Gestion de l'annuaire	Outils de création d'un annuaire. Migration de comptes unix vers ldap. Importation de fichiers ldif. Méthodes et commandes de consultation dans un annuaire. Travaux pratiques : recherche composée dans un annuaire. Consultation depuis un client de messagerie. Gestion des permissions. Outils graphiques de consultation.
Sécurité	Authentification, contrôle d'accès, chiffrement des transactions Annuaire et PKI
Architecture	Distribution, réplication d'annuaires. Le besoin de synchronisation, les méthodes. Travaux pratiques : création d'un serveur esclave, avec réplication de toutes les informations du serveur maître.
Intégration	Dans le réseau d'entreprise : Exemples : pam/ldap, samba/ldap. Travaux pratiques : création d'un annuaire ldap pour samba, configuration d'un module pam-ldap, Mise en œuvre pour un serveur de messagerie.
Le marché	Présentation des principaux annuaires.

Supervision nagios : utilisation

Durée: 3 jours
1380 €

du 23 au 25 Janvier
du 26 au 28 Mars
du 4 au 6 Juin

du 24 au 26 Septembre
du 5 au 7 Novembre
du 17 au 19 Décembre

Public:

Exploitants et utilisateurs d'un système de supervision Nagios, et de l'interface Centreon.

Objectifs:

Connaître les fonctionnalités de Nagios, maîtriser l'interface d'exploitation Centreon, savoir ajouter de nouveaux tests, savoir mettre en place une politique d'alerte, créer de nouveaux types de notification.

Connaissances préalables nécessaires:

Connaissance de l'architecture d'un système d'information, bases tcpip et bases systèmes.

Programme:

Supervision : définitions	Les objectifs de la supervision, les techniques disponibles. Objets supervisés.
Les services et ressources	Rappels sur les principes HTTP, SMTP, NNTP, POP3, PING. Définition des ressources à surveiller.
Présentation de Nagios	Les fonctionnalités Supervision, exploitation. Surveillance des services réseaux, Surveillance des ressources (charge CPU, espace disque) Envoi d'alarme vers des contacts déterminés; Déclenchement de scripts pour corriger les problèmes.
Utilisation	Les premiers pas avec Nagios : la page d'accueil. Travaux pratiques : utilisation de Nagios pour la supervision d'un ensemble d'hôtes et de services de test. Vue d'ensemble de l'état du réseau. Les hôtes et services Cartographie du réseau Visualisation des tests Détection des pannes Recherche d'hôte Arrêts programmés
Configuration Nagios	Objets à définir : hôtes, groupes, services, dépendances, notifications, escalades Description des serveurs à surveiller, des contacts, création de groupes de serveurs, de groupes de contacts Notion de hiérarchie avec les hôtes parents, les dépendances de services, hôtes et groupes.
Déploiement	Sur les hôtes, principes de NRPE, NSCA. Travaux pratiques : écriture de scripts de déploiement. (NRPE)

Supervision nagios : utilisation

Les plugins	Principe de fonctionnement. Quelques plugins courants
L'interface Centreon	Les fonctionnalités, les sites de référence, L'architecture Nagios/Centeron. Le positionnement par rapport à Nagios
Installation de Centreon	Prérequis Travaux pratiques : Installation des bibliothèques, configuration de Centreon et de l'interface web.

Administration Nagios

Durée: 5 jours
2195 €

du 23 au 27 Janvier
du 26 au 30 Mars
du 4 au 8 Juin

du 24 au 28 Septembre
du 5 au 9 Novembre
du 17 au 21 Décembre

Public:

Les administrateurs systèmes, administrateurs réseau, et toute personne souhaitant mettre en place un système de supervision.

Objectifs:

Connaître les fonctionnalités de Nagios. Savoir installer, configurer et administrer le produit. Savoir développer des nouveaux plugins.

Connaissances préalables nécessaires:

Notions sur le réseau, bases de TCP/IP. Connaissance d'un langage de script.

Programme:

Supervision : définitions	Les objectifs de la supervision, les techniques disponibles. Rappels sur les principes HTTP, SMTP, NNTP, POP3, PING Définition des ressources à surveiller.
Présentation Nagios	Les fonctionnalités Supervision, exploitation. Surveillance des services réseaux, Surveillance des ressources (charge CPU, espace disque).
Architecture	Principe de fonctionnement et positionnement des différents modules. Les plugins et extensions
Installation	Configuration requise. Site de référence. Travaux pratiques : Installation et mise à jour, Paramétrage de base, démarrage Nagios
Utilisation de nagios	Premiers pas avec nagios : la page d'accueil. Vue d'ensemble de l'état du réseau. Détail des hôtes et services. Cartographie du réseau. Détection des pannes réseau. Les hôtes et services. Travaux pratiques : recherche d'un hôte, arrêt programmée d'hôtes et services. Liste des vérifications programmées. Edition de rapports.

Administration Nagios

Configuration	<p>Etude du fichier de configuration standard nagios.cfg. Description des serveurs à surveiller, création de groupes de serveurs. Description des contacts, et création de groupes de contact, escalades Définition des services et groupes de services. Les notions de hiérarchie, dépendances : hôtes et services. Configuration de l'interface web d'administration. Etude du fichier cgi.cfg</p>
Optimisation de l'ordonnanceur	<p>Méthode d'ordonnancement. Délai entre chaque test. Entrelacement des services. Tests concurrents. Fréquence de récupération.</p>
Contrôle et débogage	<p>Analyse des fichiers de logs. Commandes de contrôle. Mode d'exécution des plugins. Options détaillées.</p>
Les plugins	<p>Principe de fonctionnement. Mise en œuvre des plugins standards. Travaux pratiques : Personnalisation de Nagios par développement de nouveaux plugins.</p>
Gestionnaire d'évènements	<p>Mécanisme de traitement d'erreur. Normalisation. Algorithmie de l'ordonnanceur. Macros d'évènements. Démarche d'implémentation. Exemple : relance d'un serveur web.</p>
Lien SNMP	<p>Présentation du protocole SNMP. Hétérogénéité des superviseurs et du parc supervisé. Tests actifs et passifs.</p>
Supervision distribuée	<p>Principe des agents. Sur les hôtes, principes de NRPE, NSCA. Travaux pratiques : écriture de scripts de déploiement. (NRPE) installation de nsca, et configuration.</p>
Superviseurs redondants	<p>Méthodes de redondance. La haute disponibilité : mode fail-over, configuration d'un superviseur secondaire, Gestionnaire : panne du superviseur, panne du service nagios. Greffon de test du maître.</p>
Centralisation NDO	<p>Fonctionnalités et composants. Travaux pratiques : Mise en œuvre de NDO</p>

Administration Nagios

Extensions Nagios

Intégration avec Cacti, Centreon

La solution FAN : Full Automated Nagios

Présentation : installation intégrée, logiciels fournis, versions supportées.

Démonstration d'une installation à partir de FAN.

Reporting avec Cacti

Durée: 3 jours
1380 €

du 19 au 21 Mars
du 21 au 23 Mai

du 17 au 19 Septembre
du 26 au 28 Novembre

Public:

Administrateurs systèmes et réseaux souhaitant mettre en œuvre le logiciel Cacti pour mesurer les performances.

Objectifs:

Comprendre le fonctionnement de cacti, savoir l'installer et le configurer.

Connaissances préalables nécessaires:

Connaissances des systèmes linux, des réseaux TCP/IP.

Programme:

Introduction	Présentation des fonctionnalités : mesure de performances, représentation graphique des données d'état de équipements, ou des suivis de performances. L'architecture de cacti : serveur web, RRDTool pour le stockage, scripts avec PHP, bash, Perl, ..
Installation	Prérequis : RRDTool, MySQL, PHP, un serveur http (Apache ou IIS). Les packages RPM nécessaires, les modules PHP, configuration du serveur apache et de la base MySQL. Installation et configuration de cacti.
Fonctionnement	La collecte de données, le stockage, la présentation. Configuration du poller, collecte d'informations par snmp, Stockage et graphiques avec RRDTool (Round Robin Database).
Les graphes	Utilisation de cacti pour la déclaration des devices, et la création des graphes. Configuration du hostname, host template, des options SNMP, .. Visualisation des graphes : création d'arbres.
Gestion des utilisateurs	Les utilisateurs par défaut : admin et guest, modification (options de connexion, d'accès aux graphes,...), création, suppression d'utilisateurs, activation/désactivation, ...
Les templates	Principe, apport des templates. Les templates de base : Data, graph et host templates. Création de templates , modification, import, export de templates.

Zimbra : installation et configuration

Durée: 3 jours
1470 € HT

du 26 au 28 Mars
du 14 au 16 Mai

du 27 au 29 Août
du 22 au 24 Octobre

Public:

Administrateurs réseaux et toute personne souhaitant comprendre le fonctionnement du produit Zimbra.

Objectifs:

Savoir installer, configurer et administrer l'outil Zimbra.

Connaissances préalables nécessaires:

Il est demandé aux participants de connaître les notions de base sur les réseaux TCP/IP, et sur les systèmes Linux.

Programme:

Introduction	Présentation Zimbra Collaboration Suite Fonctionnalités : plate-forme collaborative pour gérer les agendas, partager des documents, des contacts, émettre/recevoir des mails, Les différents clients : applications, solutions web, ou mobile Solution de service hébergée Positionnement par rapport aux produits concurrents : OpenGroupware, PHPgroupware, ... Lotus Domino, MS Exchange, ...
Architecture	Serveur Groupware Différents accès clients : Applications : Zimbra desktop, ou thunderbird ou Outlook Express; Navigateur pour le client web; Accès mobile Composants du serveur : Apache, postfix, MySQL, Openldap, Lucene, ...
Installation et configuration	ZCS Server Prérequis Configuration DNS Installation du serveur Zimbra et des composants : ldap, postfix, apache, jetty, ... Configuration de base : domaine ldap, mot de passe, anti-spam, messagerie, etc .. Comptes utilisateurs et groupes
Accès clients	Installation de Zimbra Desktop Mise en œuvre de l'interface web : Zimbra webclient Démonstration de la synchronisation avec Zimbra Mobile Webclient sur android

Gestion de Parc avec OCS et GLPI

Durée: 3 jours
1375 € HT

du 20 au 22 Février
du 10 au 12 Avril
du 25 au 27 Juin

du 15 au 17 Octobre
du 3 au 5 Décembre

Public:

Administrateur devant gérer un parc de systèmes dans un environnement de production.

Objectifs:

Savoir installer et configurer, et utiliser les outils OCS et GLPI.

Connaissances préalables nécessaires:

Une bonne connaissance des systèmes Unix/Linux est nécessaire.

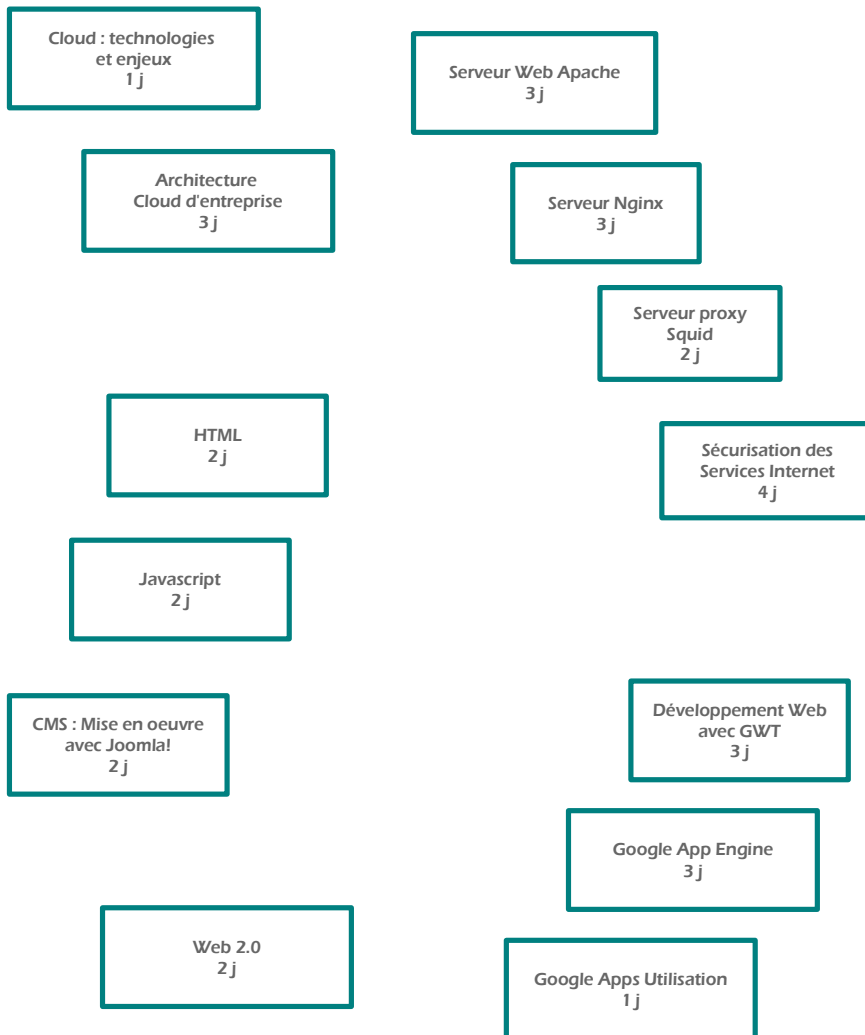
Programme:

Introduction	<p>Le besoin : inventaire et suivi des configurations matérielles et logicielles</p> <p>Présentation OpenComputer and Software Inventory Next Generation</p> <p>Fonctionnalités, informations collectées</p>
Architecture OCSNG	<p>architecture client/serveur</p> <p>Les composants du serveur OCSNG : base de données, serveur de communication, agents, serveur de déploiement, console d'administration</p> <p>Intégration avec GLPI</p>
Installation	<p>systèmes supportés, installation depuis les packages</p> <p>création des utilisateurs</p> <p>Les agents : procédure d'installation et configuration du lancement au démarrage du système</p>
Configuration	<p>configuration par l'interface web: fonction IPDISCOVER, détection des doublons (adresses Mac)</p> <p>notion de TAG : quelques exemples d'utilisation</p> <p>Les différents paramètres de chaque machine gérée.</p>

Gestion de Parc avec OCS et GLPI

Utilisation glpi	<p>La Console Centrale, l'accès aux différentes fonctionnalités.</p> <p>L'inventaire : requêtes prééfinies. Mise à jour des TAGS, recherche multi-critères, recherche par analyse du TAG, export des données, création de gabarits.</p> <p>Le module Administration : gestion des droits, affectation des logiciels à une catégorie, affectation des ordinateurs à une entité, création de règles, utilisation de dictionnaires.</p> <p>Le module Configuration : définition et modification des composants, configuration de l'affichage, du niveau de journalisation, . Notifications par mail.</p> <p>Le module Assistance (help-desk) : suivi des incidents, gestion des appels, des tickets, du planning, des interventions, extraits de statistiques.</p>
Les plugins	<p>Principe des plugins. Mise en œuvre pratique des plugins tracker, reports, racks, data injection, item unistallation.</p>
Import OCS NG	<p>Objectif et principe de fonctionnement. Configuration du mode OCSNG. Options d'importation. Mode d'import OCS. Import OCS par le plugin massocsimport.</p>
Gestion du cycle de vie.	<p>Le besoin. Plugin nd'injection de fichiers CSV (data_injection) Liaison d'une machine. Plugin de désinstallation d'une machine, de suppression d'une machine. Statuts des matériels. Gestion des machines en stock. Machines en réparation et réformées.</p>
Télédéploiement de paquets	Principe et architecture. Notion de priorité et action à exécuter
Exploitation OCS glpi	Sauvegardes, journalisation. Lien avec un annuaire ldap.

Filières Internet et Web



Cloud : technologies et enjeux

Durée: 1 jour
510 € HT

le 9 Février
le 13 Avril

le 19 Juillet
le 4 Octobre
le 13 Décembre

Public:

Architecte, chef de projet, et toute personne souhaitant comprendre la notion de cloud, et plus précisément les solutions de cloud d'entreprise.

Objectifs:

Connaître les possibilités des solutions de cloud, ainsi que les contraintes de mise en œuvre.

Connaissances préalables nécessaires:

Connaissance générale des systèmes d'informations.

Programme:

Introduction

Définition : cloud computing,
Les types de clouds: SaaS, PaaS, IaaS. Clouds privés et publics.
Fonctionnalités disponibles :
partage de données,
stockage distant (services EBS et S3 d'Amazon)
mise à disposition de services (SNS, SQS, ...), applications,
pilotage distant de systèmes locaux,
mise à disposition de ressources à la demande.

Technologies

virtualisation, services Web
Les acteurs du marché des clouds: Amazon, Eucalyptus, IBM, microsoft, openStack, rackspace, salesForce.
Comparaison des offres.

Positionnement par rapport aux autres architectures

Positionnement par rapport aux centres de données (datacenter):
vSphere
La gestion de serveurs IBM en grappe: Capacity On Demand, console HMC et micro-partitionnement.
Les architectures JEE en grappes. Présentation de WebSphere Cloud.
Comment migrer les applications vers un cloud?

Cloud d'entreprise

Comment s'organise une infrastructure d'entreprise en cloud?
Comment migrer les services réseau: dns, dhcp, smtp, ...
Organisation des machines virtuelles en sous-réseaux, switches et câbles virtuels. Découpage d'une grappe cloud en zones.
Urbanisation.
Les répartiteurs de charges disponibles dans les clouds.
Est ce que le cloud sera compatible IPv6?
Comment intégrer la voix sur IP dans un cloud?

Architecture cloud d'entreprise

Durée: 3 jours
1515 € HT

du 6 au 8 Février
du 10 au 12 Avril

du 16 au 18 Juillet
du 1er au 3 Octobre
du 10 au 12 Décembre

Public:

Architecte, chef de projet, et toute personne souhaitant comprendre la notion de cloud, et plus précisément les solutions de cloud d'entreprise.

Objectifs:

Connaître les possibilités des solutions de cloud, ainsi que les contraintes de mise en œuvre.

Connaissances préalables nécessaires:

Connaissance générale des systèmes d'informations.

Programme:

Introduction

Définition : cloud computing,
positionnement par rapport aux autres architectures
Fonctionnalités disponibles :
accès facilité depuis n'importe quel terminal de connexion,
partage de données,
stockage distant,
mise à disposition de services, applications,
pilotage distant de systèmes locaux,
mise à disposition de ressources selon les besoins.
Exemples : AWS, Google, Salesforce, Office Web

Technologies

Les éléments qui permettent le développement du cloud:
outils de développement web : comme GWT, ajax, les webservices;
réseaux haut débit; réseaux sans fils;
voix sur IP;
terminaux mobiles, solutions embarquées;
outils de recherche (spécifiquement dans un environnement distribué);
datacenter; virtualisation de serveurs.

Cloud d'entreprise

Cloud privé :
infrastructure non partagée et dédiée à une entreprise ou une organisation.
Les avantages d'une solution de cloud privé : sécurité, localisation des données.
Solutions disponibles :
un environnement privé mis à disposition par un fournisseur de cloud standard,
des produits permettant d'installer un cloud interne à l'entreprise.
Exemples de produits de cloud privé :
Eucalyptus, Xen Cloud Platform, VMWare VSphere, Enomaly.
Fonctionnalités :
déploiement sur des clusters,
déploiement de composants sur différents serveurs,
possibilité de maintenance "on-line".

Architecture cloud d'entreprise

Mise en œuvre avec Eucalyptus

Présentation.
Architecture du produit : cloud-controller, walrus.
Travaux pratiques :
installation depuis les packages RPM.
Configuration :
enregistrement des composants Eucalyptus,
mot de passe, adresse IP du "cloud controller".
Configuration de l'hyperviseur (xen ou kvm).
Lancement d'une première machine virtuelle.
Mise en œuvre du réseau entre machines virtuelles.
Les différents mode de gestion du réseau par eucalyptus.
Outils d'administration d'eucalyptus :
interface web d'administration,
ligne de commande (gestion des nœuds, des utilisateurs, ..)

Méthodes et outils

Mise en œuvre du cloud computing :
Développement applications web avec GWT,
virtualisation de postes de travail avec nx, vmware Desktop,
virtualisation serveurs XenApp,

Serveur WEB : apache

Durée: 3 jours
1270 €

du 23 au 25 Janvier
du 12 au 14 Mars
du 4 au 6 Juin

du 24 au 26 Septembre
du 26 au 28 Novembre

Public:

Toute personne souhaitant configurer, installer et exploiter un serveur web à base d'apache.

Objectifs:

Comprendre le fonctionnement d'apache, et savoir le mettre en œuvre, l'installer, le configurer et l'administrer.

Connaissances préalables nécessaires:

Il est demandé aux participants de connaître les bases de TCP/IP.

Programme:

Introduction	Présentation, historique. Fonctionnalités. Travaux pratiques : installation, configuration de base Principe, le rôle des modules.
Protocole HTTP	Fonctionnement, format des requêtes. Méthodes. Syntaxe d'une URL.
Configuration du serveur	Environnement, gestion des processus, requêtes, connexions client : le fichier httpd.conf. Configuration des MPM, des DSO. Connexion des clients. Exemple de dialogue.
Configuration d'un site principal	Nom interne du serveur, rappel sur les DNS, page d'accueil, types de fichiers, les alias, définition de chemins relatifs, la balise index Ports et interfaces d'écoute. Distribution des documents. Documents par défaut et chemins relatifs. Travaux pratiques : mise en œuvre avec création d'un site et configuration du serveur apache.
Hôtes virtuels	Principe, configuration, hôtes virtuels basés sur l'adresse IP, sur le nom
WebDynamique	Principe des scripts CGI, fonctionnement, mise en place.
Exploitation Apache	Administration du site, personnalisation des pages d'erreurs Les fichiers journaux: analyse directe, analyse par webalizer Travaux pratiques : mise en charge du site et visualisation du comportement.

Serveur WEB : apache

Sécurité

- Exécution des processus.
- Portée des balises.
- Expressions rationnelles.
- Octroi de capacités.
- Restriction d'accès par répertoire, par méthode, ...
- Authentification HTTP.
- Sécurité avec SSL et HTTPS :
principe, configuration par défaut.
- Certificat et clé du serveur.
- Génération de clés SSL.

Administration serveur Nginx

Durée: 3 jours
1350 €

du 12 au 14 Mars
du 4 au 6 Juin

du 15 au 17 Octobre
du 17 au 19 Décembre

Public:

Toute personne souhaitant configurer, installer et exploiter un serveur Nginx.

Objectifs:

Comprendre le fonctionnement du serveur nginx, et savoir le mettre en œuvre, l'installer, le configurer et l'administrer.

Connaissances préalables nécessaires:

Il est demandé aux participants de connaître les bases de TCP/IP et des technologies web.

Programme:

Introduction	Présentation, historique, licence. Fonctionnalités : serveur http, proxy inverse, proxy de messagerie, diffusion de contenus vidéo, ... Plate-formes supportées. Architecture : principe de serveur asynchrone, modularité.
Mise en œuvre	Choix des modules, d'une version/distribution. Travaux pratiques : installation, démarrage, configuration de base
Protocole HTTP	Fonctionnement, format des requêtes. Méthodes. Syntaxe d'une URL.
Configuration du serveur nginx	Etude du fichier /etc/nginx/nginx.conf : http-block, server-block, location-block.
Configuration d'un site principal	Nom interne du serveur, rappel sur les DNS, page d'accueil, types de fichiers, les alias, définition de chemins relatifs, Travaux pratiques : mise en œuvre avec création d'un site et configuration du serveur nginx.
Virtuals Hosts, locations	Principe, configuration, hôtes virtuels basés sur l'adresse IP, sur le nom. Configuration des URLs.
Scripts CGI	Exécution d'applications en PHP, en Python.
Serveur proxy	Configuration Nginx en proxy inverse et en proxy messagerie.
Exploitation	Administration du site. Les fichiers journaux: création de fichiers de logs séparés pour chaque hôtel virtuel. Mécanisme de rotation des fichiers journaux. Analyse des informations stockées dans les logs. Travaux pratiques : mise en charge du site et visualisation du comportement.

Administration serveur Nginx

Sécurité

Le module HttpAuthBasicModule.
Mise en œuvre des directives `auth_basic` et `auth_basic_user_file`.
Création des login/mot de passe : `htpasswd-b` ou `htpasswd-perl`.
Restriction d'accès en fonction de l'adresse IP :
directives `allow` et `deny`,
et avec restriction par mot de passe : directive `satisfy`.

Migration

Comparaison des serveurs Nginx et Apache.
Conseils de migration.

Serveur Proxy Squid

Durée: 2 jours
905 €

du 13 au 14 Février
du 24 au 25 Mai

du 13 au 14 Septembre
du 22 au 23 Novembre

Public:

Toute personne souhaitant mettre en place un serveur Squid afin d'optimiser et sécuriser les accès Internet de l'entreprise.

Objectifs:

Comprendre les principes de base d'un serveur de cache Internet. Savoir élaborer les configurations optimisant le fonctionnement du serveur Squid. Mettre en place les contrôles d'accès dans une architecture solide et sécurisée.

Connaissances préalables nécessaires:

Des connaissances minimales sur TCP/IP (adressage, fonctionnement) ainsi que sur le fonctionnement du Web sont nécessaires pour suivre ce cours.

Programme:

Besoin	Serveur Proxy, fonctionnement, multi serveurs proxys, hiérarchie de serveurs cache, cache transparent, accélérateur Web ou proxy inversé.
Squid	Présentation, sécurité, architecture externe. Exemple d'utilisation, systèmes d'exploitation concernés, logiciels complémentaires.
Installation	Installation à partir de paquetages, fichiers de configurations, configuration de base, test du serveur.
Configuration des postes clients	configuration manuelle, automatique. Scripts d'auto-configuration, filtrage suivant DNS, par protocole. Clients en mode texte ,robots. Installation dans le navigateur.
Configuration du serveur	principe et syntaxe des ACL. Optimisation de l'utilisation du serveur. Restriction d'accès par hôte, par réseau, par plage horaire, par jour, par site. Mise en cache des données. Méthodes d'authentification.
Administration	Surveillance, support SNMP. Configuration par WebMin. Fichiers journaux
SquidGuard	Présentation, les groupes source, groupes de destination. Réécriture d'URL, règles d'accès. Principe de la base de données, utilisation, considérations de performances

Sécurisation des services internet

Durée: 4 jours
1860 €

du 30 Janvier au 2 Février
du 2 au 5 Avril
du 25 au 28 Juin

du 1er au 4 Octobre
du 17 au 20 Décembre

Public:

Les administrateurs réseaux, et toute personne souhaitant sécuriser les services internet et/ou intranet.

Objectifs:

Connaître les moyens de sécuriser les réseaux IP et d'apporter des solutions aux risques liés à internet.

Connaissances préalables nécessaires:

Connaissances de bases sur les réseaux IP et les systèmes d'exploitation.

Programme:

Introduction	<p>La sécurité : Définitions Le DIC : Disponibilité, Intégrité, Confidentialité Evaluation des risques. Elaboration d'une politique de sécurité. Définitions: DMZ, proxy, pare-feux, routeurs filtrants Fonctionnement, mise en œuvre</p>
Les ports de niveau 5	<p>Rappels sur la notion de ports. Principe d'empilement des trames. Les ports UDP. Exemples de trames.</p>
Outils réseaux	<p>Le principe des traces, les informations disponibles dans les captures de trames. Travaux pratiques : Traçage des flux réseaux: wireshark, tcpdump.</p>
DMZ et pare-feux	<p>Définition, et architecture Fonctionnement des pare-feux.</p>
Firewall iptables	<p>Description du filtrage avec iptables. Syntaxe. Traduction d'adresses. Traduction de ports. Notion de politique de sécurité par défaut. Sécurisation de l'ouverture d'un port. Travaux pratiques : Mise en œuvre d'une fonction d'activation d'ouverture avec les iptables.</p>

Sécurisation des services internet

SSH et GPG	<p>SSH : Secure SHell Présentation du protocole de connexion SSH Utilisation de la cryptographie. Authentification par clé. Travaux pratiques : configuration, sécurisation de l'ouverture du port.</p> <p>GPG : GNU Privacy Guard Transmission de messages chiffrés et signés.</p>
VPN et tunnels	<p>Objectifs. Fonctionnement. Mise en œuvre : tunnels ssh, stunnel. Travaux pratiques avec openVPN. Principe d'IPsec. Implémentation avec FreeS/WAN.</p>
Proxy	<p>Introduction : la fonction d'un serveur proxy. Les multi-serveurs proxys. Mise en œuvre avec squid. Architecture Squid. Exemples d'utilisation. Travaux pratiques : configuration réseau d'un serveur Squid. Les Access Control List. Optimisation de la bande passante. Programmes d'authentification. Exemple d'authentification NCSA, LDAP.</p>
Services reseaux	<p>Sécurisation DNS: architecture redondante, DNS fermé, Hidden master, Stealth DNS. Mise en œuvre de la sécurisation d'un DNS. Authentification. Les signatures TSIG. Les ACL.</p>
Messagerie	<p>Sécurisation. Notion de relais ouvert. Outils de test du serveur. Travaux pratiques : mise en œuvre avec postfix. Apport de Cyrus.</p>
Acces reseaux	<p>Authentification par un serveur radius. Techniques de filtrage à base d'iptables. Mise en œuvre d'un système d'authentification par utilisateur et non par adresse: NuFW.</p>
Détection de failles	<p>Outils : snort, openvas. Travaux pratiques : mise en œuvre et réalisation d'un audit de sécurité de l'environnement de test.</p>

HTML

Durée: 2 jours
950 €

du 27 au 28 Février
du 11 au 12 Juin

du 24 au 25 Septembre
du 10 au 11 Décembre

Public:

Développeurs de pages HTML.

Objectifs:

Etre capable de développer des documents HTML. Explorer l'ensemble des possibilités offertes par HTML.

Connaissances préalables nécessaires:

Connaissances générales sur Internet.

Programme:

Présentation	HTML : le langage du Web. Contexte historique.
Environnement et structure	Le concept des balises. Le bon usage des balises. Particularités et pièges du HTML. Le document HTML minimum.
Mise en forme	Document: Objets d'un document. En-tête d'un document HTML. Texte: Titres. Paragraphes. Contrôle de passage à la ligne. Formatage, alignement, taille, couleur, police. Caractères spéciaux. Commentaires. Autres balises de texte.
Les listes	Numérotées. A puces. Imbriquées. Les listes de définition.
Les tableaux	Cellules. Fusion. Gestion de la taille du tableau. En-tête et légende. Bordures. Groupes de colonnes et de lignes.
Les hypertextes	Balise de liens. Liens externes, liens internes. Types d'URL: site, adresse électronique, téléchargement. Target. Attribut title. Couleur des liens. Liens et feuilles de style.
Images	Formats des images du Web. Insertion d'une image, espace autour d'une image, alignement, couleur d'arrière-plan, image d'arrière-plan, lien sur une image. Images réactives
Les éléments multimédias	Flash, Real, Quicktime, Pdf et d'applets Java
Frames	Page de cadres élémentaire. Cadre avec colonnes. Cadre avec lignes. Attributs des cadres. Liens entre les cadres. Cadres en ligne (balise iframe). Cadre ou tableau.
Formulaires	Déclaration de formulaire. Zone de texte mono-ligne, multi-lignes. Menu déroulant. Boutons radio, checkbox. Envoi, annulation, commande. Formulaires cachés, formulaires de transfert de fichier, de mot de passe. Organisation des éléments d'un formulaire.
Les feuilles de style	Les propriétés : couleur, police, marge Les classes Dynamiser un site avec CSS Gérer les positionnements

Web Dynamique avec JavaScript

Durée: 3 jours
1350 €

du 29 Février au 2 Mars
du 13 au 15 Juin

du 26 au 28 Septembre
du 12 au 14 Décembre

Public:

Toute personne désireuse d'acquérir la maîtrise de la programmation en langage JavaScript, son intégration dans les pages HTML, et la méthode AJAX.

Objectifs:

Développer l'interactivité de sites Web grâce à l'utilisation du langage JavaScript et exploiter les fonctionnalités des feuilles de style à travers Javascript.

Connaissances préalables nécessaires:

Il est demandé aux participants de connaître le principe de fonctionnement d'Internet et du Web, le langage HTML. La connaissance d'un langage de programmation est utile.

Programme:

Introduction	Besoin, historique, cadre d'utilisation.
Rappels sur les formulaires HTML	Mode d'appels des scripts depuis les pages HTML. Comment créer une zone de saisie? Les différents modes de saisie HTML.L'interpréteur JavaScript: entrées/sorties
Les bases du langage	La balise "SCRIPT". Syntaxe du langage. Appel de fonctions, récupération des saisies utilisateurs, passage d'arguments, instructions et opérateurs, chaînes de caractères. Structure d'un programme: contrôle: tests, branchements, boucles
Organisation	La hiérarchie des objets: l'objet "document" Les noms de variables, les constantes, les caractères spéciaux Programmation des objets de base.
Développement avancé	Les ordres standards: les liens, les boutons radios, les zones de saisies (JavaScripts), gestion de la souris Gestion des événements, des cookies. Interfaçage avec Java
Interaction avec les feuilles de style	CSS multiples, héritages multiples, règles de priorité Les blocs de texte, les calques, les couleurs, mises en forme de paragraphes, DIV ... Accès et modification des styles via Javascript.

CMS : mise en œuvre de Joomla

Durée: 3 jours
1470 €

du 19 au 21 Mars
du 11 au 13 Juin

du 3 au 5 Septembre
du 19 au 21 Novembre

Public:

Chefs de projets, développeurs web, administrateurs web

Objectifs:

Comprendre le fonctionnement de Joomla, savoir l'installer et le configurer.

Connaissances préalables nécessaires:

Connaissance des technologies Web.

Programme:

Introduction	Les CMS : Content Management System, fonctionnalités, différents CMS disponibles. Présentation de Joomla. Historique du projet. Les apports et points forts.
Installation et configuration	Prérequis matériel et logiciel. OS supportés. Procédure d'installation et démarrage.
Les utilisateurs	Enregistrement des utilisateurs : différentes possibilités, restriction, formulaire d'enregistrement, création de groupe d'utilisateurs.
Le site web	interface administrateur, changement de style, moteur de recherche. Les templates : layout, couleurs, images, polices; templates installés par défaut, adaptation de templates, gestion avec le template-manager. Les composants et modules Joomla. Travaux pratiques : adaptation de templates, et création de composants simples.
Les contenus	Notions d'articles, de catégories, de sections. Restriction et ouverture des accès. Gestion des articles, menus, contacts, hyperliens, images,... Travaux pratiques : création d'un site par chaque participant sur la base d'un squelette commun. Modifications, gestion des versions des documents.
Migration vers Joomla	Démarche pour migrer un site existant vers le CMS Joomla. Les difficultés rencontrées, les limites.
La sécurité	Les risques inhérents à la publication web Quelques solutions : sauvegardes, mises à jour du produit, sécurisation du serveur

Présentation Web 2.0

Durée: 1 jour

Prix et dates: nous consulter

Public:

Chefs de projet, architectes, et toute personne souhaitant savoir quelles sont les particularités des technologies web 2.0., et les outils disponibles.

Objectifs:

Connaître les principes techniques du web 2.0, et les principales fonctionnalités et produits disponibles.

Connaissances préalables nécessaires:

Une bonne connaissance des principes et techniques d'internet.

Programme:

Introduction	Historique : le Web 2.0 Summit Aspects techniques : interfaces web riches, mécanismes collaboratifs.
Techniques web 2.0	format HTML, feuilles de style CSS, XML, javascript, syndication RSS Ajax avec XMLHttpRequest
Les services Web 2.0	Présentation wiki, galeries, blogs, syndication,...
Wiki	Fonctionnalités Produits disponibles Utilisation, mise en œuvre, choix techniques à réaliser : hébergement, logiciels, administration du wiki, ..
Galeries, blogs	Services disponibles et outils
Flux RSS	Principe de la syndication format RSS, lecteurs de flux, notion d'abonnement, Navigateurs compatibles
Google Apps	Services en ligne : courrier, messagerie instantanée, agenda, ... Principe. Exemples d'applications.

Développement Web avec Ajax

Durée: 2 jours

Prix et dates: nous consulter

Public:

Développeurs web.

Objectifs:

Savoir programmer avec les technologies Ajax.

Connaissances préalables nécessaires:

Il est demandé aux participants de connaître le principe de fonctionnement d'Internet et du Web, le langage HTML. La connaissance d'un langage de programmation est utile.

Programme:

Introduction	Besoin, historique, cadre d'utilisation. Les apports d'AJAX, rôle et limitations. Les technologies mises en œuvre: XML, HTML, CSS. Architecture applicative.
Bases XML :	Nommage des éléments, importations d'espaces de nommage, répartition des définitions. Les entités d'un DTD: les entités de caractères, les attributs (CDATA, ID, ENTITY, ENTITIES, NMTOKEN, NMTOKENS). XML "bien formé", XML "valide". Accès aux éléments et attributs.
Pages HTML	Mode d'appel des scripts. Formulaires HTML: comment créer une zone de saisie? Les différents modes de saisie HTML Les feuilles de style CSS et XML
Développement avec AJAX	Le Modèle AJAX : principe Asynchronous Javascript Xml. Principe de XmlHttpRequest. Requêtes Http/XML vers le serveur. Traitement du résultat avec XML. Exemple avec l'API Google
Mise en œuvre	Instanciation. Gestion des exceptions: try et catch. REQUETE : méthodes et propriétés. Exploitation des requetes GET et des requetes POST
Exploitation des données	Propriétés ".onreadystatechange", ".readystate", ".status" et ".statustext". Utilisation des données de ".responseText" et ".responseXML". Gestion des événements.

Google Apps Utilisation

Durée: 1 jour
460 €

le 20 Février
le 18 Juin

le 24 Septembre
le 26 Novembre

Public:

Toute personne intéressée par la solution Google Apps et souhaitant en comprendre le fonctionnement.

Objectifs:

Connaître les possibilités offertes par Google Apps, et savoir mettre en œuvre les différents outils.

Connaissances préalables nécessaires:

Connaissances des fonctionnalités classiques des outils de bureautique.

Programme:

Présentation	Principe : applications disponibles en ligne en mode collaboratif Fonctionnalités : messagerie, agenda, gestion de documents (traitement de texte, feuilles de calcul, présentations, ..) Apports de la solution Google Apps : pas de données en local, pas d'installation sur les postes clients, gain en coûts de maintenance, ..
gmail	Création d'un compte, stockage des mails, recherche, anti-spams, interopérabilité avec Outlook
Google Agenda	gestion de rendez-vous lien avec la messagerie partage d'agendas accès mobile envoi d'alerte sms
Google Documents	Prérequis : systèmes d'exploitation, navigateurs Formats de fichiers Gestion des accès : sécurité, partage de fichiers, ... Recherche de documents Mise en œuvre de quelques exemples de textes, feuilles de calcul, présentations.
Google Groupes	Création de listes de diffusion, de groupes. Partage de documents
Google Sites	Création de pages Web pour des équipes projets Centralisation des documents, feuilles de calcul, ...
Sécurité	Les mécanismes de sécurité : Anti-spam , le service postini Sécurisation des données partagées

Google App Engine

Durée: 3 jours
1570 € HT

du 21 au 23 Février
du 19 au 21 Juin

du 25 au 27 Septembre
du 27 au 29 Novembre

Public:

Développeurs, chefs de projets souhaitant comprendre les principes de développement d'applications web avec Google App Engine.

Objectifs:

Connaître l'architecture Google App Engine, et savoir mettre en œuvre les différents composants pour le développement d'applications.

Connaissances préalables nécessaires:

Connaissance d'un langage de programmation et des principes de base du web.

Programme:

Introduction	Présentation de Google App Engine Principe : développement d'applications web en utilisant l'infrastructure Google Langages de programmation : Python, Java Positionnement par rapport aux outils concurrents comme AWS.
Architecture	Navigateur, google's load balancer, clusters google, RPC pour l'accès aux données, Base de données datastore, Gestion des utilisateurs : authentification google, Environnement d'exécution : sandbox Les limites : par rapport aux fichiers, sockets, threads, au temps d'exécution, aux appels en C Travaux pratiques : première application avec Google App Engine : "Hello world"
Outils de développement	le SDK pour chaque langage Les APIs d'accès aux services Google : gestion de la mémoire, envoi de mails, utilisation du format XMPP (messagerie GoogleTalk), programmation de tâches en mode différé (cron) Travaux pratiques : développement d'une application avec accès à la base de données, et programmation d'envoi de mail à intervalles réguliers
Limites	Quotas sur les nombres de requêtes, la bande passante, le temps processeur, le nombre d'accès à la base de données, le volume de données

Développement Web avec GWT

Durée: 3 jours
1495 € HT

du 17 au 19 Janvier
du 4 au 6 Avril

du 4 au 6 Juillet
du 3 au 5 Octobre
du 5 au 7 Décembre

Public:

Développeurs Web et toute personne souhaitant comprendre le fonctionnement de GWT.

Objectifs:

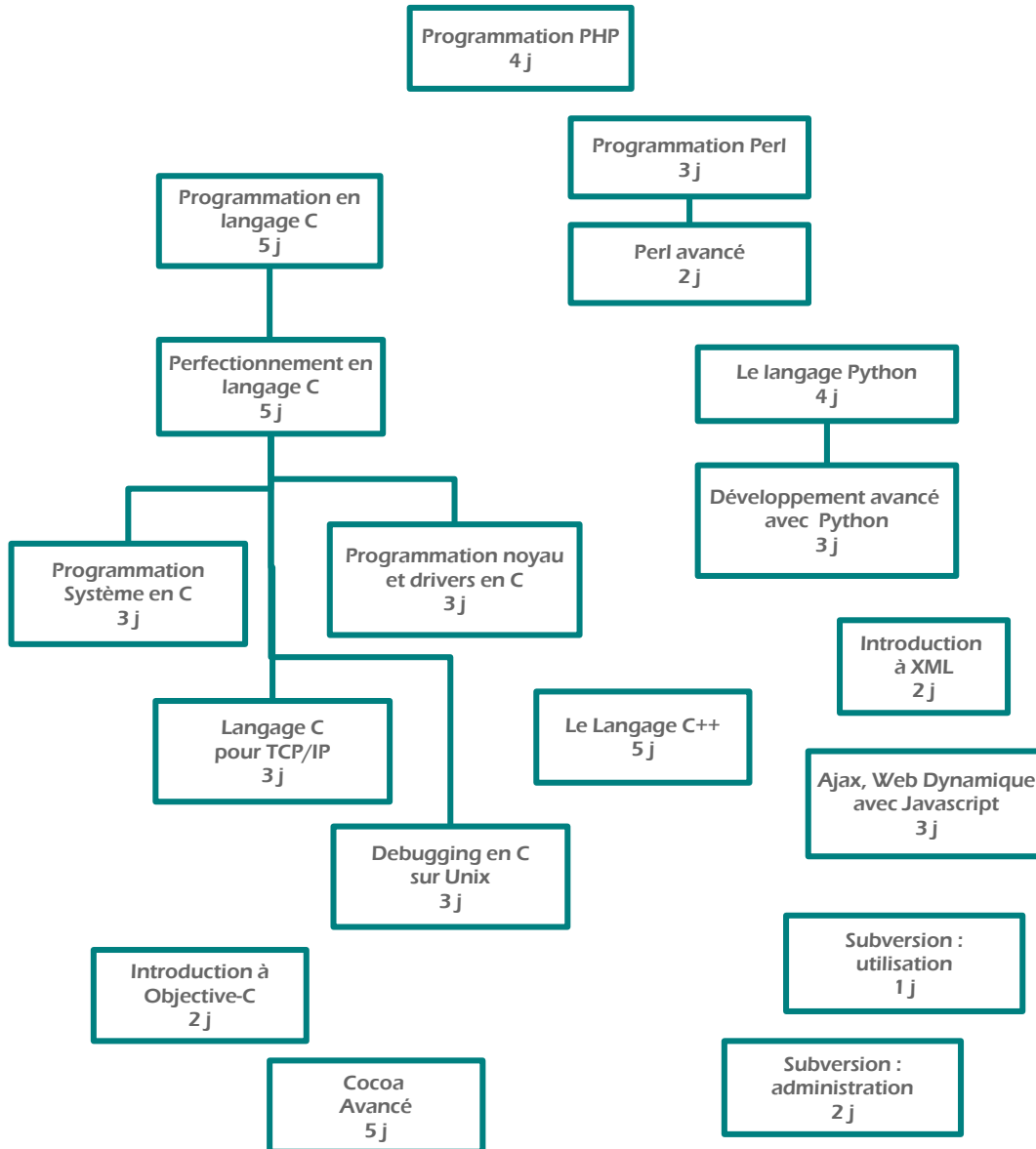
Comprendre le principe de GWT, savoir le mettre en œuvre pour développer des applications Web.

Connaissances préalables nécessaires:

Une bonne connaissance de Java est nécessaire, et des contraintes du développement web.

Programme:

Introduction	Présentation de Google Web Toolkit Historique, licence Apache 2.0 Apports de GWT par rapport à Java Principe : développement réalisé en Java, traduction en javascript par le compilateur
Composants	Elements fournis par le GWT SDK : Le compilateur Javascript Le navigateur pour les tests L'émulateur JRE Les widgets : composants graphiques Mise en œuvre : réalisation d'une première application
Widgets	Composants graphiques Exemples de réalisation avec les composants fournis en standard avec le SDK Bibliothèques complémentaires : EXT-GWT GWT Components Library GWT Widget Library GWTiger Dojo, ...
Extensions	GWT RPC : requêtes à un serveur distant Google App Engine : utilisation des API Google
Outils de développement	Présentation des différents IDE : GWT Designer (plugin Eclipse), gwt Developer, ... Tests unitaires avec JUnit Speed Tracer : extension du navigateur Chrome pour optimiser les applications Web



XML Développement

Durée: 3 jours

Prix et dates: nous consulter

Public:

Toute personne souhaitant acquérir des notions de base sur le langage XML.

Objectifs:

Maîtriser les outils et techniques XML basiques nécessaires au déploiement d'une application XML.

Connaissances préalables nécessaires:

Une connaissance de base des concepts de l'internet et du langage HTML et d'un langage de programmation sont nécessaires pour suivre ce cours.

Programme:

Introduction	Principales utilisations. Concepts élémentaires XML : historique : HTML, le web et l'héritage de SGML. Le standard XML. Une architecture faiblement couplée basée sur XML. Les standards associés : XSL, XPath, XQuery. Gestion de documents.
Syntaxe et sémantique de XML	Balisage. Eléments, attributs, entités, codage. Syntaxe et usage des espaces de noms.
DTD et schémas XML	Syntaxe des DTD et des schémas XML Règles et qualité de l'écriture de DTD. Limites des DTD. Modularité apportée par la combinaison des NameSpaces et par XMLSchéma. Ecriture de schémas XML à l'échelle de l'entreprise.
XML dans l'architecture	Les architectures Web : SVG, XUL, ... XML dans les architectures distribuées. Appels à distance : SOAP.
XSTL et standards connexes.	Transformation avec XSLT et XPath. Présentation de XSLFO.
Introduction à DOM	Document Object Model. Représentation mémoire des documents XML sous forme d'arbre DOM. Interface du DOM.
Persistance des données XML	Diverses solutions de stockage. Langage de requêtes XML. Les problématiques de stockage.

Programmation en langage C

Durée: 5 jours
1960 €

du 27 Février au 2 Mars
du 21 au 25 Mai

du 27 au 31 Août
du 15 au 19 Octobre
du 10 au 14 Décembre

Public:

Tout développeur souhaitant apprendre le langage C

Objectifs:

Connaître et maîtriser les concepts de base du langage C. Savoir écrire des programmes simples et acquérir des méthodes de programmation.

Connaissances préalables nécessaires:

Connaissance d'un langage de programmation.

Programme:

Le C	Présentation du langage C. Avantages et inconvénients. Architecture, syntaxe. Structure d'un programme C. Compilation.
Contrôle de programme	Instructions de contrôle. Boucles for et while. Les tests. Les branchements avec break, continue, return, exit, goto et switch.
Manipulation de données	Types de données. Les variables, tableaux, chaînes de caractères. Déclarations de variables. Utilisation des types: variables entières, réelles, structurées. Les types primitifs : char, short, int, long, float, double. Type statique, registres. Manipulation des tableaux : initialisation, accès aux tableaux. Traitement des chaînes de caractères : initialisation, saisie, accès Notions sur les variables externes. Conversion, règles de portée.
Fonctions	Fonctions de base: affichage et lecture des données. Entrées/sorties formatées : options d'affichage des caractères, entiers, ... Fonctions spécifiques aux chaînes de caractères : strcpy, strcat, strchr,strupr, strlen. Fonctions personnalisées: définitions, règles de fonctionnement, récursivité.
Programmation structurée	Notion de structure. Les unions, champs binaires, types énumérés. Définition, déclaration, utilisation de structures. Exemples d'utilisation des champs binaires et mise en œuvre. Déclaration d'énumération avec enum. Etude d'exemples.

Programmation en langage C

Les opérateurs.

opérateurs de calcul, simplification d'écriture, opérateurs de décalage (>> et <<), et binaires(| ^).

Erreurs de conversion implicite.

Opérateurs de comparaison (== <= >= ? ...) et opérateurs logiques (! ||)

Priorités des opérateurs.

Librairies

Introduction à la notion de librairie. La librairie standard. les fichiers inclus.

Introduction aux différentes phases de compilation, édition de liens.

Allocation dynamique

Présentation, les pointeurs.

Principe de l'allocation dynamique.

Applications. Exemples des listes chaînées et arbres binaires.

Pointeurs sur les fonctions.

Perfectionnement en langage C

Durée: 5 jours
2020 €

du 19 au 23 Mars
du 4 au 8 Juin

du 10 au 14 Septembre
du 22 au 26 Octobre
du 17 au 21 Décembre

Public:

Toute personne amenée à programmer, à superviser ou à modifier des logiciels écrits en langage C.

Objectifs:

Compléter des connaissances en langage C par une formation approfondie sur les mécanismes fondamentaux de fonctionnement.

Connaissances préalables nécessaires:

Il est demandé aux participants de connaître les structures et fonctions de base du langage C.

Programme:

Rappels	Les variables, type statique. Variables statiques et variables registres. Conversions.
Structures	Présentation, intérêt des structures. syntaxe de définition, déclaration et d'accès aux éléments des structures. Exemples: copie de structures Structures avancées, unions, tableaux, champs binaires, drapeaux.
Fonctions de saisie, affichage	Options avancées de printf. Mise en forme paramétrée. Saisie avancée. Rappel du principe : décomposition du flux d'entrée. Les types 'ensemble'.
Les opérateurs	Rappels sur les opérateurs de calculs. Opérateurs logiques, opérateurs binaires. Travaux pratiques : mise en œuvre des opérateurs de décalage. Priorité des opérateurs.
Fonctions	Pointeurs sur les fonctions. Applications aux interpréteurs.
Allocation dynamique	Principe d'allocation mémoire. Syntaxe de malloc et free. Travaux pratiques de mise en œuvre. Fonctions avancées (calloc et realloc) : intérêt et applications.
Pratique	Les listes chaînées, les arbres binaires. Applications à l'organisation des données.

Perfectionnement en langage C

Techniques de programmation

Les phases de compilation :
précompilation , assemblage, édition de liens.
Définition de constantes.
Contrôle de compilation.
Les macro-instructions.
Conventions de nommage.
Comparaison avec les fonctions.
Les fichiers inclus : #include.

Bibliothèques

Méthode, syntaxe
Les bibliothèques standards : libc.a, libm.a, libcur.a
Fonctions disponibles dans la bibliothèque mathématique.

Les entrées/sorties

Mécanisme de stockage des fichiers.
Méthode d'accès, les descripteurs de fichiers.
Fonctions open/close.
Travaux pratiques :
écriture d'une fonctions permettant de tester l'existence d'un fichier.
Fonctions read/write.
Mise en œuvre avec lecture/écriture de structures.
Modes d'ouvertures spécifiques :
avec positionnement dans le fichier, avec création du fichier, ...
Options : O_TRUNC, o_SYNC, O_NDELAY.
Le type FILE : mise en œuvre de fprintf, fscanf, fgets, fputs.

Programmation système en C sur Unix/Linux

Durée: 3 jours
1430 €

du 13 au 15 Février
du 26 au 28 Mars
du 9 au 11 Mai

du 1er au 3 Octobre
du 10 au 12 Décembre

Public:

Toute personne amenée à programmer, à superviser ou à modifier des logiciels écrits en langage C et liés au système d'exploitation.

Objectifs:

Compléter des connaissances en langage C par une formation approfondie sur les mécanismes d'accès au système d'exploitation. L'accent sera particulièrement sur les fichiers, pointeurs, allocations de mémoire, communications et les bibliothèques systèmes.

Connaissances préalables nécessaires:

Il est demandé aux participants de bien connaître les structures et fonctions de base du langage C.

Programme:

Rappels

Architecture d'un programme écrit en C.
Phases de compilation.

Gestion de la mémoire

Rappel sur l'organisation de la mémoire.
L'adressage par les pointeurs.
Les opérateurs `&` et `*`.
Les pointeurs et les arguments de fonctions.
Les calculs d'adresses.
Les fonctions d'allocation `malloc` et `free`,
et les appels systèmes: `sbrk`, `realloc`.
Travaux pratiques :
écriture d'un allocateur de mémoire.

Programmation système en C sur Unix/Linux

Communications inter-processus.

Les différentes méthodes : pipes, fifo, signaux, files de messages.

Signaux et interruptions : les principaux signaux.

Travaux pratiques :

émission d'un signal avec `kill()`, réception du signal par `signal()`.

Sémaphores et appels concurrents :

principe de fonctionnement des sémaphores.

Travaux pratiques :

mise en œuvre avec `semget`, `semctl`, `semop`.

Segments de mémoires partagées :

Définitions de constantes et structures,

Travaux pratiques :

création d'un segment de mémoire partagée avec `shmget`,
attachement, détachement d'un segment avec `shmat`, `shmdt`.

Files de messages :

constantes et structures nécessaires pour la manipulation des files de messages.

Travaux pratiques :

mise en œuvre de la primitive `msgget()`,

gestion des files de messages (consultation, modification, suppression) avec `msgctl()`

Envoi d'un message à une file : `msgsnd()`.

Segments partagés:

Définition d'un segment de mémoire partagé. Description et mise en œuvre des appels systèmes `shmat()`, `shmget()`.

Utilisation de sémaphores pour la gestion des accès concurrents au segment.

Sockets BSD:

Mise en œuvre des prises réseaux pour la communication interprocessus.

Exemple avec des liens locaux. Extension aux liens distants.

Communications inter-machines.

Programmation système en C sur Unix/Linux

Les processus et la parallélisation

Création de processus.
Définition et mise en œuvre des primitives `fork()`, `clone()`, `setsid()`.
Limites d'utilisation. Introduction aux threads.
Les threads. La norme et les implémentations.
L'implémentation Posix: NPTL.
Cycle de vie des threads: création, destruction.
Synchronisation entre threads, détachement du processus principal, attente de fin d'exécution.
Attributs des threads.
Gestion de la mémoire consommée, gestion de la pile de données.
Gestion des accès concurrents, principe de l'exclusion mutuelle.
Travaux pratiques :
mise en œuvre des mutex.
Coopération de traitements entre threads.
Mise en œuvre des conditions variables. Gestion des signaux dans un thread.
Ordonnancement de threads.

Programmation noyau et drivers en C sur Linux

Durée: 3 jours
1480 €

du 13 au 15 Février
du 2 au 4 Mai

du 27 au 29 Août
du 12 au 14 Novembre

Public:

Tout développeur souhaitant gérer les modules du noyau, ou en programmer de nouveaux.

Objectifs:

Comprendre le fonctionnement des modules dans le noyau, et savoir concevoir des drivers.

Connaissances préalables nécessaires:

Connaissance du système d'exploitation Linux, maîtrise de la programmation en langage C.

Programme:

Architecture	Architecture d'un système Unix. Mode user, mode kernel. Logs. Notion de pilotes/drivers. Architecture d'un système Linux. Notion de modules. Les distributions: desktop, embarquées (openWRT). Gestion des modules: ajout, suppression, paramétrage. Travaux pratiques : ajout d'un module simple sur une distribution standard. Présentation de la busybox pour les distributions embarquées.
Compilation noyau	Compilation d'un noyau pur officiel. Description de la chaîne de compilation. Options de compilations. Mise en place du nouveau noyau. Travaux pratiques : Modification d'un module pilote. Ajout au noyau précédent. Création d'un module de base.
Programmation de modules	Principes fondamentaux : timers et alarmes, journalisation, échanges de données kernel-mode/user-mode, interactions dynamiques avec un module, passage de paramètres, gestion des tâches task_struct, allocation mémoire kmalloc, verrouillage du noyau en cas d'accès concurrents (lock_kernel).

Langage C pour TCP/IP

Durée: 2 jours

Prix et dates: nous consulter

Public:

Toute personne désireuse d'approfondir ses connaissances sur les réseaux TCP/IP.

Objectifs:

Comprendre les mécanismes fondamentaux de TCP/IP au travers d'une programmation d'outils en langage C: traceroute, ping et analyseur de trames.

Connaissances préalables nécessaires:

Il est demandé aux participants de connaître les bases de TCP/IP et du langage C.

Programme:

Rappels	Rappels sur le fonctionnement de TCP/IP et de son utilisation par le système d'exploitation
Lecture IP	Programmation d'une interface d'accès à la couche réseau. Lecture de trames réseau. Décomposition et étude d'un analyseur de trames. Modification du programme.
Ecriture IP	Envoi d'une trame ICMP sur le réseau. Ecriture d'un ping simplifié. Ecriture de l'utilitaire traceroute.

Debugging en C sur Unix

Durée: 1 jour

Prix et dates: nous consulter

Public:

Développeurs en langage C. Exploitants et administrateurs Unix.

Objectifs:

Maîtriser les différentes phases de la compilation d'un programme écrit en C sur une machine Unix. Connaître les outils d'exploitation permettant de suivre une exécution, d'en analyser les performances.

Connaissances préalables nécessaires:

Il est demandé aux participants de connaître les bases du système Unix et du langage C.

Programme:

Compilation	Les différents langages de programmation. Comment un programme utilise les appels systèmes. Les différentes phases de compilation Reconnaissance des types de fichiers. Extraction du langage depuis un exécutable binaire . Exploitation des fichiers objets: objdump, objcopy, readelf, size, strings
Bibliothèques	Bibliothèques standards. Fonctions et appels systèmes. Commandes d'exploitation des bibliothèques: ar, ranlib. Gestion des noms et symboles: nm, strip
Projet	Outils de développement d'un projet: lint, make, cb. Découpage d'un programme en modules. Installation par make. Structure d'un Makefile.
Debugging	Les debuggers: dbx, dbg, adb. Exploitation des fichiers core. Création d'un dump mémoire. Informations nécessaires aux équipes de développement pour corriger les erreurs de programmation.

Le langage C++

Durée: 5 jours

Prix et dates: nous consulter

Public:

Toute personne amenée à programmer, à superviser ou à modifier des logiciels écrits en langage C++ et liés au système d'exploitation.

Objectifs:

Apprendre la programmation objet au travers du langage C++. Maîtriser l'analyse et la conception de programmes orientés objets.

Connaissances préalables nécessaires:

Il est demandé aux participants de connaître les bases d'un langage structuré (Pascal, C, ou autre).

Programme:

POO	Définitions: besoin, cadre d'utilisation, termes utilisés, organisation, historique.
Base	Architecture d'un programme écrit en C++; variables simples; affichage: cout et cin; fonctions; surcharges. Structure d'un programme: boucles, tests, branchements
Approche objets	Structures, classes, constructeurs. Fonctions membres, fonctions inline, surcharges des membres, accès: public, private, protected. Fonctions amies. Imbrication d'objets, dérivation, héritage. Fonctions virtuelles, polymorphisme. Classes abstraites.
Données	Les différents types de variables et attributs (tableaux, données statiques, constantes), règles de portées. Autoréférence. Allocation dynamique: new et delete. Tableaux d'objets
Compléments sur les opérateurs	Opérateurs de calcul, comparaison, binaires. Surcharge d'opérateurs. Surcharge de << et >>. Fonction-opérateur amie.
Système	Gestion des erreurs. Les exceptions. Contrôle des formats de données, redirections, traitement des fichiers, stream. Les manipulateurs, analyses de chaînes.
Compilation	Préprocesseur, contrôle de compilation, macros instructions. Patrons de fonctions, patrons de classes. Fichiers inclus.

Introduction à Objective-C

Durée: 3 jours
1360 €

du 6 au 8 Février
du 14 au 16 Mai

du 10 au 12 Septembre
du 12 au 14 Novembre

Public:

Tout développeur souhaitant programmer en Objective-C, en particulier pour aborder le développement Cocoa pour les applications iPhone, iPad et Mac OS X.

Objectifs:

Comprendre les principes de base de la programmation avec Objective-C, et les bases de la programmation objet.

Connaissances préalables nécessaires:

Des notions de programmation sont nécessaires, quel que soit le langage. La pratique du C est un plus mais n'est pas requise.

Programme:

Introduction	<ul style="list-style-type: none"> Historique du langage Objective-C Positionnement par rapport au C, C++, Smalltalk, Java Caractéristiques du langage Domaines d'utilisation Outils de développement : le compilateur gcc, l'IDE Xcode Présentation du framework Cocoa
Bases	<ul style="list-style-type: none"> Programmation orientée objet La notion de classe, l'héritage Appel de méthode : le message Format du message Encapsulation, polymorphisme La classe NSObject
Mise en œuvre	<ul style="list-style-type: none"> Définition d'une classe Gestion de la mémoire Allocation, désallocation Instanciation d'un objet : alloc, init retain, release, autorelease Travaux pratiques

Programmation Cocoa avancée Mac et iPhone

Durée: 5 jours
2150 € HT

du 5 au 9 Mars
du 11 au 15 Juin

du 1er au 5 Octobre
du 3 au 7 Décembre

Public:

Tout développeur maîtrisant les bases de Cocoa et Objective-C.

Objectifs:

Affiner ses connaissances en Objective-C, explorer le framework Cocoa, améliorer ses pratiques de développement. Ce cours cible autant l'iPhone que le développement d'applications Mac OS X.

Connaissances préalables nécessaires:

Une pratique déjà solide de la programmation avec Cocoa est indispensable.

Programme:

Les bases du runtime	<ul style="list-style-type: none"> Le runtime Objective-C (catégories, protocoles, ajouts d'Objective C 2.0) Les mécanismes de délégation et de notification Les blocks Comprendre et utiliser le Key-Value Coding / Key-Value Observing (KVC/KVO) L'utilisation du KVC/KVO dans Cocoa et Cocoa Touch Travaux pratiques KVC/KVO
Les Cocoa Bindings	<ul style="list-style-type: none"> Les Cocoa Bindings, une exploitation complète du KVO/KVC Utiliser les Cocoa Bindings depuis Interface Builder ou depuis Xcode Les contrôleurs (NSObjectController, NSArrayController) TP Cocoa Bindings : créer ses propres contrôles "bindés"
La persistance avancée	<ul style="list-style-type: none"> Qu'est ce qu'est Core Data et ce qu'il n'est pas... Présentation de l'environnement Core Data (schéma entité-relation) L'intégration avec Xcode, Interface Builder, KVC/KVO et les Cocoa Bindings La pile Core Data (NSManagedObjectContext, NSManagedObject, NSPersistentStoreCoordinator, les Persistent Stores) Les Fetch Requests, les Predicats Les performances TP Core Data

Programmation Cocoa avancée Mac et iPhone

Créer des interfaces

- La migration de schéma
- Relation entre stores
- Threading et Core Data / Les NSOperations
- TP Core Data
- Présentation de Core Animation
- La notion de Layer
- Les propriétés auto-animées
- Le Timing
- La création de ses propres animations
- TP Core Animation

Les outils avancés

- Exploiter le Runtime Objective-C
- Method Swizzling
- Debugging avancé avec GDB
- Les outils d'analyse de code et de performance (Clang, Instruments, Shark...)

Subversion utilisateur

Durée: 1 jour
405 €

le 26 Mars
le 14 Mai

le 10 Septembre
le 19 Novembre

Public:

Chefs de projet, développeurs, ingénieurs logiciels, architectes d'application, équipes de qualification

Objectifs:

S'approprier les principes de travail sur un référentiel partagé pour la gestion de version

Connaissances préalables nécessaires:

Connaissance en gestion de projet, développement.

Programme:

Introduction	Le besoin : gestion de version, développement projet. Historique de Subversion. Caractéristiques.
Les fonctionnalités	Contrôle des versions Gestion des fichiers, répertoires et des modifications Historique des changements Fonctionnement en réseau, partage des fichiers
Les concepts	Définitions : référentiels, branches, étiquettes, révision
Les outils clients	Plugin Eclipse, ant le mode commande : svn

Subversion Administrateur

Durée: 2 jours
815 €

du 27 au 28 Mars
du 15 au 16 Mai

du 11 au 12 Septembre
du 20 au 21 Novembre

Public:

Administrateurs de l'outil subversion

Objectifs:

S'approprier les principes de travail sur un référentiel partagé pour la gestion de version.

Connaissances préalables nécessaires:

Connaissance en gestion de projet, bonne connaissance de l'administration du système Linux

Programme:

Introduction	La Gestion de Configuration Logicielle - Gestion de version, développement projet - Historique de Subversion
Les caractéristiques de Subversion	Architecture client/serveur - Le Référentiel - Interfaces d'accès: mode commande, interfaces graphiques
Fonctionnalités	- Contrôle des versions - Gestion des fichiers, répertoires et des modifications - Historique des changements - Fonctionnement en réseau, partage des fichiers
Concepts et terminologie	Référentiels - Branches - Etiquettes - Copies - Révision
Travail en équipe avec Subversion	Les outils client: Plugin Eclipse, ant, le mode commande, Tortoise - Récupérer les sources - Création d'une branche - Les fusions - Gestion des branches - Valider/Annuler des modifications - Résolution des conflits - Gestion des verrous - Création d'une étiquette - Analyse de l'historique - Bonnes pratiques
Configuration du serveur	Choix du serveur - Serveur svnserv - svnserv avec ssh - Serveur http Apache - Configuration réseau - Gérer plusieurs méthodes d'accès - Gestion de la sécurité : utilisateurs, groupes, droits d'accès

Subversion Administrateur

Administration du référentiel

Stratégie de déploiement

- Mode de stockage : système de fichier ou base de données
- Création et configuration du référentiel
- Sauvegarde et restauration
- Surveillance et optimisation
- Les outils svnlook et svn admin
- Ajout de projets
- Migration
- Gestion de l'historique

Programmation PHP

Durée: 4 jours
1700 €

du 2 au 5 Avril
du 9 au 12 Juillet

du 22 au 25 Octobre
du 26 au 29 Novembre

Public:

Développeurs, concepteurs/réalisateurs, webmestres, chefs de projet Web, architectes techniques.

Objectifs:

Maîtriser la syntaxe du langage. Développer des applications Web dynamiques en PHP. Comprendre l'orientation web de PHP. Gérer des formulaires et les accès aux données. Gérer les utilisateurs de l'application.

Connaissances préalables nécessaires:

Connaissances de base web et html, bases de données Sql,. Pratique de la programmation objet. Connaissances des concepts des architectures multi-tiers.

Programme:

Introduction	Vue d'ensemble de PHP. Structure de base d'une page PHP. Règles de nommage. Installation de PHP (mécanismes d'installation).
Syntaxe de base du langage	Variables, constantes, types, tableaux. Fonctions. OPérateurs. Gestion des chaînes de caractères, des dates, des nombres.
Les structures de contrôle	Les contrôles conditionnels : if et switch. Les contrôles itératifs : while, do while, for, continue. Inclusion de fichiers : fonctionnement et utilisation. Comment sortir d'une structure de contrôle. Interruption d'un script.
Introduction aux concepts objet	Concept Objet. Fonctions et classes. Gestion des exceptions.
Gestion des formulaires	Récupérer les informations du formulaire. Construction de l'interface utilisateur. Contrôles. Gestion des codes et pages d'erreur. Les filtres.
Connectivité avec les SGBDR	Les principaux SGBDR. SOLite et MySQL. Connexion et déconnexion. Lire et mettre à jour les données. Gérer les erreurs. Requêtes préparées.
La gestion des sessions.	Fonctionnement des sessions. Débuter une session et assigner des variables. Suppression des variables de session. Destruction d'une session. Les cookies.

Programmation PHP

PHP et le système de fichiers.

La sécurité sur les fichiers.

Les fonctions de manipulation d'images.

Création de graphiques (JPGRAPH, ARTICHOW).

Génération de fichiers Excel.

Les concepts de sécurité, l'authentification.

Initiation au langage Pascal

Durée: 5 jours

Prix et dates: nous consulter

Public:

Tout programmeur en langage Pascal.

Objectifs:

Connaître et maîtriser les concepts de base du langage Pascal. Savoir écrire des programmes simples et acquérir des méthodes de programmation.

Connaissances préalables nécessaires:

Aucune connaissance préalable n'est nécessaire pour suivre ce cours

Programme:

Le Pascal	Présentation du langage Pascal; Architecture, syntaxe. Structure d'un programme Pascal. Compilation.
Contrôle de programme	Instructions de contrôle. Boucles. Les tests. Branchements.
Variables	Déclarations de variables; utilisation des types: variables entières, réelles, structurées, enregistrements. Notions sur les variables externes.
Fonctions et procédures	Fonctions de base: affichage et lecture des données. Procédures. Passages d'arguments par valeur, par référence. Fonctions personnalisées: définitions, règles de fonctionnement, récursivité.
Manipulation de données	Tableaux, chaînes de caractères Présentation. Enregistrement. Pointeurs. Conversions, règles de portée. Opérateurs. Affichage et saisie avancés.
Librairies	Introduction à la notion de librairie. La librairie standard. les fichiers inclus. Introduction aux différentes phases de compilation, édition de liens.

Programmation Perl

Durée: 3 jours
1180 €

du 13 au 15 Février
du 2 au 4 Avril
du 9 au 11 Juillet

du 8 au 10 Octobre
du 17 au 19 Décembre

Public:

Tout développeur souhaitant acquérir les bases de la programmation en Perl.

Objectifs:

Comprendre les principes de base de Perl, connaître la syntaxe de base.

Connaissances préalables nécessaires:

La connaissance d'un langage de programmation sera appréciée.

Programme:

Introduction	Présentation de Perl : caractéristiques, positionnement par rapport à d'autres langages (C, Java, Shell, PHP) Plate-formes d'utilisation Installation de Perl Syntaxe Exécution d'un programme Perl Quelques règles utiles
Les bases	syntaxe, structure des programmes Les données Déclaration de variables Opérateurs de liaison, de décalage, ... Manipulation de scalaires
Les fonctions :	Principe, appel d'une fonction, passage des arguments, Renvoi d'une valeur Appel d'une fonction Visibilité des variables Les références : définition, création de références, références symboliques Prototype Fonctions prédéfinies
Structure d'un script Perl	Les tests, boucles Opérateurs de contrôle, modificateurs
Expressions régulières et variables spéciales	Les expressions régulières, les méta-caractères Motifs particuliers Quantificateurs et classes Opérateur de substitution de motif, de lettre Variables spéciales Constantes particulières
Tableaux	Manipulation de tableaux, hachage, sauvegarde des tableaux

Programmation Perl

Fichiers

Entrées-sorties standards
Manipulation de fichiers
Les redirections
Opérateurs de test de fichiers

Exécution

différentes méthodes d'exécution
compilation , exécution

Perl avancé

Durée: 2 jours
935 €

du 5 au 6 Avril
du 12 au 13 Juillet

du 11 au 12 Octobre
du 20 au 21 Décembre

Public:

Tout développeur souhaitant acquérir les bases de la programmation en Perl.

Objectifs:

Maîtriser la programmation en Perl, particulièrement les modules et la programmation objet.

Connaissances préalables nécessaires:

La connaissance des bases du langage Perl est indispensable.

Programme:

L'environnement	opérations sur les fichiers communication inter-processus
Structures complexes	Références anonymes Tableaux de tableaux Hash de hash Opérateur ref() Opérateur Dumper
Programmation objet	Les constructeurs, manipulation des objets, Méthodes, notion d'héritage Test de classes
Les modules	Définitions. CPAN : les modules archivés Installation et utilisation d'un module Blocs BEGIN et END
Module Perl/TK	Présentation Principe des widgets Fenêtre principale Widgets de base
Module DBU	Présentation et installation Exemples d'utilisation

Le langage Python

Durée: 4 jours
1905 €

du 12 au 15 Mars
du 29 Mai au 1er Juin

du 17 au 20 Septembre
du 6 au 9 Novembre

Public:

Tout développeur souhaitant acquérir les bases de la programmation en Python.

Objectifs:

Connaître les possibilités du langage Python, maîtriser les techniques de programmation et apprendre les bonnes pratiques de développement.

Connaissances préalables nécessaires:

La connaissance d'un langage de programmation sera appréciée.

Programme:

Introduction	Présentation Python : caractéristiques, positionnement par rapport à d'autres langages. Installation. Utilisation de l'interpréteur. Premier programme en Python.
Les bases	Principaux types de données : nombres, booléens, chaînes de caractères. Déclaration de variable, typage dynamique, mots clés réservés. Les opérateurs : priorité, associativité, opérateurs d'affectation, logiques, de comparaison. Quelques fonctions utiles : print(), input(). Structures conditionnelles : if et elif. Boucle while, mots clés break et continue.
Chaînes de caractères et listes	Définition et manipulation de chaînes de caractères. Le type séquence. Les listes : définition, accès à un élément. Les références. Les tuples. Manipulation de listes : mot clé del, fonctions list et range; parcours d'une liste.
Les fonctions	Présentation, déclaration et appel d'une fonction. Portée des variables, mot clé global. Passage d'arguments, les arguments par défaut. La récursivité. Les fonctions Lambda : définition, utilisation Fonctions intégrées
Programmation Objet	Rappels sur la programmation objet Les classes en Python Constructeurs, attributs privés, méthodes, héritage

Le langage Python

Les fichiers	Méthodes d'accès aux fichiers : ouverture (accès en mode lecture, écriture, ajout, ...), fermeture, le mot clé with; lecture dans un fichier
Les types de données complexes	Les listes de listes. Une liste de tuples. Les dictionnaires. Parcours d'un dictionnaire. Les méthodes update(), clear(), pop(), del(), values, keys()
Modules et expressions régulières	Les modules : définition, la fonction help() Importer des fonctions Créer ses propres modules Expressions régulières, les caractères spéciaux. Groupes et classes de caractères. Le module 're'.
Gestion des exceptions	Principe, exemples d'exceptions. Mots clés try et except, else et finally Les assertions

Développement avancé avec Python

Durée: 3 jours
1450 €

du 26 au 28 Mars
du 11 au 13 Juin

du 1er au 3 Octobre
du 3 au 5 Décembre

Public:

Les développeurs en Python.

Objectifs:

Maîtriser les fonctionnalités comme la gestion des graphiques, des bases de données, les liens avec les langages C et Java, le développement d'applications Web, et l'utilisation de framework comme Django.

Connaissances préalables nécessaires:

Connaissance de base de Python.

Programme:

Programmation graphique	Différentes solutions : PyOt, Tkinter, PyGTK, wxWidgets, Caractéristiques de chaque solution. Travaux pratiques avec le module Tkinter. Création d'objets (fenêtres, boutons, ...), appel des méthodes associées (grid(), pack(), ...)
Gestion des bases de données	Les différentes méthodes : création d'une base avec les modules Gadfly, interfaçage MySQL avec MySQLdb, accès à postgresSQL avec les modules PyGreSQL ou Psycopg, ... Travaux pratiques avec une base postgresSQL et le module Psycopg.
Développement web	Présentation et comparaison des frameworks et langages de template. Les frameworks disponibles : CherryPy, Paste, CPS, Django, TurboGears, Pylons, ... Les langages de templates : Myghty, Python Server Pages, Cheetah, Zope, Mise en œuvre de Django. Installation, configuration initiale : création d'un projet, serveur de développement Django, configuration des accès aux bases de données, ...), création et activation de modèles, développement d'une application simple.

Développement avancé avec Python

Liens avec les langages C et Java

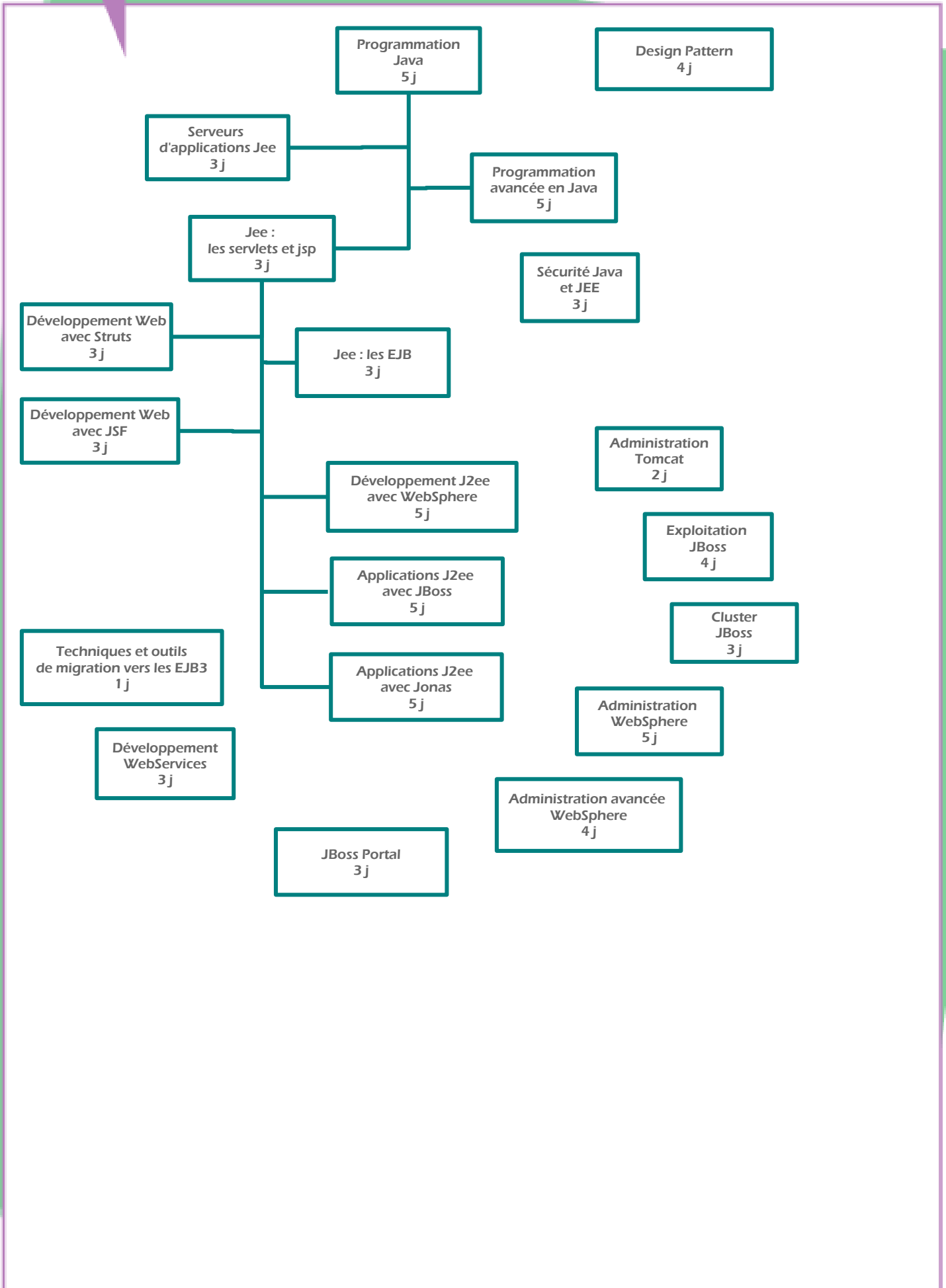
Les besoins :
accès à des programmes en C ou à des classes Java depuis Python,
bénéficier des avantages de Python depuis des programmes Java,

..

Les outils :

Jython, Jepp (Java Embedded Python), JPE (Java Python Extension),
Boost.Python, ...

Filières Langages Java et JEE



Programmation Java

Durée: 4 jours
1700 €

du 24 au 27 Janvier
du 13 au 16 Mars
du 22 au 25 Mai
du 19 au 22 Juin

du 28 au 31 Août
du 9 au 12 Octobre
du 20 au 23 Novembre

Public:

Développeurs Java.

Objectifs:

Connaître les outils nécessaires à la programmation et l'exécution de programmes en Java, et acquérir les notions de base de programmation objet en Java.

Connaissances préalables nécessaires:

Connaissance d'un langage de programmation.

Programme:

Présentation	Utilisations de Java: sur le Web, en application lourde, en application distribuée, dans une architecture JEE, dans les cartes à puce, systèmes embarqués. Définitions: JDK, JRE.
Conception de base	Programmation objet, les réutilisables. Syntaxe d'une classe. Principe de l'encapsulation. Constructeurs, attributs et méthodes. Accesseurs. Différence entre objet et classe. Instanciation. Conventions de nommage.
Notions de base	Syntaxe du langage: Variables, références; Instructions et opérateurs, Tableaux, Structure d'un programme. Contrôle: tests, branchements, boucles
Classes de base	La classe System. Les chaînes de caractères. Date et heure. Les classes Integer, Double.
Exécution	Exceptions de base. La structure try/catch. Multi-tâches (threads). Les groupes de Thread. Les horloges. Synchronisation entre tâches.
Fichiers	Les classes principales d'entrées/sorties. Accès à un fichier, copie d'un fichier. Méthodes d'accès aux fichiers
Graphique	Les classes graphiques, gestion d'images, présentation, animation. Gestion des évènements. Gestion de la souris. Applets.
Développement	Abstraction et interfaces. Création et utilisation d'interface. Exceptions, pile d'exécution. Création et utilisation d'exceptions.

Programmation Java

Outil de développement

Eclipse : environnement de développement extensible par plugins.

Principes de fonctionnement, installation et premiers pas avec Eclipse.

JDT : Java Development Tools, eclipse Europa

Programmation avancée en Java

Durée: 5 jours
2030 €

du 23 au 27 Janvier
du 19 au 23 Mars
du 4 au 8 Juin

du 3 au 7 Septembre
du 12 au 16 Novembre

Public:

Développeurs Java.

Objectifs:

Compléter des connaissances de base, Acquérir un niveau d'expertise en Java.

Connaissances préalables nécessaires:

Il est demandé aux participants de connaître les notions de base du langage Java.

Programme:

Le langage	Travaux pratiques de rappels. Présentation du JDK utilisé. Nouveautés du JDK 1.5/1.6
Bibliothèques	Rappels sur la compilation. Principe des librairies, JFC, Java Beans, Swing. Gestion des paquetages. Fichiers jar. CLASSPATH. Présentation de ant.
Collections	Manipulation d'ensembles d'objets. Listes, collections, ensembles, dictionnaires.
Stream	Principe de InputStream et OutuputStream Exemples de mise en œuvre. DataInputStream et DataOutputStream.
Accès fichiers et réseau	StreamTokenizer et bibliothèque NIO La bibliothèque net : mise en œuvre avec des exemples.
JDBC	Mécanismes de connexion aux bases de données. Principes de JDBC. Utilisation des méta données. Exploitation du ResultSet. Gestion du transactionnel. Exemple d'application transactionnelle de Java à SQL
XML	Historique, notions de bases. Les schémas. Présentation des API de programmation : DOM, SAX. Utilisation des API XSLT, JAXB, DIGESTER. Travaux pratiques : écriture d'un mini parseur XML.
Réseau	Les classes principales d'accès au réseau Programmation par socket, sérialisation. Transferts d'objets au travers du réseau. Création d'une application client/serveur réseau. Accès aux ressources partagées, synchronisation, verrous. Utilisation des threads.

Programmation avancée en Java

Accès concurrents	Accès aux ressources partagées, synchronisation, verrous Utilisation des threads Multi-threading : interface Runnable et classe Thread, les états d'un thread, interruption, priorités, synchronisation; Wait(), Notify() et NotifyAll().
Java Management eXtension	Supervision avec JMX Principe des MBeans, et exemples de MBeans standards fournis à partir du jdk 1.5
Graphiques	Présentation des outils graphiques. Les composants. Techniques d'affichage et rafraichissement. Gestion des évènements. Les différents objets : cadres , menus, cases à cocher, ...
Sécurité	Introduction à la sécurité Java. Protection du système vis à vis des applications tierces: SecurityManager, ClassLoader. JCE. Chiffrement, chiffrement asymétrique, hachage
Intégration	exécution de commandes système depuis Java. appel de programme en C depuis Java: classes natives. appel de Java depuis le C. Instanciation d'une JVM.

Sécurité Java et JEE

Durée: 3 jours
1420 €

du 5 au 7 Mars
du 14 au 16 Mai

du 5 au 7 Septembre
du 5 au 7 Novembre

Public:

Tout développeur souhaitant maîtriser la sécurité des applications Java et Jee.

Objectifs:

Connaître les risques potentiels dans l'utilisation de Java, et les parades à mettre en œuvre, les moyens de sécuriser les applications Jee.

Connaissances préalables nécessaires:

Il est demandé aux participants de connaître les notions de base du langage Java.

Programme:

Besoins	Les risques Politique de Sécurité Evaluation des risques en fonction des différents modes d'utilisation de Java (applets, application, servlets)
Sécurisation de la JVM	Limites naturelles imposées par Java : gestion mémoire. Contrôle du bytecode par la machine virtuelle. Mise en œuvre du SecurityClassLoader
Protection de l'exécution	Exécution protégée : SecurityManager, ClassLoader. Surcharge des méthodes d'accès: lecture, écriture, exécution, ouverture de socket, autorisation de connexions... TP: Protection des accès sur le disque local d'une application.
Chiffrement	Les mécanismes de signature. Création de clés publiques et privées. Les clés RSA, DSA. Signature d'un document. Les algorithmes SHA 1 with DSA, MD5 with RSA. Les MessageDigest. Les algorithmes MD2, MD5, SHA-1, SHA-512 TP: Vérification de l'authenticité d'un document
Certificats	Cycle de vie d'un certificat. La fabrique de certificats Java. Les certificats de modification X509.
Contrôle	Rappel sur les ACL. Le paquetage java.security.acl. Ajout d'entrée, vérification d'accès.
Obfuscation	Principe Techniques d'obfuscation Solutions commerciales
JAAS	Présentation Fonctionnement et mise en œuvre

Sécurité Java et JEE

Sécurité Jee

Exemples avec WebSphere et JBoss

Le service de sécurité

Sécurité Web et EJB

Autorisations EJB V3

Accès applicatifs et lien avec un annuaire ldap

Mise en œuvre des certificats avec jee.

JEE : les servlets et JSP

Durée:3 jours

Public:

Les développeurs java souhaitant intégrer les technologies des servlets et des pages JSP.

Objectifs:

Comprendre les mécanismes des servlets et JSP. Savoir créer des pages Web dynamiques. Les travaux pratiques ont lieu avec le serveur Tomcat.

Connaissances préalables nécessaires:

Il est demandé aux participants de connaître la programmation Java, ainsi que les techniques de base Internet (HTML, serveur HTTP).

Programme:

Définitions	Application Java, applet, servlet, JSP. Le modèle MVC.
Besoins	pages HTML dynamiques, communication avec des bases de données, applications Java, et inter-servlets. Rappels sur le modèle JEE, les composants d'un serveur d'application Java.
servlets	Cycle de vie: Instanciation, initialisation, traitement des requêtes et arrêt des servlets, persistance, sessions contexte de session HTTP, cookies, réécriture d'URL. méthodes d'accès, objet de requête et objet de réponse chaînage de servlets, utilisation en filtres, communication entre servlets. Travaux pratiques : modification des images d'un site
JSP	Java Server Pages : positionnement par rapport aux servlets. Balises JSP. Redirections et enchainements. Cycle de vie.
Architecture	Découplage Java/HTML à l'aide de JavaBean. Exemple avec le pattern commande.
XML	Extensible Markup Langage : présentation, utilisation dans les fichiers de configuration
Gestion des données	Connexion aux bases de données : JDBC, pool de connexions. Accès aux données depuis les servlets. Gestion des transactions. Exécution d'instructions SQL, utilisation de blob. Mapping XML.
Mise en œuvre	échange de données entre un client HTML et une base de données sur un serveur Web. Exemple d'application transactionnelle de Java à SQL.
Ajax JSP Tag Library	Utilisation d'Ajax dans les pages JSP : les balises disponibles

JEE : les EJB

Durée: 3 jours
1400 €

du 25 au 27 Janvier
du 28 au 30 Mars
du 14 au 16 Mai

du 29 au 31 Août
du 29 au 31 Octobre

Public:

Développeurs Java, concepteurs, chefs de projet.

Objectifs:

Mettre en œuvre des applications Jee manipulant des EJB 3. Développer des EJB3. Déployer les applications Jee dans un serveur d'applications.

Connaissances préalables nécessaires:

Il est demandé aux participants de connaître la programmation en Java. La connaissance des architectures distribuées est un plus.

Programme:

Les concepts des architectures distribuées	L'architecture distribuée : C/S, architecture n-tiers. L'architecture serveur d'applications. La plateforme Java EE. Le RMI (Remote Method Invocation).
Programmation avec JNDI	Présentation de Java Naming and Directory Interface. Les services de désignation, d'annuaire. L'architecture JNDI.
Les spécifications de Java EE 5	Les annotations Les génériques
Les spécifications EJB2 et EJB3	Normes EJB 2.0, EJB 3.0. Le Modèle Vue Contrôleur (MVC). Contexte d'utilisation des EJB. Les différents types d'EJB : session, entity, message driven.
Les EJB session stateless, stateful	Utilité, cycle de vie, développement, déploiement. Mode conversationnel avec les beans à état. Problématiques de concurrence et de clustering. EJB session et Web Services.
Les EJB entité et la norme JPA	Norme JPA (Java Persistence API) et configuration. Gestionnaire de persistance. Relation avec les graphes d'objets. Les mécanismes d'héritage.
Le langage EJB-QL (Query Language)	Les types de requêtes. Jointure et restrictions.
Gestion des transactions	Les différents modèles transactionnels. Gestion des transactions distribuées. Transactions de niveau conteneur ou bean. Transactions au niveau du client.
Gestion de la sécurité	Sécurité Jee avec JAAS. Sécurité par programmation. Sécurité déclarative.

JEE : les EJB

Architecture MOM avec les EJB MDB

Rappel des concepts JMS et MOM.
Développement d'EJB MDB (message driven bean) et de clients.
Déploiement d'une architecture MOM.

Développement Web Services

Durée: 3 jours
1535 €

du 27 au 29 Février
du 2 au 4 Avril
du 25 au 27 Juin

du 1er au 3 Octobre
du 3 au 5 Décembre

Public:

Les chefs de projets et développeurs souhaitant concevoir et développer des web services.

Objectifs:

Savoir développer des applications utilisant les techniques des Web services.

Connaissances préalables nécessaires:

Ce cours présente les WebServices en environnement Java.

Programme:

Introduction	Historique. Définitions. Les différents types de webservices : ws-* et RESTful. Architecture technique. Rappels sur XML. Notion de schéma XML.
Le protocole SOAP	Présentation : Simple Object Access Protocol pour l'échange de messages XML. L'interopérabilité avec SOAP, les avantages. Structure d'un message. Exemples.
WSDL	Définition. Structure d'un document WSDL. Définition d'un service. Gestion de la sécurité.
Les annuaires UDDI	Universal Description, discovery and Integration pour la recherche des services web disponibles. Les annuaires publics. Structure des données. Travaux pratiques : mise en œuvre de jUDDI Publication d'un Webservice.
WebServices et Java	Le Langage WSDL : Web Service Description Langage Les API Java pour XML : JAXP, JAXB, SAAJ : pour le traitement des données XML JAX RPC, lancement de procédures distantes JAXM, messages XML; JAXR, identification de services web Le WSDP : Java Web Services Developer Pack.

Développement Web Services

WebServices et Axis	Présentation Axis. Principe de fonctionnement. Travaux pratiques : installation Axis, lancement du serveur, développement d'un web Service, et déploiement avec Axis. Interrogation depuis un client Perl, PHP, Java. Génération du code client à partir de la définition WSDL.
Mise en œuvre	Passage de paramètres. Maintenance de session avec la méthode <code>setMaintainSession()</code> Gestion des attachements. Sérialisation personnalisée avec <code>typeMapping</code> .
Axis 2	Présentation, historique Fonctionnalités. Travaux pratiques : installation, lancement du serveur, tests. Installation de l'application web Axis2. Administration du produit par l'interface Web. Déploiement d'un service Web.
Web Services et sécurité	Le besoin. Identification des menaces. Différents moyens de sécurisation : WS-Security, Username Token, X.509 Certificate Token Profile. L'authentification HTTP. Authentification du client. Création des rôles de sécurité. Pose de contraintes. Vérification. Travaux pratiques : mise en œuvre HTTPS avec tomcat. Génération de clés. Déclaration du connecteur sécurisé sous Tomcat.
Le standard JAX-RS	La technologie des webservices RESTful: Representational State Transfer. Format des données transférées : XML, JSON. Le WADL : Web Application Description Language. Les implémentations : Apache-CXF, Jersey. Mise en œuvre de webservices RESTful avec Jersey.
Web Services et mobilité	DPWS : Devices Profile WebServices. Présentation, objectifs. Exemples de mise en œuvre du standard DPWS

Techniques et outils de migration vers les EJB3

Durée: 1 jour

Prix et dates: nous consulter

Public:

Les développeurs d'applications JEE

Objectifs:

Connaître les principales différences entre les EJB3 et les EJB2, et les difficultés et solutions lors de la migration vers les EJB3.

Connaissances préalables nécessaires:

Bonnes connaissances de Java/JEE et des EJB

Programme:

Rappels sur les standards EJB 2.1	Les interfaces, méthodes d'accès, descripteurs de déploiement
Nouveautés EJB 3.0	Utilisation des annotations. Exemples de Bean Session, Entité et MD en v2.1 et leur équivalent en v3
Interopérabilité	Compatibilité ascendante des EJB3. Utilisation d'EJB en version 2 avec des EJB 3. Utilisation des fichiers de déploiement xml avec les EJB 3 Mise en œuvre d'une application faisant cohabiter les deux versions d'EJB.
Compatibilité et migration	Organisation des migrations, les limites de la cohabitation des EJB2 et EJB3 Les jointures entre EJB 2.1 et EJB3, le service ejb3-deployer dans JBoss, ... Les solutions techniques
Automatisation	Migration en masse des composants EJB : les outils du marché

Serveurs d'application JEE

Durée: 2 jours
930 €

du 15 au 16 Mars
du 10 au 11 Mai

du 3 au 4 Septembre
du 8 au 9 Novembre

Public:

Les chefs de projets et toute personne souhaitant comprendre le fonctionnement de l'architecture JEE.

Objectifs:

Connaître l'architecture et les principes des serveurs d'applications. Savoir concevoir une application avec le modèle JEE.

Connaissances préalables nécessaires:

Aucune connaissance préalable n'est requise pour suivre ce cours.

Programme:

Serveurs d'application	Introduction. Transactions. Architecture des applications web. Les différents éléments et leurs rôles
Le modèle JEE	L'architecture JEE. Le modèle JEE. servlets, Java Server Pages, EJB. Spécifications. Les composants d'un serveur d'application Java
Les produits	Présentation de différents serveurs d'application du marché : Geronimo, JBoss, Jonas, WebLogic, WebSphere Comparatif produits: version de JDK, type d'administration, automatisation, industrialisation, besoin en ressources, support des EJB3, ... Portabilité des applications JEE.
Administration	Définition des différents objets à gérer : serveurs Web, serveur d'application, moteur de servlets, container, EJB, hôtes virtuels, connecteurs JDBC. Le service de nommage JNDI.
Conteneur Web	Servlets, pages JSP: pages HTML dynamiques, communication avec bases de données et applications Java. Les frameworks de développement : objectifs et techniques mise en œuvre Struts, JSF, Apache MyFaces, Spring
Développement avec les EJB	Les EJB : spécifications (état actuel et limites). EJB entité, EJB session, EJB Message-driven. Apports des EJB3. Jointures. Transactions. Présentation de EJB-QL. Relations entre le développeur d'EJB et le DBA.
Intégration	Transactions. Utilisation de JTA, JTS. Transactions explicites. Transactions gérées par container. Sécurité: JAAS, rôles, groupes de permissions. Mécanismes de connexion aux bases de données : JDBC, pool de connexions. Connexion interapplicatives avec JCA.

Serveurs d'application JEE

Déploiement et exploitation

Cycle de vie d'une application. Industrialisation. Les fonctions à assurer: code, assemblage, nommage, création des fichiers de description en XML, des fichiers jar, mise en production.
Les outils de développement: eclipse, WSAD.
Les outils de mise en production.

Développement JEE avec WebSphere

Durée: 5 jours
2315 €

du 12 au 16 Mars
du 4 au 8 Juin

du 3 au 7 Septembre
du 10 au 14 Décembre

Public:

Développeurs et intégrateurs d'applications JEE avec le serveur d'applications WebSphere.

Objectifs:

Connaître l'architecture et les principes du serveur d'applications WebSphere 6.1 et 7.0. Savoir concevoir, développer, et déployer une application..

Connaissances préalables nécessaires:

Il est demandé aux participants de connaître la programmation Java, ainsi que les techniques Internet (serveur HTTP, HTML,...) et quelques notions de bases de données.

Programme:

Serveurs d'application Jee	Le modèle JEE. Architecture des applications web. les composants d'un serveur d'application Java : serveur HTTP, moteur de servlets, serveur d'EJB.
WebSphere	Présentation des fonctionnalités du produit WebSphere La terminologie et les objets WebSphere Application Server : application d'entreprise, serveur d'application, application web, sources de données, pilote JDBC, hôtes virtuels. L'architecture des applications Jee avec WebSphere.
Servlets, pages JSP	pages HTML dynamiques, communication avec bases de données et applications Java, communication avec d'autres servlets. Liaison JSP/JavaBean.
Développement	Composants d'une application d'entreprise: .ear, .jar, .rar, .war Déploiement dans WebSphere. Utilisation des outils d'assemblage d'applications: wrd, rad, ... Regroupement de composants unitaires en une application. Création d'une application sur-mesure (créer un .ear, un .war, etc.).
Développement d'un EJB	Types d'EJB : EJB entité, EJB session, EJB Message-driven. Persistance et transactions. Méthodes de base: ejbCreate(), ejbActivate(), setSessionContext(), ejbStore(),... Fichiers de description en XML. Les fichiers spécifiques de WebSphere: ibm-application-bnd/ext/ext-pme.xmi, ibm-ejb-jar- *.xmi, ibm-web-*.xmi.
EJB Session	EJB stateless, stateful. Notion de conversation. Fichiers de déploiement. Contraintes de conversation. Transactions. Travaux pratiques : création d'un EJB stateless "Conversion de devises", création d'un EJB stateful "Caddie"

Développement JEE avec WebSphere

EJB Entity	<p>Persistance avec les EJB BMP, CMP, CMP2. Fichiers de déploiement. Définition des clés primaires. Définition des méthodes findBy. Gestion des jointures. Travaux pratiques : création d'un EJB BMP persistant, d'un CMP de type "Compte Bancaire" et de CMP avec jointure. Relations entre EJB (one-one,one-many,many-many). Le langage EJB-QL (EJB Query Language) Correspondance objet/relationnel. Mapping XML.</p>
Intégration	<p>Transactions. Utilisation de JTA, JTS. Transactions explicites. Transactions gérées par container. Mise en évidence du retour arrière déclenché par le conteneur WebSphere. Sécurité. Mise en place d'une politique de sécurité lors du déploiement de l'application. Sécurité: JAAS, rôles, groupes de permissions.</p>
Introduction aux EJB3	<p>Apport du jdk1.5. Annotations et génériques. Principes de mise en œuvre des EJB3. Simplification de la définition des interfaces. Utilisation des annotations en remplacement du descripteur de déploiement. Simplification de la correspondance objet/relationnel. Compatibilité ascendante. Les EJB3 sous WebSphere 6.1.</p>
EJB Session en version 3	<p>Reprise de l'EJB Caddie. Migration en version 3. Dépendance d'injection. Intercepteurs de méthodes métier.</p>
EJB Entité en version 3	<p>Les fichiers de persistance. Méthodes d'accès: FIELD/PROPERTY. Mapping par défaut. Relations en EJB3. Héritage. Synchronisation avec la base de données, détachement/attachement.</p>

Développement JEE avec JOnAS

Durée: 5 jours

Prix et dates: nous consulter

Public:

Développeurs, chefs de projet et intégrateurs d'applications en modèle JEE.

Objectifs:

Connaître l'architecture et les principes du serveur d'EJB JOnAS. Savoir concevoir une application avec le modèle JEE. Savoir programmer et utiliser les EJB.

Connaissances préalables nécessaires:

Il est demandé aux participants de maîtriser la programmation Java.

Programme:

Le modèle JEE	Les composants d'un serveur d'application Java : serveur HTTP, moteur de servlets, serveur d'EJB. Terminologie: application web, sources de données, pilote JDBC.
JOnAS	Présentation des fonctionnalités du produit JOnAS Support des spécifications JEE : EJB, JTA, JDBC, JMS, JMX, JNDI Installation, configuration standard. Composition d'une version personnalisée avec l'outil Ant, vérification avec JOnAS Check. Les fichiers de configuration : jonas.properties, carol.properties, jonas-web.xml, jonas-ejb-jar.xml. Architecture d'une application JEE avec JOnAS : intégration avec Apache, Tomcat, Jetty.
Gestion des ressources	Fournisseurs d'URL, fournisseurs JMS (JORAM), sessionMail. Mécanismes de connexion aux bases de données : JDBC, pool de connexions. Présentation de JDO. Mapping XML. Connexion interapplicatives avec JCA.
Servlets, pages JSP	pages HTML dynamiques, communication avec bases de données et applications Java, communication avec d'autres servlets
Développement avec les EJB	Les EJB : spécifications (état actuel et limites). EJB entité, EJB session, EJB Message-driven fichiers de description en XML Le générateur de classes d'interposition: GenIC Jointures en CMP2. Relations entre EJB, présentation de EJB-QL. JOnAS JTM: Gestion des transactions. Sécurité: rôles, groupes de permissions. Intégration dans Eclipse avec le plugin Jonas.
Application répartie	Le service de nommage L'accès depuis Java : l'interface JNDI

Développement JEE avec JOnAS

Déploiement	Les différents composants d'une application : les fichiers application d'entreprise (.ear), les fichiers EJB (.jar), les fichiers application web (.war). Déploiement dans JOnAS.
Intégration	Regroupement de composants unitaires en une application. Création d'une application sur-mesure (créer un .ear, un .war, etc.).
Exploitation	La console d'administration JMX : jonas admin Gestion des performances (surveillance mémoire, threads) Analyse des fichiers de logs dans JOnAS et dans Tomcat pour les requêtes http.

Applications JEE avec JBoss

Durée: 5 jours
2315 €

du 12 au 16 Mars
du 18 au 22 Juin

du 10 au 14 Septembre
du 12 au 16 Novembre

Public:

Développeurs et intégrateurs d'applications en modèle JEE.

Objectifs:

Connaître l'architecture et les principes du serveur JBoss. Savoir concevoir une application avec le modèle JEE, et la mettre en œuvre avec Jboss et Tomcat.

Connaissances préalables nécessaires:

Il est demandé aux participants de connaître la programmation Java, ainsi que les techniques Internet (serveur HTTP, HTML,...) et quelques notions de bases de données.

Programme:

Serveurs d'applications	Architecture des applications web. Jee et le modèle MVC. Composants d'un serveur d'application JEE: serveur HTTP, moteur de servlets, conteneur d'EJB. Positionnement des outils : Jboss, Tomcat, Apache, ... Terminologie: application web, conteneur, sources de données, pilote JDBC, hôtes virtuels.
JBoss	Présentation. Le projet JBoss. L'architecture Jboss : le noyau, la couche services, la couche présentation, la couche application, le principe JMX et les Mbeans. L'implémentation JMX dans JBoss. Installation, configuration. Utilisation de la console JMX.
Développement	Composants d'une application d'entreprise: .ear, .jar, .rar, .war Outils d'assemblage d'applications: ant, plugins, ... Regroupement de composants unitaires en une application. Création d'une application sur-mesure (créer un .ear, .war, ...). Déclaration d'une source de données.
Servlets, pages JSP	Cycle de vie des servlets, servlets thread-safe Définition des JSP. Liaison JSP/JavaBean.
JNDI	Introduction, présentation de l'API. JNDI pour JEE.

Applications JEE avec JBoss

Développement d'un EJB	<p>Choix du modèle. Types d'EJB : EJB entité, EJB session, EJB Message-driven. Persistance et transactions. Annotations spécifiques EJB 3. Annotations et génériques. Principes de mise en œuvre des EJB3. Fichiers de description en XML. Les fichiers spécifiques: jboss.xml, jboss-web.xml, ... Annotations en remplacement du descripteur de déploiement. Simplification de la correspondance objet/relationnel. Compatibilité ascendante. Les EJB3 sous JBoss.</p>
EJB Session	<p>EJB stateless, stateful. Notion de conversation. Fichiers de déploiement. Contraintes de conversation. Transactions. Dépendance d'injection. Intercepteurs de méthodes métier. Travaux pratiques : création d'un EJB stateless "Conversion de devises", création d'un EJB stateful "Caddie"</p>
EJB Entité	<p>Persistance avec les EJB. Fichiers de déploiement. Définition des clés primaires. Définition des méthodes de recherche (query). Gestion des jointures en EJB3. Les fichiers de persistance. Méthodes d'accès: FIELD/PROPERTY. Mapping par défaut. Utilisation de l'EntityManager. Relations entre EJB (one-one,one-many,many-many). Le langage EJB-QL Héritage. Synchronisation avec la base de données, détachement/attachement. Correspondance objet/relationnel sous JBoss: jbosscmp-jdbc.xml. Travaux pratiques : création d'une application utilisant des jointures Many/Many</p>
EJB MessageDriven	<p>Développement de composants orientés message Principe des MOM (Middlewares Orientés Message). Présentation de JMS. Développement d'un EJB Message Driven (MDB). L'interface JMS MessageListener. Traitement asynchrone. La méthode onMessage(). Fichiers de déploiement. Cycle de vie des MDB. Gestion des erreurs.</p>
Transactions	<p>Les transactions distribuées: commit à deux phases. Utilisation de JTA, JTS. Transactions explicites ou gérées par container. Annotations de transaction. Gestion des exceptions dans un contexte transactionnel. Mise en œuvre d'une application EJB3 transactionnelle. Mise en évidence du retour arrière déclenché par le conteneur JBoss et de la propagation du contexte transactionnel.</p>
Sécurité	<p>JAAS: rôles, groupes de permissions. Mise en œuvre dans le conteneur d'EJB. Rôles et permissions sur les méthodes. Intégration de la sécurité dans le serveur d'applications. Mise en place d'une politique de sécurité lors du déploiement de l'application. Mise en œuvre d'une application en cluster.</p>

Conception JEE avec les Design Patterns

Durée:4 jours

Prix et dates: nous consulter

Public:

Chefs de projets, architectes et ingénieurs concepteurs.

Objectifs:

Comprendre les patrons de conception (Design Patterns) et les apports de UML2. Savoir les mettre en œuvre dans une architecture technique de conception.

Connaissances préalables nécessaires:

Il est demandé aux participants de connaître les notions de programmation objets,et des architectures JEE et EJB

Programme:

Définitions	Principes des solutions de conception cataloguées. Méthodologie: définition des besoins techniques, des classes "types" du pattern, des collaborations entre classes.
Présentation	Rappels d'architecture. Le modèle MVC,les outils de mise en œuvre (framework, ...). Les design patterns. Description formelle. Choix et utilisation d'un patron. Présentation des patrons de conception: origine, les 3 familles (création, structuration et comportement), autres patrons
Patrons de création	Rendre un système indépendant de la façon dont les objets sont créés. Etude des patrons: fabrique abstraite, constructeur, prototype, singleton
Patrons de structuration	Composer des objets pour obtenir de nouvelles fonctionnalités Identifier des structures de classes abstraites et évolutives Etude de: adaptateur, pont, composite, décorateur, façade, proxy
Patrons de gestion des comportements	Identifier des coopérations évolutives. Etude de: chaîne de responsabilités, commande, interpréteur, itérateur, médiateur, memento, observateur, état, stratégie, patron de méthode, visiteur.
Communication	Coopérations découplées utilisables dans un système distribué Etude de: publication/souscription, proxy
Mise en œuvre	Modélisation des patrons de conception et intégration à un diagramme de classes UML. Création d'une application JEE en utilisant les patterns de Sun : SessionFacade, BusinessDelegate, serviceLocator, ..

Administration WebSphere

Durée: 5 jours
2315 €

du 30 Janvier au 3 Février
du 16 au 20 Avril
du 25 au 29 Juin

du 24 au 28 Septembre
du 19 au 23 Novembre

Public:

Techniciens d'exploitation, administrateurs WebSphere.

Objectifs:

Connaître l'architecture et les principes du serveur d'applications WebSphere. Savoir installer, configurer et exploiter des applications sous WebSphere. Les travaux pratiques sont réalisés avec la version 6 et 7.

Connaissances préalables nécessaires:

Une bonne connaissance des concepts de l'internet, des systèmes d'exploitation, et quelques notions de base sur les bases de données.

Programme:

WebSphere	Présentation des fonctionnalités du produit WebSphere. Rappels sur la terminologie et les objets WebSphere Application Server. Architecture. Topologie WebSphere.
Installation	Le produit WebSphere Application Server, Base de données, IBM-HTTP server, jdk. Installation manuelle. Installation automatique.
Administration du système	Fonctionnement de la console Les objets à administrer : arborescence, groupes de serveurs, nœuds, applications d'entreprises. Paramètres de configuration : la base de données de configuration, les fichiers XML.
Intégration	Utilisation de l'outil d'assemblage d'applications (WRD). Le service de nommage: l'interface JNDI. Gestion des hôtes virtuels Variables WebSphere. Domaines de réplication Création de profils serveurs.
Ressources	Sources de données. Exemple: mise en œuvre avec PostgreSQL. Modification de la base de données cible. Fournisseurs d'URL, fournisseurs JMS, les sessions JavaMail. Connexion interapplicatives avec JCA. Gestion du pool de connexions sous WebSphere Application Server.
Sécurité	Les niveaux de sécurité dans WebSphere (Système, ressources, les rôles, etc ...). Définition des rôles. Authentification Idap. Configurer la sécurité du serveur

Administration WebSphere

Outils d'administration	Console d'administration. scripts prédéfinis Outils de migrations, collector. Administration à distance, automatisation. wsadmin: présentation, objets supportés: \$AdminApp, \$AdminConfig, \$AdminControl, \$AdminTask JACL: Syntaxe de base, contrôle de programme, exemples. Jython: Syntaxe de base, contrôle de programme, exemples.
Surveillance	Informations émises par WebSphere, les exceptions, les messages de la console d'administration, les fichiers de traces. Journalisation. Journal de la JVM. Vérification de la configuration. Résolution des incidents
Gestion des performances	Etude du Ressource Analyser. Exploitation de l'advisor.
Multi serveurs	Déclaration de plusieurs serveurs sur la même machine dans le même profil. Routeur HTTP IBM: Configuration Mise en place d'un cluster avec affinités de Session sans gestionnaire de déploiement

Administration Tomcat

Durée: 2 jours
915 €

du 26 au 27 Janvier
du 15 au 16 Mars
du 7 au 8 Juin

du 27 au 28 Septembre
du 29 au 30 Novembre

Public:

Toute personne souhaitant configurer, et administrer une application avec Tomcat.

Objectifs:

Comprendre le fonctionnement de Tomcat, et savoir le mettre en œuvre, l'installer, le configurer et l'administrer, optimiser le fonctionnement du serveur.

Connaissances préalables nécessaires:

Il est demandé aux participants de connaître les bases tcp/ip, http, java (jsp, servlets)

Programme:

Concepts de base	L'architecture des applications web : les différents composants. Présentation de Tomcat. Les versions de Tomcat. Utilisation de Tomcat avec le serveur Web Apache Servlets, JSP et composants, architecture d'un site Java
Installation	Installation de Tomcat et modification des paramètres de base. Exercice : installation du serveur Tomcat, positionnement des variables d'environnement, lancement du serveur, tests de fonctionnement.
Configuration	Architecture de Tomcat : "engine", services, "context". L'arborescence, les variables d'environnement : JAVA_HOME, TOMCAT_HOME, CLASSPATH.
Console Manager	Présentation, fonctions disponibles : état du serveur, accès à la documentation, à la console d'administration.
Déploiement d'applications web	Descripteurs XML, les fichiers .war Déploiement à chaud. Tomcat Manager. Travaux pratiques : déploiement d'une application simple par l'interface d'administration de Tomcat.
Console d'administration	Travaux pratiques : installation et configuration des droits d'accès dans le fichier tomcat-users.xml Fonctions disponibles : liste des ports écoutés par tomcat pour les requêtes http, applications déployées sur chaque hôte virtuel, configuration des connecteurs, de la sécurité, (utilisateurs, groupes, rôles), des sources de données.

Administration Tomcat

Sécurité	<p>Sécurisation et permissions. Les domaines, les rôles : définitions, principe de fonctionnement et configuration. Gestion des utilisateurs, modification de rôles, gestion des mots de passe. Le stockage des informations de sécurité : JDBC, Datasource, JNDI, JAAS, mémoire Security Manager. Travaux pratiques : configuration de domaines de sécurité, création et affectation d'utilisateurs, spécification des pages protégées, mise en place des mots de passe</p>
Fichiers de configuration	<p>Etude des paramètres à positionner dans le fichier server.xml, les balises server, engine, host, context, logger, loader. Le descripteur de déploiement web.xml : déclaration de la servlet, lien entre la servlet et la requête. Configuration des sources de données : présence des pilotes, optimisation des paramètres de connexion. Travaux pratiques : mise en place d'un pilote pour l'accès à une base postgresSQL. Les connecteurs : la balise connector Le connecteur HTTP, fonctionnement Tomcat en standalone. Sécurisation : configuration du protocole https.</p>
Performances et tests	<p>gestion de la charge : load-balancing Mise en place de clusters. Interface apache/tomcat avec mod-jk. Travaux pratiques : installation et configuration d'un répartiteur mod_jk Conservation des sessions : mise en place des sticky sessions.</p>
Journalisation	<p>Mise en œuvre de la journalisation avec log4j. Analyse des logs Fichiers de logs, résolution d'incidents. Travaux pratiques : Configuration de Log4j. Modification des niveaux de journalisation. Différents types ventilations. Centralisation des logs vers une machine syslog externe.</p>
Exploitation	<p>Intégration JMX. Suivi des performances. Suivi du ramasse miettes (garbage collector) avec jconsole. Paramètres de la JVM pour gérer au mieux l'espace mémoire. Travaux pratiques: mise sous stress avec jmeter, suivi et amélioration de la configuration</p>

Administration JBoss

Durée: 4 jours
1910 € HT

du 23 au 26 Janvier
du 19 au 22 Mars
du 29 Mai au 1er Juin

du 24 au 27 Septembre
du 26 au 29 Novembre

Public:

Exploitants, administrateurs d'applications JEE fonctionnant avec JBoss.

Objectifs:

Savoir installer et configurer JBoss. Savoir intégrer une application JEE, en assurer la mise en production, l'exploitation.

Connaissances préalables nécessaires:

Une bonne connaissance des concepts de l'internet, des systèmes d'exploitation, et quelques notions de base sur les bases de données.

Programme:

JBoss	<p>Présentation</p> <p>L'architecture Jboss : le noyau, la couche services, la couche présentation, la couche application, le principe JMX et les Mbeans. Les outils JBoss: Hibernate, AOP, cache IDE.</p> <p>Présentation du noyau JBoss.</p> <p>Présentation du micro-conteneur.</p> <p>Installation, configuration, arborescence des fichiers.</p> <p>Les services disponibles.</p> <p>Adaptation de la configuration : ajout/suppression de services</p> <p>Visualisation dans la console JMX de JBoss.</p>
Applicatif	<p>Terminologie: application web, container, sources de données, pilote JDBC, hôtes virtuels.</p> <p>Configuration des services essentiels : JBossWeb, JNDI, JBossMQ, ...</p> <p>Configuration du scanner de déploiement d'applications.</p>
Lien avec Tomcat	<p>Installation de Tomcat et modification des paramètres de base</p> <p>Activation du service Tomcat dans la configuration Jboss.</p> <p>Le fichier server.xml. Administration du service web.</p> <p>Mise en place d'hôtes virtuels.</p>
Gestion des ressources	<p>Fournisseurs d'URL, fournisseurs JMS, sessionMail.</p> <p>Mécanismes de connexion aux bases de données : JDBC, pool de connexions. Modification de la base de données cible.</p> <p>Exemple avec postgresql.</p> <p>Connexion interapplicatives avec JCA.</p>
Sécurité avec JAAS et SecurityManager	<p>Intégration de la sécurité dans le serveur d'applications.</p> <p>Mise en place d'une politique de sécurité lors du déploiement de l'application. Sécurité: JAAS, rôles, groupes de permissions.</p>

Administration JBoss

Automatisation	Exploitation par scripts avec twiddle. Recherche de MBeans. Ecriture d'un script de visualisation de statistiques sur un cluster: taux d'utilisation mémoire, nombres de requêtes, connexions JDBC. Scripts locaux avec bshDeployer.
Intégration	Configuration de JBoss en multi-instances sur un même serveur. Accès depuis un serveur apache, mise en place d'une liaison ajp à l'aide de mod_jk. Ecriture d'un service .sar JBoss permettant de simuler une application batch.
Journalisation	Mise en œuvre de la journalisation avec log4j. Analyse des logs Fichiers de logs, résolution d'incidents. Configuration de Log4j. Modification des niveaux de journalisation. Différents types ventilations. Centralisation des logs vers une machine syslog externe.
Suivi	Gestion des performances. Suivi du garbage collector. Installation du MBean Jboss Profiler. Suivi du ramasse miettes (garbage collector) avec jconsole. Paramètres de la JVM pour gérer au mieux l'espace mémoire. Travaux pratiques: mise sous stress avec jmeter, suivi de la charge et amélioration de la configuration. Optimisation et allègement des configurations JBoss. Utilisation du MBean SNMP. Envoi de trap SNMP à une console de supervision. Lien vers la supervision avec Nagios via SNMP. Automatisation d'alertes en cas d'incident.
JBoss Operations Network	Fonctionnalités : déploiement, gestion des ressources disponibles, arrêt/démarrage des applications, supervision : émission d'alertes, monitoring, ... Principe de fonctionnement, architecture, prérequis techniques. Installation de JBoss ON, configuration et mise en œuvre.

Administration WebSphere avancée

Durée: 4 jours

Prix et dates: nous consulter

Public:

Techniciens d'exploitation, administrateurs WebSphere.

Objectifs:

Connaître l'architecture et les principes du serveur d'applications WebSphere. Savoir administrer des applications sous WebSphere, et expertiser une application sous WebSphere 6.1 et 7.0 Network Deployment.

Connaissances préalables nécessaires:

Une bonne connaissance des concepts de l'internet, des systèmes d'exploitation, et de WebSphere Application Server.

Programme:

Network Deployment	Gestion des nœuds Installation du gestion de nœuds. Installation d'un nœud piloté. Transformation d'un serveur autonome en nœud piloté. Fonctionnement des agents locaux. Gestion des serveurs.
Cluster	Gestion de cluster: ajout de nœud, visualisation. Gestion des sessions répliquées. Mise en évidence des reprises de session. Déploiement d'applications sur la cellule. Déploiement séquentiel.
Automatisation	En ligne de commande, en mode script. scripts avancés de gestion des nœuds Utilisation de Jython. Positionnement par rapport à JAAS. Les apports de Python. Les fonctions, modules, traitement des paramètres. wsadmin: Présentation, objets supportés: AdminApp, AdminConfig, AdminControl, AdminTask Appel des API Java, Jython et XML
Redirecteur HTTP	Routeur HTTP IBM: Configuration Mise en place des affinités de Session. Chiffrement SSL en amont et en aval du plugin. Liaison avec un serveur Apache interne en SSL.
Sécurisation des accès	Mise en place d'un serveur LDAP en tant que référentiel utilisateurs. Chiffrement. Mise en place d'un fournisseur externe d'authentification. Utilisation d'un JACC.
Gestion des performances	Suivi des performances à l'aide de scripts: récupération des compteurs WebSphere depuis un script Jython. Relevés automatiques avec déclenchement d'alertes. Mesure des consommations mémoires, des threads actifs. Mise en œuvre d'un système de surveillance d'un pool de connexion JDBC.

Administration WebSphere avancée

Extensions

Présentation de la version WebSphere 7.0 Network Deployment.
Présentation de la version Extended Deployment: gestion répartie de l'état des services, gestion des charges, planification des travaux.

Exploitation JOnAS

Durée:4 jours

Prix et dates: nous consulter

Public:

Exploitants, administrateurs d'applications JEE fonctionnant avec JOnAS.

Objectifs:

Savoir installer et configurer JOnAS. Savoir intégrer une application JEE, en assurer la mise en production, l'exploitation.

Connaissances préalables nécessaires:

Une bonne connaissance des concepts de l'internet, des systèmes d'exploitation, et quelques notions de base sur les bases de données.

Programme:

JOnAS	<p>Présentation L'architecture JOnAS : le noyau, la couche services, la couche présentation, la couche application, le principe JMX et les Mbeans. Présentation du noyau JOnAS. Présentation de l'architecture OSGi. Installation, configuration, arborescence des fichiers. Les services disponibles. Adaptation de la configuration : ajout/suppression de services Visualisation dans la console JMX de JOnAS.</p>
Applicatif	<p>Terminologie: application web, container, sources de données, pilote JDBC, hôtes virtuels. Configuration des services essentiels : tomcat6, JNDI, quartz, ... Déploiement d'applications: mode manuel, mode automatique. Assemblage d'une application. Présentation de GenIC.</p>
Lien avec Tomcat	<p>Installation de Tomcat et modification des paramètres de base Activation du service Tomcat dans la configuration JOnAS. Le fichier tomcat6-server.xml. Administration du service web. Mise en place d'hôtes virtuels.</p>
Gestion des ressources	<p>Fournisseurs d'URL, fournisseurs JMS, sessionMail. Présentation de Joram et speedo. Mécanismes de connexion aux bases de données : JDBC, pool de connexions. Modification de la base de données cible. Travaux pratiques : exemple avec postgresql. Connexion interapplicatives avec JCA.</p>
Sécurité avec JAAS et SecurityManager	<p>Intégration de la sécurité dans le serveur d'applications. Travaux pratiques : Mise en place d'une politique de sécurité lors du déploiement de l'application. Sécurité: JAAS, rôles, groupes de permissions.</p>

Exploitation JOnAS

Automatisation

Recherche de MBeans. Accès aux MBeans depuis une page JSP.
Travaux pratiques : création d'une page de contrôle d'état du serveur.
Exploitation par scripts avec mbean.jar.
Récupération d'informations sur un MBean. Exécution de commandes.
Ecriture d'un script de visualisation de statistiques sur un cluster: taux d'utilisation mémoire, nombres de requêtes, connexions JDBC.
Création de graphes depuis mbean.jar.

Intégration

Configuration de JOnAS en multi-instances sur un même serveur.
Accès depuis un serveur apache, mise en place d'une liaison ajp à l'aide de mod_jk.
Travaux pratiques :
Ecriture d'un service .sar JOnAS permettant de simuler une application batch.

Suivi

Gestion des performances.
Fichiers de logs, résolution d'incidents.
Principe de monolog.
Travaux pratiques : configuration de monolog.
Modification des niveaux de journalisation.
Automatisation d'alertes en cas d'incident.
Corrélation des logs :
ventilation des journaux monolog vers une machine centrale syslog,
utilisation de Log4J en remplacement de monolog,
ventilation des journaux monolog vers une machine log4j.
Présentation et installation de JASMINe.
Utilisation de JASMINe en outil de suivi.

JBoss Portal

Durée: 3 jours

Prix et dates: nous consulter

Public:

Développeurs souhaitant mettre en œuvre JBoss Portal

Objectifs:

Comprendre le fonctionnement de JBoss Portal, savoir l'installer et le configurer et développer des portlets. Ce cours est basé sur de nombreux travaux pratiques.

Connaissances préalables nécessaires:

Bonne connaissance des concepts JEE, de la programmation Java.

Programme:

JBoss	<p>Présentation</p> <p>L'architecture Jboss : le noyau, la couche services, la couche présentation, la couche application, le principe JMX et les Mbeans.</p> <p>Les outils JBoss: Hibernate, AOP, cache IDE.</p> <p>Présentation du noyau JBoss.</p> <p>Présentation du micro-conteneur.</p> <p>Installation, configuration, arborescence des fichiers.</p> <p>Les services disponibles.</p> <p>Adaptation de la configuration : ajout/suppression de services</p> <p>Visualisation dans la console JMX de JBoss.</p>
Introduction à JBoss Portal	<p>Rappels sur les portlets :</p> <p>normes JSR-168 Portlets specifications (1.1) et JSR-286 : Portlets specifications (1.2)</p> <p>Fonctionnalités du portail :</p> <p>personnalisation, single sign on, et agregation de contenus</p> <p>Terminologie : portlets container, page, instance</p> <p>Positionnement de JBoss Portal et JBoss AS dans l'architecture JEMS.</p>
Installation et configuration	<p>Les prérequis techniques.</p> <p>Installation et configuration de base.</p> <p>Démarrage de JBoss Portal.</p> <p>Configuration du conteneur de portlets</p>
Développement	<p>l'API portlet, spécifications;</p> <p>Etats et modes;</p> <p>Descripteurs de déploiement;</p> <p>librairie de tags des portlets,</p> <p>gestion du single sign on :</p> <p>module d'authentification, intégration avec JAAS;</p>

JBoss Portal

Administration du portail	Administration du portail : connexion administrateur, droits par défaut, gestion des contenus, CMS "Administration portlet" : insertion de nouvelles pages, édition pages html, ... gestion des utilisateurs et des rôles, et groupes : ajout, suppression, affectation dans un groupe; WSRP, Webservices Remote Portlet : intégration avec les webservices, publication de remote portlets; Gestion des portlets : "Management Portlet", ajout/configuration de portlets, configuration de la personnalisation et localisation des contenus; Les portlets disponibles : JBoss portlet swap, téléchargement et déploiement de portlets; instanciation multiple;
Intégration avec un framework MVC	interaction avec les servlets, jsp dans un environnement avec jsf, spring, struts
Evolutions	Les nouvelles specifications : la JSR 286

Clusters JBoss

Durée: 3 jours
1470 €

du 20 au 22 Février
du 9 au 11 Mai

du 17 au 19 Septembre
du 17 au 19 Décembre

Public:

Architectes, administrateurs d'applications JEE souhaitant configurer, optimiser des clusters JBoss.

Objectifs:

Comprendre le fonctionnement d'une grappe de serveurs JBoss, savoir les configurer et les administrer. Comprendre les impacts sur le développement d'applications JEE en cluster.

Connaissances préalables nécessaires:

Une bonne connaissance de Jboss est nécessaire et des environnements applicatifs JEE.

Programme:

JBoss	Rappels sur l'architecture Jboss : le noyau, la couche services, la couche présentation, la couche application, le principe JMX et les Mbeans. Les outils JBoss: AOP, cache IDE, JGroups. Installation, configuration.
Haute disponibilité	Définitions : disponibilité, mirroring, redondance, cluster Architectures avec routeur HTTP ou routeur IP.
Cluster	Définitions: nœud, partition, serveur, application. Architecture HA JBoss : HAPartition, les jChannel et Jboss Cache. Les MBeans HA. Fonctionnement en cluster ou en grappes de serveurs. Travaux pratiques : Configuration d'un cluster.
Déploiement d'applications	Principe des applications réparties. Le service farm. Travaux pratiques : création d'une application simple, déploiement sur JBoss en utilisant le service farm. Tests : accès à l'application depuis chaque nœud de la partition.
Configuration clusters JBoss	Les fichiers de configuration : jboss-service.xml, log4j.xml, jbossmq-state.xml, ... configuration du réseau.

Clusters JBoss

<p>JGroups</p>	<p>Présentation du projet JGroups intégré dans JBoss. Architecture de JGroups. Configuration détaillée de JGroups : paramètres de protocoles et paramètres de découverte (PING, MPING, TCPGossip,TCPPING, UDP, TCP) Principe des Tunnels entre nœuds JBoss. Travaux pratiques : Création d'une grappe autour d'une partition. Séparation de la grappe en multi-partitions. Modifications des paramètres réseaux. Séparation de la grappe en plusieurs domaines multicast . Configuration du mode de lancement en ligne de commande et à l'aide de fichiers de configuration. Gestion de versions différentes de JBoss sur une même grappe. Problème des cast.</p>
<p>Applications</p>	<p>Déploiement d'application dans une grappe. Modes de fonctionnement réparti ou singleton. Mise à jour des applications. Réplication de sessions HTTP. Réplication d'EJB. HA-JNDI. Travaux pratiques : Mise en évidence de la disponibilité du service de nommage.</p>
<p>Routage HTTP</p>	<p>Distribution de sessions HTTP. Routage de niveau 7. Les outils. Mise en place de pound, mod_jk et haproxy. Travaux pratiques : mise en œuvre de solutions pour la répartition de charge.</p>
<p>Architecture</p>	<p>Principe d'un référentiel central d'applications. Architecture d'exploitation pour les applications. Intégration des fichiers journaux. Regroupement. Travaux pratiques : configuration de Log4j. Liaison d'une grappe avec une base de données répliquée.</p>
<p>Automatisation</p>	<p>Le langage de scripts. Les objets accessibles. Les commandes Twiddle. Requêtes d'interrogation sur un MBean. Journalisation de Twiddle. Travaux pratiques : Ecriture d'un script de visualisation de statistiques sur un cluster: taux d'utilisation mémoire, nombres de requêtes, état des connexions JDBC. Ecriture d'un script donnant une vue globale des applications d'une grappe: état, nom, charge.</p>

Développement Web avec Struts

Durée: 3 jours
1410 €

du 20 au 22 Février
du 10 au 12 Avril

du 16 au 18 Juillet
du 22 au 24 Octobre

Public:

Développeurs Java.

Objectifs:

Comprendre le fonctionnement de Struts, savoir développer une application en utilisant struts, connaître les possibilités d'interfaçage avec d'autres technologies Java.

Connaissances préalables nécessaires:

Une bonne connaissance de Java est nécessaire, et plus précisément des technologies jsp, et servlets.

Programme:

Le standard JEE	L'architecture MVC : Model View Controller. Technologies Java, XML Rappel sur la méthode de construction d'une application avec les pages jsp et les servlets.
Présentation Struts	Historique. Le projet Jakarta. Objectif de struts. Le principe, les outils disponibles.
Architecture	Le modèle MVC: les objets métiers, les objets IHM. Le rôle des différents composants: ActionServlet, pages JSP, objets ActionForm, Action, et ActionServlet. Les javabeans de stockage. La persistance : différentes solutions. La vue: HTML dynamique avec Struts. les formulaires dynamiques; affichage de beans avec struts, tests (empty, equal, greaterEqual). Le contrôleur: principe, la méthode execute. Exemple de servlet Action. Les actions standards : ForwardAction, DispatchAction
Mise en œuvre	Installation de struts. Assemblage d'une application, les fichiers web.xml et struts-config.xml, les fichiers .TLD Travaux pratiques d'intégration avec Tomcat, en environnement Eclipse.
Extensions	Struts Validator: intérêt, principe, les fichiers XML, la vue, l'assemblage. Mise en œuvre sur un exemple. DynaActionForms: implémentation, validation. Struts Templates: objectif, principe, la balise include, les pages templates, les balises tiles (tiles:insert, tiles:getAsString) Développement d'un exemple: création d'une page générique avec les templates.

Développement Web avec Struts

Exploitation

les classes de gestion d'erreur, traces : exemples avec log4J,
performances : mesures avec Junit,
les messages utilisateurs : les taglibs associées,
automatisation : génération des fichiers avec Ant : présentation
de l'outil, gestion des dépendances, exemples.

Evolutions

Autres frameworks MVC pour JEE : JSF, Tapestry, Wicket, et Struts
2
Présentation de Struts 2 : historique, architecture

Développement Web avec JSF

Durée: 3 jours
1410 €

du 20 au 22 Février
du 10 au 12 Avril

du 16 au 18 Juillet
du 22 au 24 Octobre

Public:

Développeurs Java.

Objectifs:

Comprendre le mécanisme des Java Server Faces, savoir les mettre en œuvre pour développer une application Web.

Connaissances préalables nécessaires:

Une bonne connaissance de Java est nécessaire, et plus précisément des technologies jsp, et servlets.

Programme:

Le standard JEE

L'architecture MVC : Model View Controller.

Rappel sur la méthode de construction d'une application avec les pages jsp et les servlets.

Présentation JSF

Les apports de JSF par rapport à Struts

Les différentes implémentations :

la référence Sun, Apache MyFaces, Smile, ECruiser

Architecture d'une application

Développement avec les JSF

Le cycle de vie d'une page JavaServer Faces

Les balises

Les composants JavaBeans : backing beans

Gestion des messages, des événements

Les fichiers de configuration, descripteurs de déploiement

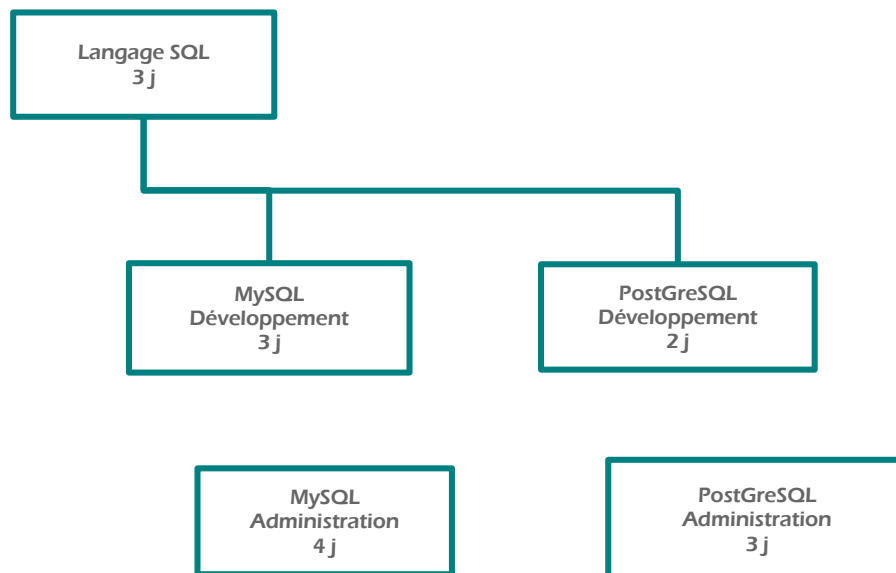
Définition des règles de navigation

Travaux pratiques avec Apache MyFaces et Tomcat

Intégration

Lien avec Struts Shale Framework.

Utilisation des pages JSF avec AJAX



Langage SQL

Durée: 3 jours
1350 €

du 17 au 19 Janvier

du 6 au 8 Mars

du 9 au 11 Mai

du 19 au 21 Juin

du 11 au 13 Septembre

du 16 au 18 Octobre

Public:

Analystes, développeurs, utilisateurs. Exploitants, administrateurs de bases de données.

Objectifs:

Maîtriser les fonctionnalités standards du langage SQL. Soumettre des requêtes en interactif et interpréter les résultats.

Connaissances préalables nécessaires:

Concepts relationnels

Programme:

Introduction

Rappel sur le modèle relationnel

Les composantes de SQL

Les tables

La norme SQL

La requête SELECT

Syntaxe générale. Projection de colonnes. Sélection de lignes.

Les différents types de prédicats.

Les expressions. Les fonctions. L'agrégation. Le tri.

Les tables temporaires .

La jointure

Syntaxe.

Inner join. Outer join .

Les requêtes de mise à jour

INSERT, UPDATE, DELETE.

Notion de transaction : COMMIT, ROLLBACK

Les requêtes imbriquées

Le Subselect simple, le Subselect corrélié

Les opérateurs ANY, SOME, ALL, EXISTS

Opérateur ensembliste

Union de Select

La définition des objets : Data
Definition Language

CREATE TABLE, CREATE INDEX

ALTER et DROP

L'intégrité référentielle

Les VUES

Création et utilisation

Mises à jour

MySQL : Développement

Durée: 3 jours
1380 €

du 6 au 8 Février
du 10 au 12 Avril
du 16 au 18 Juillet

du 15 au 17 Octobre
du 10 au 12 Décembre

Public:

Développeurs et toute personne souhaitant mettre en place une application avec une base de données MySQL

Objectifs:

Apprendre à créer des applications avec une base de données MySQL 5.0

Connaissances préalables nécessaires:

Il est demandé aux participants de connaître les notions de base sur SQL.

Programme:

Introduction	Présentation, historique, les versions MySQL (standard, Max, Pro, Classic), les licences (GPL et commerciale). Les composants Installation outils clients
Clients MySQL	Présentation de Query Browser Construction de requêtes Tests, comparaison et debugging et optimisation de requêtes
Procédures de chargement de données	Format des données à importer Les commandes disponibles : load, insert, Développement de scripts d'importation
DDL et DML	Les spécificités de MySQL dans le mise en œuvre des requêtes SQL.
Programmation serveur	L'intérêt de la programmation côté serveur, et les limites Mise en œuvre de déclencheurs (triggers) et programmation de procédures stockées Déclaration et utilisation de variables Gestion des curseurs Evolutions prévues dans les futures versions de MySQL.

MySQL : Administration

Durée: 4 jours
1830 € HT

du 16 au 20 Janvier
du 26 au 29 Mars
du 29 Mai au 1er Juin

du 17 au 20 Septembre
du 12 au 15 Novembre

Public:

Toute personne souhaitant configurer, installer et administrer une base de données MySQL

Objectifs:

Comprendre le fonctionnement, et savoir installer, configurer et administrer une base de données MySQL (le cours est réalisé sur une version 5 de MySQL).

Connaissances préalables nécessaires:

Il est demandé aux participants de connaître les notions de base sur SQL.

Programme:

Introduction	Présentation, historique, les versions MySQL (standard, Max, Pro, Classic), les licences (GPL et commerciale). Les composants du serveur MySQL. Caractéristiques: transactions, clusters.
Installation	Choix du produit à installer : les RPMs, le code compilé ou les sources. Installation, configuration. Les scripts fournis avec MySQL : démarrage du serveur, création des tables de droits d'accès, démarrage de multi-serveurs, ... Outils graphiques.
Sécurité	Système des privilèges : principe de fonctionnement, authentification, contrôle des droits pour les requêtes Gestion des comptes utilisateurs : création/suppression de comptes, limitation des ressources, sécurisation des accès Mise en place de SSL.
Les fichiers de logs	Les erreurs, les modifications du fichier ISAM, les requêtes
Optimisation	Les requêtes en cache. Les différents points à optimiser : les requêtes, les tables, la configuration du serveur, et de l'environnement.
Sauvegardes	Les tables MyISAM et InnoDB Utilitaire myisamchk : contrôler, réparer, optimiser Vérification sur base à l'arrêt. Réparation. Vérification/réparation en cours d'exploitation. Méthode de sauvegarde des données MySQL, script mysqldump, ou mysqlhotcopy
Haute Disponibilité	Mise en œuvre de la réplication

PostgreSQL : Développement

Durée: 2 jours
905 €

du 13 au 14 Février
du 16 au 17 Avril
du 2 au 3 Juillet

du 8 au 9 Octobre
du 17 au 18 Décembre

Public:

Développeurs souhaitant mettre en place une application avec PostgreSQL

Objectifs:

Apprendre à créer des applications avec une base de données PostgreSQL

Connaissances préalables nécessaires:

Il est demandé aux participants de connaître les notions de base sur SQL.

Programme:

Présentation de PostgreSQL	PostgreSQL : serveur de base de données relationnelle objet Historique, licence, mode de développement Les composants du serveur PostgreSQL: serveur, client, connecteurs jdbc, tcl, pl, python Les apports de la version 9.0
Installation et configuration	Installation de PostgreSQL Installation avec un système de paquetage
Interagir avec le serveur	Le concept du client serveur Le client PostgreSQL Utiliser les clients graphiques
Spécificités SQL de PostgreSQL	Spécificités DDL (Définition des objets) Spécificités DML (Manipulation des données) Optimisation des requêtes
Programmation coté serveur	Importation et exportation de données L'intérêt de la programmation côté serveur et ses limites Mise en œuvre de déclencheurs (triggers) et programmation de procédures stockées Déclaration et utilisation de variables Gestion des curseurs Mise en œuvre avec PL/pgsql et présentation des autres outils disponibles

PostgreSQL : Administration

Durée: 3 jours
1340 €

du 15 au 17 Février
du 18 au 20 Avril
du 4 au 6 Juillet

du 10 au 12 Octobre
du 19 au 21 Décembre

Public:

Toute personne souhaitant configurer, installer et exploiter une base de données PostgreSQL

Objectifs:

Comprendre le fonctionnement, et savoir installer, configurer et exploiter une base de données PostgreSQL.

Connaissances préalables nécessaires:

Il est demandé aux participants de connaître les notions de base sur SQL.

Programme:

Introduction	Présentation, historique, les versions PostgreSQL, les outils complémentaires et les licences. Les composants du serveur PostgreSQL: serveur, client, connecteurs jdbc, tcl, pl, python
Installation	Choix du produit à installer : les RPMs, le code compilé ou les sources. Installation. Configuration, organisation du répertoire /var/lib/pgsql, fichier postgresql.conf. Les scripts fournis avec PostgreSQL : démarrage du serveur, création des tables de droits d'accès, démarrage de multi-serveurs, ...
Sécurité	Système des privilèges : principe de fonctionnement, authentification, contrôle des droits pour les requetes Gestion des comptes utilisateurs : création/suppression de comptes, limitation des ressources, sécurisation des accès: fichier pg_hba.conf Mapping avec les utilisateurs systèmes: pg_ident.conf
Utilisation	Commande psql. Accès aux tables. Les commandes en \ pgadmin: installation, configuration.
Exploitation	Sauvegardes/Sauvegards contrôler l'état de la table, réparer, optimiser: pg_dump, pg_dumpall, vacuumdb Méthode de sauvegarde des données PostgreSQL Montée de niveaux: Copie de serveur à serveur, réplication. Tests de régression

Initiation à
OpenOffice Writer
2 j



Utilisation Avancée
OpenOffice Writer
2 j

Initiation à
OpenOffice Calc
2 j



Utilisation Avancée
OpenOffice Calc
2 j

Programmation
OpenOffice
4 j

Durée: 2 jours

Prix et dates: nous consulter

Public:

Personnes souhaitant utiliser le traitement de texte OOWriter.

Objectifs:

Savoir utiliser les fonctions de base du traitement de texte.

Connaissances préalables nécessaires:

Aucune connaissance préalable n'est nécessaire.

Programme:

Bases	Lancement, gestion des barres d'outils, les différents modes d'affichage. Utilisation de l'aide. Ouverture/création/enregistrement de fichiers. Saisie et modifications de texte : insertion, copie, annulation, caractères spéciaux, recherche et remplacement de texte
Mise en forme	caractères: polices, effets de caractères, ... paragraphe: alignement, enchaînement, retraits, écarts, bordures pages : en-têtes, pieds de pages, ...
Styles	Les styles prédéfinis, le styliste Le catalogue Création de nouveaux styles
Quelques fonctions avancées	Les tableaux : insertion, manipulation des colonnes et lignes, scinder/fusionner les tableaux. Vérificateur d'orthographe. Insertion de champs particuliers: date, heure, numéros de page, caractères spéciaux

Durée:2 jours

Prix et dates: nous consulter

Public:

Personnes souhaitant approfondir ses connaissances sur le traitement de texte OpenOffice Writer.

Objectifs:

Savoir utiliser les fonctions avancées du traitement de texte.

Connaissances préalables nécessaires:

Savoir utiliser un traitement de texte, de préférence, OOWriter.

Programme:

Gestion de documents	Modèles : création, catalogue Document maître : insertion, déplacer, actualiser Numérotation des titres, des chapitres Table des matières et Index Les liens hypertextes Notes de bas de pages, repères de texte. Gestion des versions de document.
Outils	Vérification orthographique, Coupure de mots, AutoCorrection, AutoFormat Dictionnaire des synonymes
Calculs et graphiques	Valeurs numériques, formules Insertion/modification de graphiques
Dessins	Outils de dessin, insertion images, sons Gestion des cadres et des objets : ancrage, habillage, contour, ...
Mailing	Principe, base de données, source de données, impression
Exploitation	Importations/Exportations: formats Word, HTML. Exportations PDF. Personnalisation. Macros instructions.

Initiation à OpenOffice Calc

Durée: 2 jours

Prix et dates: nous consulter

Public:

Personnes souhaitant utiliser un tableur et désirant connaître les grands principes du tableur OpenOffice pour Linux.

Objectifs:

Savoir utiliser le tableur OpenOfficeCalc pour les opérations courantes.

Connaissances préalables nécessaires:

Aucune connaissance préalable n'est nécessaire.

Programme:

Bases	<p>Lancement, gestion des barres d'outils. Ouverture/création/sauvegarde de fichiers. Utilisation de l'aide. Définitions: classeur, feuille de calcul, cellules. Déplacement dans une feuille : recherche d'une cellule, ... Manipulations sur différentes feuilles : déplacement, Affichage : partiel, plein écran, masquer une barre d'outil, ... Sélection : des cellules, des lignes, des colonnes</p>
Saisie	<p>Le contenu d'une cellule : texte, valeurs, formules Création de séries de données, saisie semi-automatique Effacer, supprimer, copier, trier des données. Nommer les zones : sélection de plages de cellules, accès par le nom de la zone.</p>
Calculs simples	<p>Syntaxe d'une formule. Références relatives/absolues Utilisation des assistants pour créer des formules. Somme, moyenne, formules statistiques de base</p>
Mise en forme	<p>Adapter les largeurs de colonnes, hauteurs de lignes, masquer des lignes et colonnes, Format des valeurs numériques, dates et heures, alignement, etc ...</p>
Impression	<p>Mise en page, options.</p>

Utilisation Avancée OpenOffice Calc

Durée: 2 jours

Prix et dates: nous consulter

Public:

Personnes souhaitant utiliser le tableur OoCalc.

Objectifs:

Savoir maîtriser les fonctions du tableur OO Calc.

Connaissances préalables nécessaires:

Connaissance d'un tableur, de préférence OO Calc.

Programme:

Fonctions	Présentation des différentes fonctions disponibles : Classeur, Information, Mathématiques, Financières
Graphiques	Définition d'un graphique : zone de données, échelle, étiquettes, titre, légende Graphique en 3D.
Listes de données	Utilisation du filtre standard, AutoFiltre. Pilotes de données.
Styles	Créer, appliquer un style, modèle de classeur
Exploitation	Protection d'un classeur, des cellules. Gestion des versions. Importations/Exportations: format MS-Excel, ASCII, CVS, SYLK. Options d'importations. Les option spécifiques du tableur Macros instructions simples

Programmation OpenOffice

Durée: 5 jours

Prix et dates: nous consulter

Public:

Toute personne souhaitant Programmer en OOBASIC, ou transformer des macros Visual Basic pour la suite OpenOffice.

Objectifs:

Maîtriser la programmation avec OO Basic.

Connaissances préalables nécessaires:

Les notions de programmation sont indispensables, ainsi qu'une bonne connaissance du tableur Calc et du traitement de texte Writer

Programme:

OOBasic	Présentation. Caractéristiques : OOBASIC Langage Objet , classification par collections. Structures de données. Services : ouverture et enregistrement des fichiers, les filtres texte communs et les documents vierges
Classeurs	Ouverture, fermeture, sauvegarde de classeurs vierges et existants.
Feuilles	Accès nommé et indexé, ajout, recopie, déplacement et suppression de Feuilles. Activation, protection et renommage.
Cellules	Accès, Ajout, recopie, déplacement et suppression de plages de cellules nommées. Accès aux cellules par position, adresses des cellules. Propriétés des cellules, contenu des cellules. Nombres et formules, Textes et Curseurs. Formatage, formats numériques
Les tableaux	Copie ou déplacement en bloc. Propriété "DataArray". Transfert d'une cellule dans un bloc (valeur et formule). Tri et Descripteur du tri. Filtrage et descripteur d'extraction. Totalisations et Descripteur de totalisation. Recherche et remplacement de caractères : les descripteurs associés.
Fonctions utilitaires	Effacer le contenu des cellules. Insérer ou supprimer des cellules, des lignes ou des colonnes. Autres Fonctions (mathématiques, financières). Le texte et les curseurs.
Interface de saisie	Boîtes de Dialogues et l'EDI, les contrôles. Exemple de gestion des événements. Configuration par programme. Imprimer, sélection multiple. Configuration Calc
Utiliser la documentation	XRAY appliqué à Calc, recherche de propriétés et de méthodes SDK : les informations disponibles

Conditions de vente

Tous les prix indiqués dans ce catalogue sont hors taxes.

Toute inscription à une formation implique l'adhésion des présentes conditions de vente.

L'inscription est ferme à partir de la signature de la convention. Pythagore F.D. se réserve la possibilité d'annuler une session si le nombre de stagiaires est insuffisant. Pythagore F.D. informera le client au minimum dix jours ouvrables avant le début du stage. Le client peut alors reporter ou annuler son inscription.

Annulations

Toute formation commencée est intégralement due. Si l'annulation d'une inscription à un stage inter-entreprise, est faite dans la période allant du 10^e jour ouvré au 2^e jour ouvré avant le début du stage, 50% du montant des frais de formation sont dus. Si l'annulation n'a pas été faite 48h avant le début de la formation, la totalité du montant des frais de formation reste due. Dans le cas de l'annulation d'une formation ou d'un cycle de formation en intra moins de dix jours ouvrables avant le début de la formation, 50% des frais de formation restent dus.

Déroulement du stage

Tout stagiaire doit se conformer au règlement du centre de formation de Pythagore F.D. A défaut, le stagiaire pourra être exclus de la formation. Les frais de formation sont alors intégralement dus. Pendant la formation, les stagiaires restent les employés du client qui est responsable de leurs faits et gestes en application de l'article 1384 du code civil.

Paiement

Les factures sont payables, sans escompte, à réception pour les stages en inter-entreprises. Les cycles de formation d'une durée supérieure à un mois font l'objet d'une facturation mensuelle. En cas de non-paiement à son échéance, toute somme due portera intérêt de plein droit au taux d'une fois et demie le taux d'intérêt légal.

Litiges

Tout litige qui ne pourra être réglé à l'amiable sera du ressort du tribunal de Commerce de Paris.

Pythagore F.D. - 11, rue du Faubourg Poissonnière 75009 PARIS
Tél : 01 55 33 52 10 - Fax : 01 55 33 52 11
S.A.S au capital de 40 000 Euros - RCS Paris B 398 145 474

**Pour toute information, appelez-nous au
01 55 33 52 10
www.Pythagore-fd.fr**

Table des matières

Filières Systèmes

Linux/unix introduction	UX100	6
Le Shell	UX002	8
Shell avancé	UX004	9
Administration UNIX multi-plateformes	UX011	11
Administration AIX	UX018	13
Administration AIX avancée	UX019	15
Administration Solaris 9 et 10	UX040	17
Administration avancée Solaris 9 et 10	UX041	19
Introduction au logiciel libre	UX107	21
Administration Linux	UX111	22
Les services réseaux Linux	UX112	25
Haute disponibilité Linux	UX115	27
Linux système sécurisé	UX117	29
Linux sécurité des accès	UX118	31
Linux : optimisation performances métrologie	UX119	33
Administration avancée Linux	UX140	34
Administration Ubuntu Server	UX142	37
Virtualisation Linux	UX123	40
Virtualisation avec KVM	SY007	41
Virtualisation avec Xen	SY004	43
Virtualisation postes de travail	SY005	45
Virtualisation serveurs	SY006	46
Préparation certification LPI101	UC101	48
Préparation certification LPI102	UC102	49
Préparation certification LPI201	UC201	50
Préparation certification LPI202UC202		51

Filières Systèmes embarqués et mobilité

Linux temps réel	X121	54
Systèmes Linux embarqués	UX125	55
openWrt	UX126	57
Développement d'applications Android	UX128	59
Industrialisation Android	UX131	61
Developpement pour iPhone et iPad	MC020	62

Filières Réseaux et TCP/IP

Introduction aux réseaux	RS001	65
TCP/IP : protocoles et mise en œuvre	RS003	67
Administration réseaux IP	RS007	69
Routage IP avancé	RS009	70
IP: Migration vers IPv6	RS014	71
Messagerie	RS006	73
Sécurité TCP/IP	RS018	74
Installation et Configuration WIFI	RS021	75
Bus One-wire	RS125	76
Mise en œuvre du protocole SNMP	RS022	77
Voix sur IP	RS023	79
Voip - Configuration Asterisk	RS024	81
Performances et optimisation des réseaux	RS025	82
Annuaire LDAP	RS122	83
Supervision nagios : utilisation	RS129	85
Administration Nagios	RS130	87
Reporting avec Cacti	RS134	90
Zimbra : installation et configuration	RS132	91
Gestion de Parc avec OCS et GLPI	UX124	92

Filières Internet et Web

Cloud : technologies et enjeux	RS133	95
Architecture cloud d'entreprise	RS131	96
Serveur WEB : apache	RS104	98
Administration serveur Nginx	RS105	100
Serveur Proxy Squid	RS114	102
Sécurisation des services internet	RS115	103
HTML	RS102	105
Web Dynamique avec JavaScript	RS106	106
CMS : mise en œuvre de Joomla	RS127	107
Présentation Web 2.0	RS128	108
Développement Web avec Ajax	RS109	109
Google Apps Utilisation	AW002	110
Google App Engine	LW003	111
Développement Web avec GWT	LW001	112

Filières Développement

XML Développement	AS120	114
Programmation en langage C	LC001	115
Perfectionnement en langage C	LC002	117
Programmation système en C sur Unix/Linux	LC010	119
Programmation noyau et drivers en C sur Linux	LC011	122
Langage C pour TCP/IP	RS016	123
Debugging en C sur Unix	UX015	124
Le langage C++	LC004	125
Introduction à Objective-C	LC005	126
Programmation Cocoa avancée Mac et iPhone	LC006	127
Subversion utilisateur	LD002	129
Subversion Administrateur	LD003	130
Programmation PHP	LH001	132
Initiation au langage Pascal	PA001	134
Programmation Perl	LP001	135
Perl avancé	LP002	137
Le langage Python	LY001	138
Développement avancé avec Python	LY002	140

Filières Langages Java et JEE

Programmation Java	LJ001	143
Programmation avancée en Java	LJ002	145
Sécurité Java et JEE	LJ005	147
JEE : les servlets et JSP	AS004	149
JEE : les EJB	AS005	150
Développement Web Services	AS006	152
Techniques et outils de migration vers les EJB3	AS007	154
Serveurs d'application JEE	AS011	155
Développement JEE avec WebSphere	AS101	157
Développement JEE avec JOnAS	AS103	159
Applications JEE avec JBoss	AS104	161
Conception JEE avec les Design Patterns	AS122	163
Administration WebSphere	AS131	164
Administration Tomcat	AS133	166
Administration JBoss	AS134	168
Administration WebSphere avancée	AS135	170
Exploitation JOnAS	AS136	172
JBoss Portal	AS137	174
Clusters JBoss	AS144	176
Développement Web avec Struts	AS150	178
Développement Web avec JSF	AS151	180

Filières Bases de Données

Langage SQL	BD001	183
MySQL : Développement	BD010	184
MySQL : Administration	BD011	185
PostgreSQL : Développement	BD020	186
PostgreSQL : Administration	BD021	187

Filières bureautiques

Initiation à OpenOffice Writer	OW001	189
Utilisation Avancée OpenOfficeWriter	OW002	190
Initiation à OpenOffice Calc	OC001	191
Utilisation Avancée OpenOffice Calc	OC002	192
Programmation OpenOffice	OP001	193