

Catalogue des formations 2014

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

Pythagore F.D. : Apprendre à Apprendre

Nos domaines d'expertise :

- **Unix** (AIX, HP-UX, Solaris), et **Linux** (Debian, Mandriva, RedHat, Fedora, Suse) et les applicatifs Apache, Openldap, Samba, Squid, Nagios, OCS/Glpi, ...
- la **virtualisation** avec xen, kvm, lxc, et le **cloud** : eucalyptus, cloudstack et openstack
- **TCP/IP** : architecture, sécurité, administration de réseaux IP, la VoIP, ...
- **Java**, les serveurs d'applications **JEE** (Jonas, JBoss, WebSphere, Tomcat, ...)
- la programmation (langage C, technologies Jee, WebServices, PHP, Perl, Python, ..) et le développement sur mobiles android, iPhone et iPad.

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, l'industrialisation d'applications android, ...

Tous nos stages sont l'occasion de nombreuses mises en pratique et exercices de manière à permettre aux participants de bien assimiler les nouveaux concepts.

Nouveautés 2014 :

L'arrivée de solutions de **Bigdata** fiables et performantes aura un impact très important sur les équipes informatiques, qu'il s'agisse des développeurs, ou des intégrateurs et administrateurs.

Concernant les bases de données, par exemple, les technologies NoSQL remettent en cause la plupart des méthodes de travail des gestionnaires de bases de données.

Du côté des développeurs, la mise en oeuvre de concepts comme MapReduce, permet d'optimiser les temps de traitement.

Enfin, les intégrateurs et administrateurs qui utilisent l'ensemble des possibilités offertes par la virtualisation et le cloud pourront répondre de manière optimisée aux contraintes de performances, de haute disponibilité et de fiabilité souhaitées par les clients.

Nous avons choisi de construire une filière complète sur le **BigData**, depuis la présentation de l'architecture avec un stage intitulé « BigData : architecture et technologies », jusqu'à la programmation **MapReduce** avec **Hadoop**, en passant par l'indexation de données par **Lucene/Solr** et une formation « Introduction aux bases de données **NoSQL** ».

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, 8 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 ...

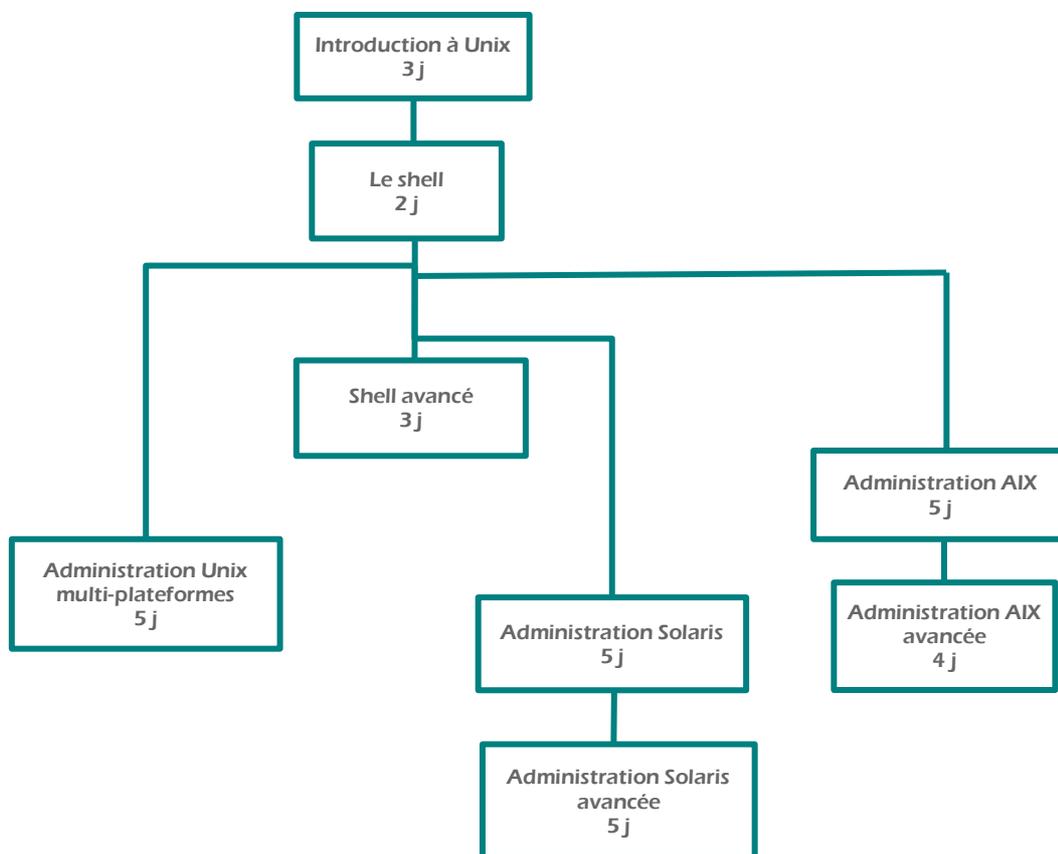
Cette solution, parfaitement adaptée aux formations techniques permet, par exemple, d'organiser des sessions multi-sites pour des entreprises ou organisations dont le personnel est réparti géographiquement sur plusieurs sites.

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

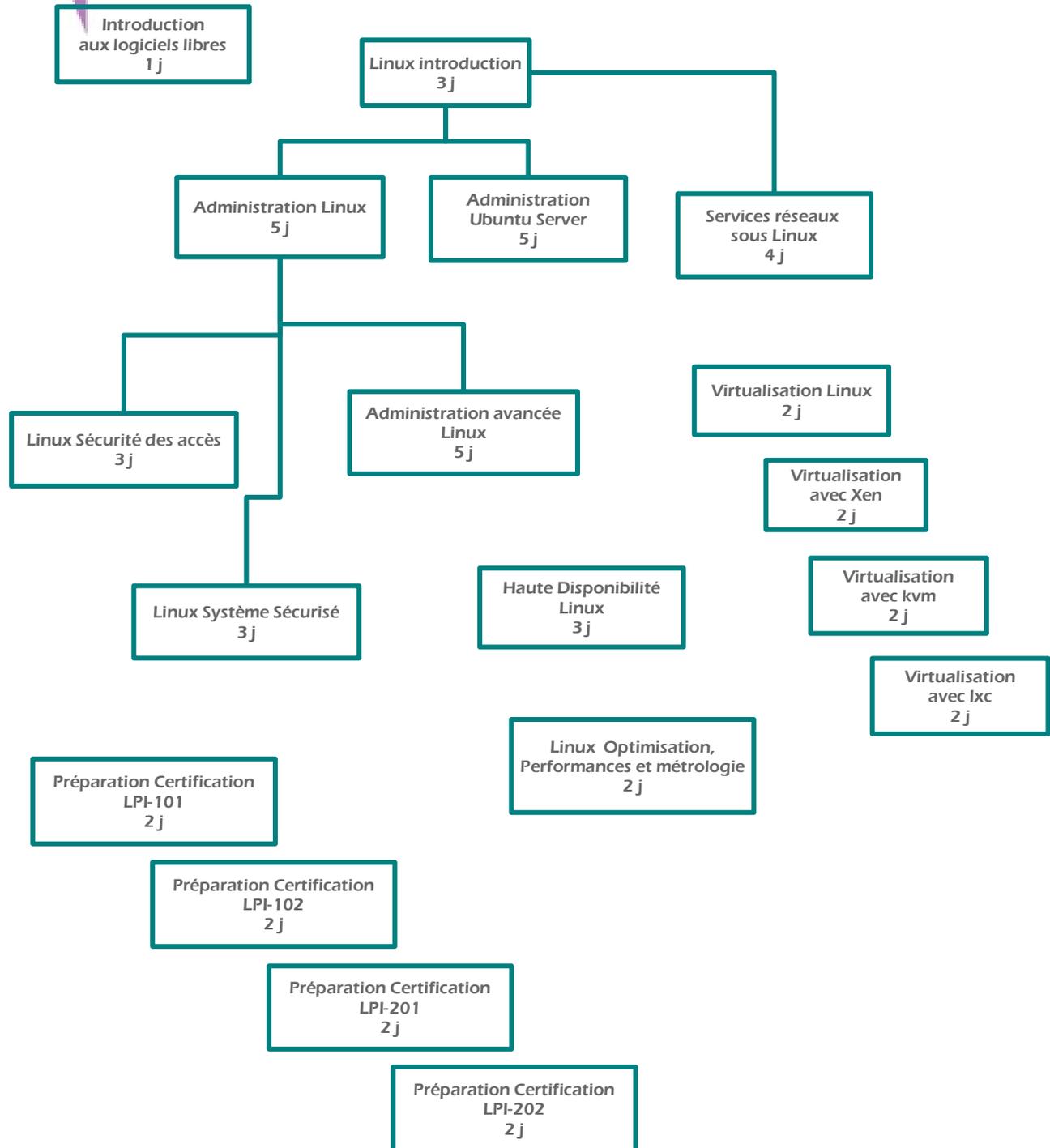
N'hésitez pas à nous contacter pour organiser vos classes virtuelles !

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
1440 €

20 au 22 janvier
10 au 12 mars
12 au 14 mai
16 au 18 juin

21 au 23 juillet
1er au 3 septembre
13 au 15 octobre
17 au 19 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 oeuvre 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 oeuvre 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
1010 €

23 au 24 janvier

13 au 14 mars

15 au 16 mai

19 au 20 juin

24 au 25 juillet

4 au 5 septembre

16 au 17 octobre

20 au 21 novembre

Public:

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

Objectifs:

Maîtriser la programmation en bourne Shell.

Connaissances préalables nécessaires:

Connaissance des principes de base du système UNIX.

Programme:

Généralités	Présentation du shell : interpréteur de commande Unix. Modes d'exécution d'un script. Les alias. Les méta-caractères.
Les variables	Portée des variables : locales, globales, environnement Les différents types de variables. Définir et manipuler des variables. Transmission de paramètres.
Fichiers d'environnement	Fichier d'initialisation général : .profile Fichier d'initialisation local : .kshrc
Les entrées/sorties	Accès en lecture/ écriture.
Structures de contrôle	Les instructions test et expr. Expressions conditionnelles Gestion des boucles Boucles for, while, until Tests 'if', tri avec 'case' Sortie de boucles avec break, continue et exit

Shell avance

Durée: 3 jours
1470 €

27 au 29 janvier

17 au 19 mars

19 au 21 mai

23 au 25 juin

8 au 10 septembre

24 au 26 novembre

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:

L'exécution des procédures	<ul style="list-style-type: none"> Programmation parallèle Programmation événementielle Gestion des alertes, gestion des travaux Exploitation
Les structures	<ul style="list-style-type: none"> Les fonctions Structuration des scripts Shell Auto-programmation - batch
Expressions régulières	<ul style="list-style-type: none"> Rappel de l'utilisation de la commande grep Les méta-caractères des expressions régulières Mise en oeuvre des expressions régulières avec grep Les possibilités de egrep
Les chaînes de caractères	<ul style="list-style-type: none"> Les expressions de variables Affectation de chaînes avec typeset (ksh) Les commandes basename et dirname
La gestion de fichiers	<ul style="list-style-type: none"> Mise à jour d'un fichier avec la redirection Définition de séparateur de champs : IFS Rediriger les entrées-sorties de tout un script Lecture d'un fichier dans une boucle Rediriger les entrées-sorties en ksh Tester les entrées-sorties standards
Exploitation	<ul style="list-style-type: none"> Quelques commandes utiles eval : réinterprétation d'une commande select : gestion de menus (ksh) getopts : décoder les options d'un script
Le filtre sed	<ul style="list-style-type: none"> Principe de fonctionnement de sed Les commandes de sed Utilisation des expressions régulières dans sed Les sous-expressions

Shell avance

La commande awk

Principes de fonctionnement de awk
Structure d'un programme awk
Les critères
Les variables prédéfinies
Les variables et les expressions
Les tableaux. Les instructions
Les fonctions prédéfinies
Les fonctions utilisateurs

Administration UNIX multi-plateformes

Durée: 5 jours
2135 € HT

3 au 7 février
14 au 18 avril
2 au 6 juin

29 septembre au 3 octobre
8 au 12 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.
Services d'impression	Lancement/arrêt des services d'impression. déclaration d'une imprimante. /etc/printcap gestion des files d'attente

Administration UNIX multi-plateformes

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 oeuvre de rrdtool. Création de graphes.

Administration AIX

Durée: 5 jours
2135 € HT

10 au 14 février
12 au 16 mai

7 au 11 juillet
6 au 10 octobre

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
Réseau	Les différents paramètres à prendre en compte. Configuration de l'interface. Configuration du nom de l'adresse du système

Administration AIX

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
1700 €

18 au 21 mars

30 septembre au 3 octobre

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 oeuvre.
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 oeuvre 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. 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 VVM 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 11

Durée: 5 jours
2135 €

13 au 17 janvier
3 au 7 mars

21 au 25 juillet
20 au 24 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. Principaux apports de Solaris 11.
Installation	Installation de Solaris à partir d'un dvd ou du réseau. Ajout de logiciel : gestion des packages et des dépôts. 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. 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 snapshots ZFS
Ordonnancement	At, cron, fichier crontab

Administration Solaris 11

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

Durée: 5 jours
2170 €

27 au 31 janvier
17 au 21 mars

1er au 5 septembre
3 au 7 novembre

Public:

Administrateurs de systèmes Solaris.

Objectifs:

Savoir installer, configurer, administrer un parc de systèmes Solaris. 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. Ai installer : principe de fonctionnement et configuration. Travaux pratiques : configuration d'un serveur d'installation automatisée.
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. Gestion des ressources.
Configuration systeme avancée	/etc/system modload/modunload, Swap Space, Core Dumps
Réseau	Configuration ip : nwaldm, ipadm, dladm
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.
Analyse de performances	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

Introduction au logiciel libre

Durée: 1 jour
485 € HT

24 janvier
28 mars
16 mai

12 septembre
21 novembre

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
2135 €

10 au 14 février
31 mars au 1er avril
30 juin au 4 juillet

15 au 19 septembre
20 au 24 octobre
1er au 5 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, s'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... 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....</p> <p>Travaux pratiques : Installation et configuration de samba pour le partage de fichiers.</p>

Les services réseaux Linux

Durée: 4 jours
1840 €

17 au 20 février
22 au 25 avril

15 au 18 septembre
3 au 6 novembre

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 Tcpcmdump 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 oeuvre 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
1495 €

24 au 26 février
26 au 28 mai

29 septembre au 1er octobre
12 au 14 novembre

Public:

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

Objectifs:

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

Connaissances préalables nécessaires:

Une bonne connaissance d'un système Unix et des réseaux IP 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 défaillants.</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 ldirectord</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 oeuvre 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. Les systèmes de fichiers haute disponibilité : DRDB (Distributed Replicated Block Device) Fonctionnalités, installation et configuration. Cluster Active/Hot standby avec ext3. Cluster Active/Active avec gfs. Export de gfs par gndb. Intégration avec heartbeat.</p>

Linux système sécurisé

Durée: 3 jours
1495 €

3 au 5 mars
2 au 4 juin

25 au 27 août
6 au 8 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 oeuvre 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 oeuvre 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
1495 €

10 au 12 mars
16 au 18 juin

13 au 15 octobre
8 au 10 décembre

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 oeuvre 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	OpenVAS (Open source Vulnerability Assessment Scanner) : principe de fonctionnement, installation. Travaux pratiques : réalisation d'un audit réseau avec openVAS.

Linux sécurité des accès

Sécurisation des accès réseau

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.

VPN , tunnels, iptables

Définitions : DMZ, coupe-feux, proxy.
 VPN et tunnels.
 Principe de fonctionnement.
 Présentation des tunnels chiffrés.
 Travaux pratiques :
 mise en oeuvre 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.

Proxy Squid

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.

Linux : optimisation performances métrologie

Durée: 2 jours
1050 €

13 au 14 mars
19 au 20 juin

16 au 17 octobre
11 au 12 décembre

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.
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
2170 €

24 au 28 février
14 au 18 avril

22 au 26 septembre
17 au 21 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 dépendances.

Détails de la commande rpm, et de ses options.

Travaux pratiques :

mise en oeuvre, 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 oeuvre, administration compatibilité NFS

Ext3, ext4 : caractéristiques et mise en oeuvre.

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 oeuvre. 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 oeuvre 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 `/etc/fstab`,
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 `make`, recompilation.

Périphériques

Périphériques non standards.
Installation de modules: `modprobe`, `insmod`.
Le répertoire `hotplug`.

Administration Ubuntu Server

Durée: 5 jours

Prix et dates: nous consulter

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, cpuinfo, ...</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

Préparation certification LPI101

Durée: 2 jours
1075 €

10 au 11 mars

8 au 9 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 paquets	Concevoir un schéma de partitionnement du disque dur Installation d'un gestionnaire de démarrage Gestion des bibliothèques partagées Utilisation du gestionnaire de paquets Debian et des gestionnaires de paquets 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

Preparation certification LPI102

Durée: 2 jours
1075 €

12 au 13 mars

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 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	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	Installation et configuration de X11. Configurer un gestionnaire d'affichage (Display Manager) Accessibilité.
Tâches d'administration	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	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	Notions élémentaires sur les protocoles Internet. Configuration réseau de base. Résolution de problèmes réseau. Configuration DNS sur les postes de travail.
Sécurité	Tâches d'administration de sécurité. Configuration de la sécurité du système. Sécurisation des données avec le chiffrement.

Préparation certification LPI201

Durée: 2 jours
1075 €

17 au 18 mars

15 au 16 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.

Preparation certification LPI202

Durée: 2 jours
1075 €

19 au 20 mars

17 au 18 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.

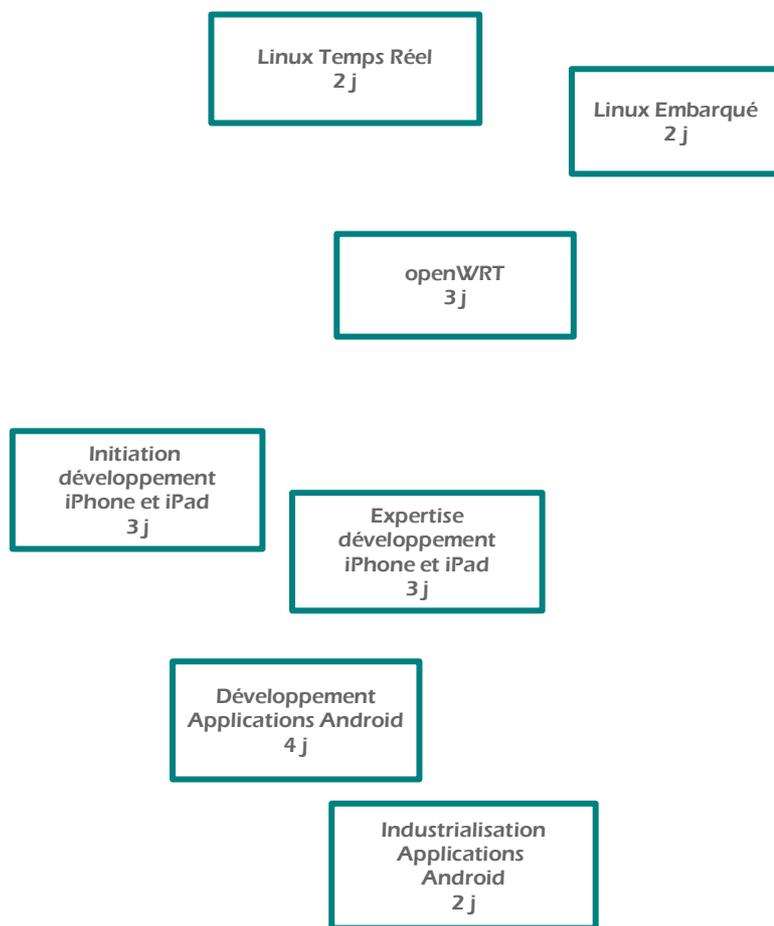
Connaissances préalables nécessaires:

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
1080 € HT

27 au 28 février
26 au 27 juin

23 au 24 octobre
18 au 19 décembre

Public:

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

Objectifs:

Connaître les principes des systèmes embarqués Linux et les différentes solutions disponibles, savoir mettre en oeuvre 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, android. Architectures et caractéristiques. Critères de choix : coût.
Construction	Les différentes étapes pour construire un système embarqué. Installation d'une chaîne de compilation croisée : ct-ng, codesourcery. 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.

Systemes Linux embarques

Dveloppement

Ajout de modules au noyau.

Travaux pratiques : Portage de netsnmp.

Compilation d'une MIB spcifique au matriel et intgration dans la distribution embarquee.

Mise en oeuvre dans un environnement multi-plateformes.

Centralisation des journaux syslog. Supervision par Nagios.

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

Visualisation des performances.

openWrt

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. Busybox. 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 oeuvre 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 oeuvre sur de la répartition de charge HTTP.
Frontal à un cluster JBoss. Visualisation des performances.

Développement d'applications Android

Durée: 4 jours
1830 €

17 au 20 mars
23 au 26 juin

25 au 28 août
20 au 23 octobre
8 au 11 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 la programmation en java.

Programme:

Introduction	Présentation du système d'exploitation Android et historique des versions : de la v1 jusqu'à la v4. Les apports de Jelly Bean. 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 oeuvre du MediaRecorder.

Développement d'applications Android

API	Accès réseau, accès au système de fichiers. Mise en oeuvre des APIs réseau. Capteurs internes. Gestion des périphériques : carte son, écran, caméra, clavier,.. Mise en oeuvre de la classe Sensor. Ecriture d'un code exploitant les capteurs de position.
Interactions d'applications	Principe de réutilisation d'extraits applicatifs. Ecriture d'une application exploitant GoogleMaps. Déclaration des possibilités applicatives: les intents filters. Exécution asynchrones, tâches de fond, tâches répétitives
Communications inter-applicatives avec AIDL	Communication multi-canaux et bi-directionnelles

Industrialisation Android

Durée: 2 jours
975 €

28 au 29 avril
7 au 8 juillet

9 au 10 octobre
4 au 5 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. Les apports de Jelly Bean.
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 "Google Play". 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

Initiation au développement iPhone/iPad

Durée: 3 jours

Prix et dates: nous consulter

Public:

Développeurs sur mobiles iPhone/iPad

Objectifs:

Connaître les possibilités du développement sur les devices iOS (iPhone, iPod touch, iPad), maîtriser le langage Objective-C et l'architecture Cocoa-Touch, explorer 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 Les principaux "design patterns" Le modèle MVC (model, view, controller) Caractéristiques des langages statiques et dynamiques Présentation du langage Objective-C Le modèle "retain-release" pour la gestion mémoire Le modèle ARC pour la gestion mémoire
Outils de développement	Les grandes phases du projet : architecture, coding, documentation, testing, debugging, optimisation, publication Présentation d'Xcode et des éléments 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 d'application et navigation	Retour sur le MVC, notion de ViewController 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

Initiation au développement iPhone/iPad

iPad

Présentation de l'iPad : similitudes et différences
Création d'une application iPad, pas à pas, utilisant différentes technologies : SplitView, WebView, parsing d'un flux RSS

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 de la documentation
Outils tierces parties
Livres et références, sites utiles

Expertise Développement iPhone iPad

Durée: 3 jours

Prix et dates: nous consulter

Public:

Développeurs sur mobiles iPhone/iPad maîtrisant déjà les bases du framework Cocoa et du langage Objective-C.

Objectifs:

Aller plus loin dans la maîtrise du langage Objective-C et de Cocoa, explorer d'autres APIs.

Connaissances préalables nécessaires:

Une bonne maîtrise du framework et du langage Objective-C est requise, ainsi qu'une expérience de la programmation objet. Il est conseillé de suivre d'abord la formation "Initiation"

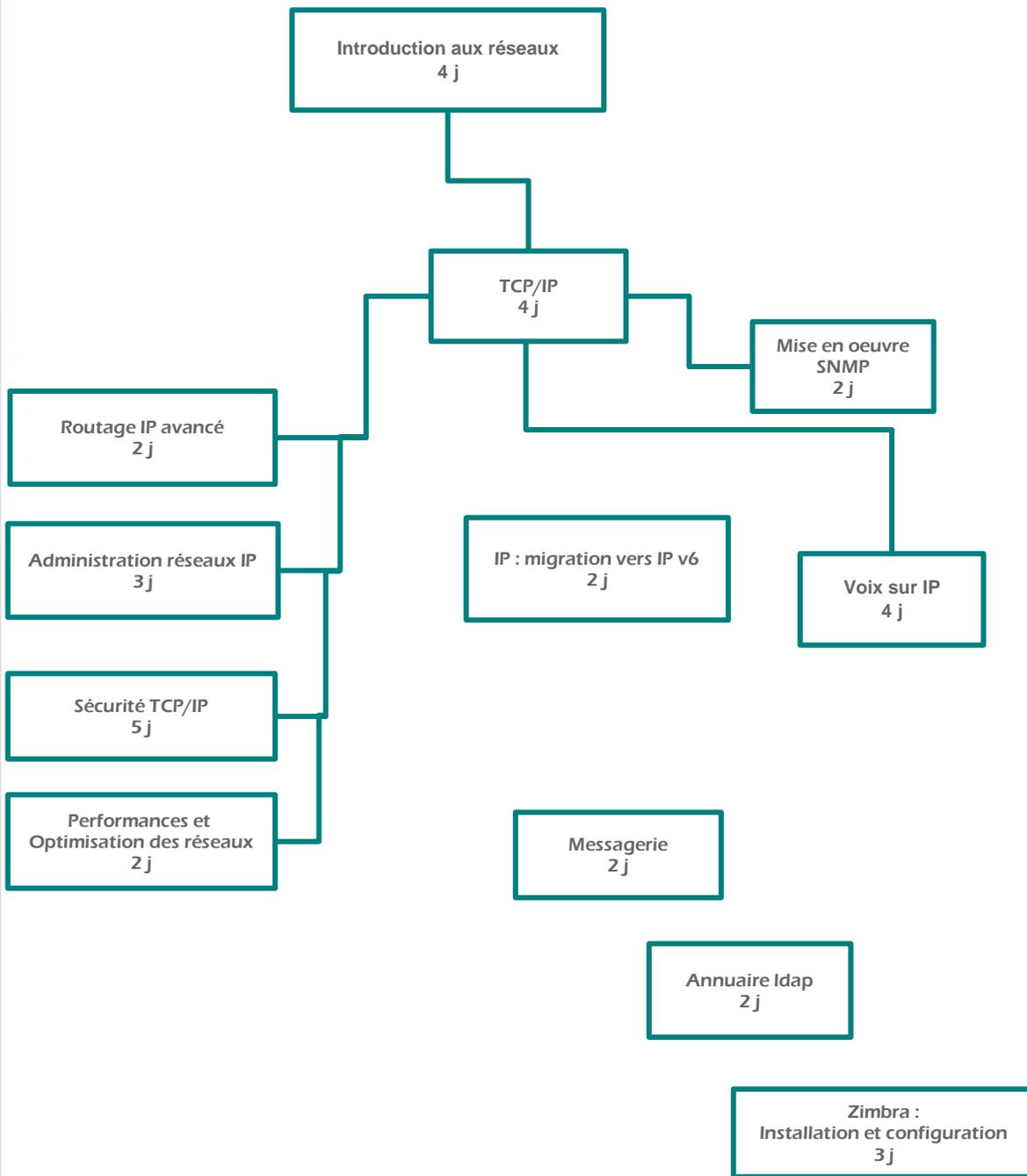
Programme:

Le framework Cocoa	Retour sur les "design patterns" Retour sur l'architecture d'une application
Objective-C	Présentation du runtime (catégories, protocoles) Les ajouts d'Objective-C 2.0 Les mécanismes de délégation et de notification Les blocks et les API modernisées Présentation de Clang et de l'analyseur statique
La WebView	Création d'un navigateur web Utilisation des fonctions standards de la WebView Utilisation de fonctions avancées (filtrage, communication entre JavaScript et Objective-C)
Application multilingue	Préparation d'une application pour l'international Gestion du texte, des ressources Utilisation des formats locaux (nombres, dates, monnaies, calendriers) Guidelines sur la localisation, méthodologie
Gestion des images	Utilisation de l'appareil photo et de la caméra Stockage des images dans la bibliothèque ou la pellicule de l'utilisateur Stockage sous forme de fichier JPG ou PNG Enrichissement dynamique des objets, exemple d'utilisation d'une catégorie
Carnet d'adresses	Création d'une application utilisant le carnet d'adresses Retour sur les ViewControllers et les vues de détail Intégration en Objective-C d'une API en C, impact sur la gestion mémoire

Expertise Développement iPhone iPad

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 Enregistrement vidéo, extraction d'une image "poster" Accès à des ressources sur Internet (pseudo-streaming, true-streaming)
Core Graphics	Présentation de Core Graphics Examen de l'API en C, intégration dans le framework Objective-C Exemples de dessin en 2D avec gestion du contexte graphique
Core Animation	Présentation des différentes API d'animation Les transitions simplifiées Déplacement des objets, intégration dans l'UI standard Transformations 2D (changement de taille, ...), et 3D (rotation dans l'espace, mise en perspective, ...)
L'accéléromètre	Présentation de l'accéléromètre Gestion avancée de la rotation et des transitions animées Création d'une application en pas à pas
Debugging et optimisation	Présentation des outils et procédures Analyse dynamique du flux de code (Shark) Examen de l'application via le runtime (Instruments) Utilisation avancée du debugger

Filières Réseaux et TCP/IP



Introduction aux réseaux

Durée: 4 jours
1940 €

13 au 16 janvier
24 au 27 mars
12 au 15 mai

8 au 11 septembre
13 au 16 octobre
1er au 4 décembre

Public:

Toute personne souhaitant acquérir des connaissances générales sur les réseaux, et plus particulièrement sur la mise en oeuvre 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, verification 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 oeuvre

Durée: 4 jours
1690 €

3 au 6 février
22 au 25 avril
10 au 13 juin

15 au 18 septembre
16 au 19 décembre

Public:

Toute personne souhaitant mettre en oeuvre 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 oeuvre 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 oeuvre : 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

TCP/IP : protocoles et mise en oeuvre

Sécurité

Ipssec (IP Security Protocol)
TP de mise en oeuvre

Administration réseaux IP

Durée: 3 jours
1440 €

24 au 26 février
5 au 7 mai

22 au 24 septembre
17 au 19 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 oeuvre de RIP sur différents routeurs, visualisation par traceroute.
DNS	Définition. Fonctionnement. Configuration. Mise en place de plusieurs serveurs DNS primaires 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 oeuvre 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, ebttables, 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 oeuvre 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
985 €

27 au 25 février
7 au 8 avril

15 au 16 septembre
24 au 25 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 oeuvre de l'entête calipso), confidentialité et entête d'authentification. Problème des entêtes noeud-par-noeud. 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 oeuvre du routage IPv6 en mode statique. Activation du mode automatique, visualisation des tables de routage obtenues.

IP: Migration vers IPv6

Gestion des adresses	<p>Mode sans état, avec état.</p> <p>DHCPv6 : Présentation. Mise en oeuvre d'un serveur dhcpv6. Cohabitation avec IPv4. Attribution statique d'adresses. Gestion du DUID.</p> <p>Stateless Address Autoconfiguration (SAA) : Utilisation de radvd en complément. Relais DHCPv6.</p> <p>Cycle de vie des adresses. Adressage aléatoire. Migration d'opérateurs.</p> <p>Choix de l'adresse client.</p>
DNSv6	<p>Mise en oeuvre d'un DNS v4/v6.</p> <p>Les différentes implémentations. DNS dynamiques.</p>
Migration v4/v6	<p>Les différentes approches : double pile, encapsulation statique, encapsulation dynamique.</p> <p>Impacts de la suppression du NAT.</p> <p>Utilisation du cycle de vie des adresses.</p> <p>Les différents tunnels. Mise en oeuvre d'un tunnel 6sur4.</p> <p>Présentation de l'encapsulation v6 dans v4 et l'extension Teredo pour les réseaux à translation d'adresses</p> <p>Comparaison des différentes approches : isanat/Teredo</p> <p>Choix d'un tunnel broker.</p>
Cas concret	<p>Exemple de migration d'une infrastructure complète.</p> <p>La sécurité : IPsec, les pare feux et les filtres. ip6tables, routeurs et répartiteurs de charge, haproxy.</p> <p>Organisation de la migration.</p> <p>Impacts de la mise en oeuvre de plusieurs dhcpv6 et de multiples agents RA.</p> <p>Problème de boucles de tunnels.</p>
Routage dynamique IPv6	<p>RIP, utilisation en IPv6. Inconvénients du protocole.</p> <p>OSPF v3 pour IPv6 : Présentation du routage des systèmes autonomes. Découverte des routeurs voisins, élection du routeur désigné, calculs des chemins.</p> <p>TP : mise en oeuvre d'OSPF sous Linux avec Quagga.</p>

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 oeuvre
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 oeuvre 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 sécurisées : 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	Chiffrement 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

Mise en oeuvre du protocole SNMP

Durée: 2 jours
985 €

27 au 28 février
10 au 11 avril

18 au 19 septembre
27 au 28 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 oeuvre 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
1920 €

10 au 13 mars
du 2 au 5 juin

1er au 4 septembre
3 au 6 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

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.

Architectures et protocoles

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

Qualité

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.

Mise en pratique

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).

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>

Performances et optimisation des réseaux

Durée: 2 jours
1045 €

6 au 7 février
6 au 7 mai

25 au 26 septembre
6 au 7 novembre

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 oeuvre 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 oeuvre des tc avec CBO puis HTB sur une interface.

Annuaire LDAP

Durée: 2 jours
950 €

27 au 28 février
22 au 23 mai

11 au 12 septembre
11 au 12 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 schéma
Mise en oeuvre	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 Idif. 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 oeuvre pour un serveur de messagerie.
Le marché	Présentation des principaux annuaires.

Zimbra : installation et configuration

Durée:3 jours

Prix et dates: nous consulter

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 oeuvre de l'interface web : Zimbra webclient

Démonstration de la synchronisation avec Zimbra Mobile

Webclient sur android

Filière Production et supervision

Sauvegardes
avec Bacula
2 j

Supervision
avec Nagios
3 j

Administration
Nagios
5 j

Supervision
avec Shinken
3 j

Reporting
avec Cacti
3 j

Gestion de parc avec
OCS et glpi
3 j

Puppet
Administration centralisée
3 j

Supervision nagios : utilisation

Durée: 3 jours
1440 €

10 au 12 février
7 au 9 avril
16 au 18 juin

25 au 27 août
13 au 15 octobre
8 au 10 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
2300 €

10 au 14 février
7 au 11 avril
16 au 20 juin

25 au 29 août
13 au 17 octobre
8 au 12 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, et mettre en oeuvre la supervision dans un environnement hétérogène.

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 programmé 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'ordonnement. 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 oeuvre 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 oeuvre de NDO</p>

Administration Nagios

Intégration Nagios

Liens avec Cacti, Centreon, PNP4Nagios

Supervision d'environnements hétérogènes :

Windows avec ns_client et check_WMI, Unix (AIX, Solaris),
Linux,

matériels réseaux (Cisco , Hp),

Monitoring d'applicatifs : services web, messagerie, serveurs Jee

Nagios et le cloud : supervision intégrée avec OpenStack.

Supervision avec shinken

Durée: 3 jours
1440 €

6 au 8 janvier
3 au 5 mars

15 au 17 juillet
29 septembre au 1er octobre
24 au 26 novembre

Public:

Exploitants et utilisateurs d'un système de supervision Shinken.

Objectifs:

Connaître les fonctionnalités de Shinken, maîtriser l'interface d'exploitation, savoir ajouter de nouveaux tests, savoir mettre en place une politique d'alerte.

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 Shinken	Historique, licence, fonctionnalités : Supervision, exploitation, surveillance des services réseaux, surveillance des ressources (charge CPU, espace disque), émission d'alertes, actions automatiques programmables, gestion de règles métier. L'architecture : Arbiter, Scheduler, Poller, Reactionner, Broker. Comparaison avec Nagios.
Installation et configuration	Sur les systèmes Linux, plusieurs méthodes possibles : par le script d'installation, en exécutant setup.py, par les RPMs Travaux pratiques : installation d'un serveur Shinken, démarrage des services. Configuration de l'interface WebUI.
Les plugins	Principe, mise en oeuvre de plugins simples. Compatibilité avec Nagios. Développement de plugins.
Performances	La haute disponibilité avec Shinken. Ajout de processus par un utilisateur. Lissage automatique de la charge par l'architecture de Shinken.
Intégration	Les outils complémentaires : Centreon, Nagvis, pnp4Nagios

Reporting avec Cacti

Durée: 3 jours
1440 €

17 au 19 mars
19 au 21 mai

8 au 10 septembre
17 au 19 novembre

Public:

Administrateurs systèmes et réseaux souhaitant mettre en oeuvre 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.

Gestion de Parc avec OCS et GLPI

Durée: 3 jours
1435 € HT

24 au 26 février

14 au 16 avril

10 au 12 juin

20 au 22 octobre

15 au 17 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	Le besoin : inventaire et suivi des configurations matérielles et logicielles Présentation OpenComputer and Software Inventory Next Generation Fonctionnalités, informations collectées
Architecture OCSNG	architecture client/serveur Les composants du serveur OCSNG : base de données, serveur de communication, agents, serveur de déploiement, console d'administration Intégration avec GLPI
Installation	systèmes supportés, installation depuis les packages création des utilisateurs Les agents : procédure d'installation et configuration du lancement au démarrage du système
Configuration	configuration par l'interface web : fonction IPDISCOVER, détection des doublons (adresses Mac) notion de TAG : quelques exemples d'utilisation Les différents paramètres de chaque machine gérée.

Gestion de Parc avec OCS et GLPI

Utilisation glpi	<p>La Console Centrale, l'accès aux différentes fonctionnalités.</p> <p>L'inventaire : requêtes préféfinies. 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.</p> <p>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.</p> <p>Mise en oeuvre pratique des plugins tracker, reports, racks, data injection, item uninstillation.</p>
Import OCS NG	<p>Objectif et principe de fonctionnement.</p> <p>Configuration du mode OCSNG.</p> <p>Options d'importation.</p> <p>Mode d'import OCS.</p> <p>Import OCS par le plugin massocsimport.</p>
Gestion du cycle de vie.	<p>Le besoin.</p> <p>Plugin d'injection de fichiers CSV (data_injection)</p> <p>Liaison d'une machine.</p> <p>Plugin de désinstallation d'une machine, de suppression d'une machine.</p> <p>Statuts des matériels.</p> <p>Gestion des machines en stock.</p> <p>Machines en réparation et réformées.</p>
Télédéploiement de paquets	Principe et architecture. Notion de priorité et action à exécuter
Exploitation OCS glpi	<p>Sauvegardes, journalisation.</p> <p>Lien avec un annuaire ldap.</p>
Fusion Inventory	<p>Solution alternative à OCS pour l'inventaire</p> <p>Principe de fonctionnement avec SNMP.</p> <p>Présentation du plugin FusionInventory et mise en oeuvre.</p> <p>Installation des agents.</p> <p>Lien avec SNMP.</p> <p>Réalisation d'un inventaire réseau.</p>

Sauvegardes avec Bacula

Durée: 2 jours

Prix et dates: nous consulter

Public:

Administrateurs , exploitants souhaitant utiliser Bacula pour les sauvegardes.

Objectifs:

Savoir mettre en oeuvre Bacula pour les sauvegardes des systèmes Unix/Linux.

Connaissances préalables nécessaires:

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

Programme:

Introduction	<p>Rappels sur les 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</p>
Présentation Bacula	<p>Fonctionnalités : sauvegardes/restaurations en réseau, planification, gestion de contraintes de production, sécurisation des sauvegardes, catalogue des sauvegardes, etc ... License GNU. Plate-formes supportées (serveurs et clients).</p>
Architecture	<p>Architecture client/serveur. Les services Bacula : les démons Director (planification, authentification, contrôles), Storage (accès aux médias), File (partie cliente); Console (interface d'administration); Catalogue (gestion des fichiers sauvegardés), Monitor (supervision des services Bacula).</p>
Installation et configuration	<p>Installation des packages Bacula et de la base de données (MySQL ou PostgreSQL ou SQLite) Configuration. Définitions (Jobs, Schedule, polls, volumes, labels) Etude de la configuration pour chaque service. Travaux pratiques :réalisation d'un jeu de sauvegardes et tests de restauration.</p>
Exploitation	<p>Fichiers de logs : logrotate, logwatch</p>

Puppet : administration centralisée

Durée: 3 jours
1410 € HT

17 au 19 février
26 au 28 mai

1^{er} au 3 septembre
12 au 14 novembre

Public:

Administrateurs, exploitants souhaitant utiliser Puppet pour la gestion centralisée des configurations logicielles.

Objectifs:

Comprendre le fonctionnement de Puppet, et savoir le mettre en oeuvre pour une administration centralisée.

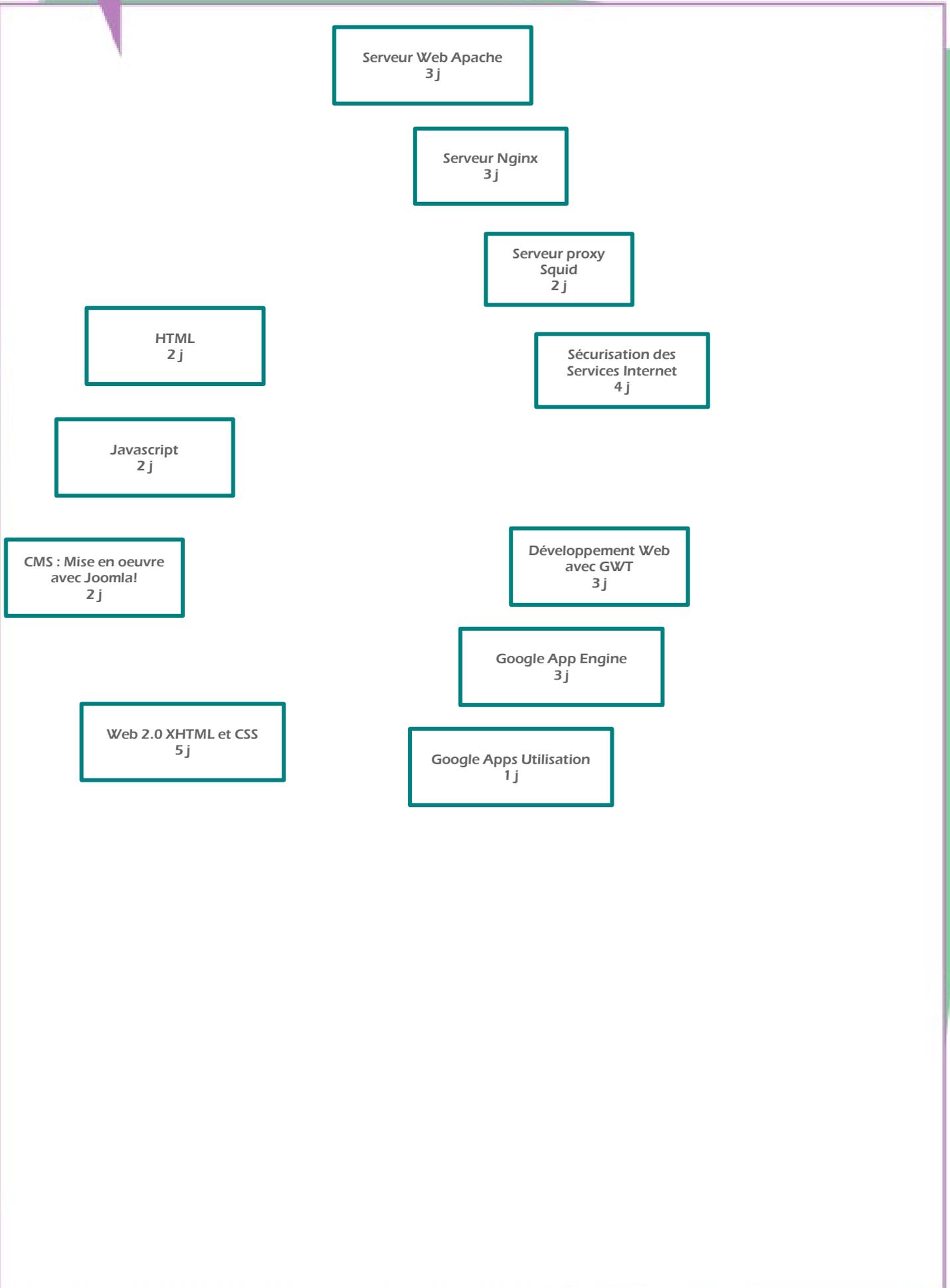
Connaissances préalables nécessaires:

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

Programme:

Introduction	<p>Présentation du besoin et des fonctionnalités de Puppet. Gestion des configurations logicielles. Collecte et centralisation des informations. Définition des configurations cibles, mises à jour automatiques ou manuelles. L'orchestration dans le cas d'un cloud d'entreprise. Les différentes versions, et présentation de Puppet Enterprise.</p>
Architecture	<p>Principe client-serveur, modules de configuration, les agents Puppet, la console, l'outil de gestion de cloud. Mode opératoire : définition des configurations, Vérification de l'état des clients. Simulation des changements proposés par Puppet. Application sur les systèmes cibles.</p>
Les outils	<p>Facter : pour le recensement des informations des clients. RalsH : un langage de script shell. Puppet Dashboard : interface de rapport d'activité des agents. Mcollective : pour exécuter des commandes en parallèle sur les serveurs cibles. La forge de modules et Puppet Module Tool pour utiliser des modèles de configurations.</p>
Installation et configuration	<p>Prérequis systèmes. Travaux pratiques : Installation d'une infrastructure Puppet : serveur, base de données, agents. Configuration des rôles : maître, agent, console... Gestion des certificats sur les clients.</p>
Les modules	<p>Présentation de "Puppet Forge" et recherche de modules. Le langage de description de configuration. Création de modules, mise en oeuvre, et dépôt sur la forge.</p>

Filières Internet et Web



Serveur WEB : apache

Durée: 3 jours
1325 €

17 au 19 février
14 au 16 avril
2 au 4 juin

15 au 17 septembre
3 au 5 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 oeuvre, 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. Les nouveautés de la version 2.4 : améliorations du noyau, nouveaux modules. Les techniques de migration vers la version 2.4. 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 oeuvre 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. Apports de la solution fastCGI.

Serveur WEB : apache

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.
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.
Le mode proxy	Architecture forward proxy et Reverse proxy. Travaux pratiques : installation et configuration. Gestion du cache avec le module mod_cache.

Administration serveur Nginx

Durée: 3 jours
1400 €

10 au 12 mars
2 au 4 juin

13 au 15 octobre
15 au 17 décembre

Public:

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

Objectifs:

Comprendre le fonctionnement du serveur Nginx, et savoir le mettre en oeuvre, 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, ... Plates-formes supportées. Architecture : principe de serveur asynchrone, modularité.
Mise en oeuvre	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 oeuvre 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 oeuvre 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

Prix et dates: nous consulter

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
1940 €

27 au 30 janvier
31 mars au 3 avril
23 au 26 juin

13 au 16 octobre
15 au 18 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

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 oeuvre

Les ports de niveau 5

Rappels sur la notion de ports.
Principe d'empilement des trames.
Les ports UDP.
Exemples de trames.

Outils réseaux

Le principe des traces, les informations disponibles dans les captures de trames.
Travaux pratiques :
Traçage des flux réseaux: wireshark, tcpdump.

DMZ et pare-feux

Définition, et architecture
Fonctionnement des pare-feux.

Firewall iptables

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 oeuvre d'une fonction d'activation d'ouverture avec les iptables.

SSH et GPG

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.
GPG : GNU Privacy Guard
Transmission de messages chiffrés et signés.

Sécurisation des services internet

VPN et tunnels	Objectifs. Fonctionnement. Mise en oeuvre : tunnels ssh, stunnel. Travaux pratiques avec openVPN. Principe d'IPsec. Implémentation avec FreeS/WAN.
Proxy	Introduction : la fonction d'un serveur proxy. Les multi-serveurs proxys. Mise en oeuvre 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.
Services reseaux	Sécurisation DNS : architecture redondante, DNS fermé, Hidden master, Stealth DNS. Mise en oeuvre de la sécurisation d'un DNS. Authentification. Les signatures TSIG. Les ACL.
Messagerie	Sécurisation. Notion de relais ouvert. Outils de test du serveur. Travaux pratiques : mise en oeuvre avec postfix. Apport de Cyrus.
Acces reseaux	Authentification par un serveur radius. Techniques de filtrage à base d'iptables. Mise en oeuvre d'un système d'authentification par utilisateur et non par adresse : NuFW.
Détection de failles	Outils : snort, openvas. Travaux pratiques : mise en oeuvre et réalisation d'un audit de sécurité de l'environnement de test.

HTML et CSS

Durée: 2 jours
1010 €

3 au 4 mars
12 au 13 mai

6 au 7 septembre
17 au 18 novembre

Public:

Développeurs de pages HTML.

Objectifs:

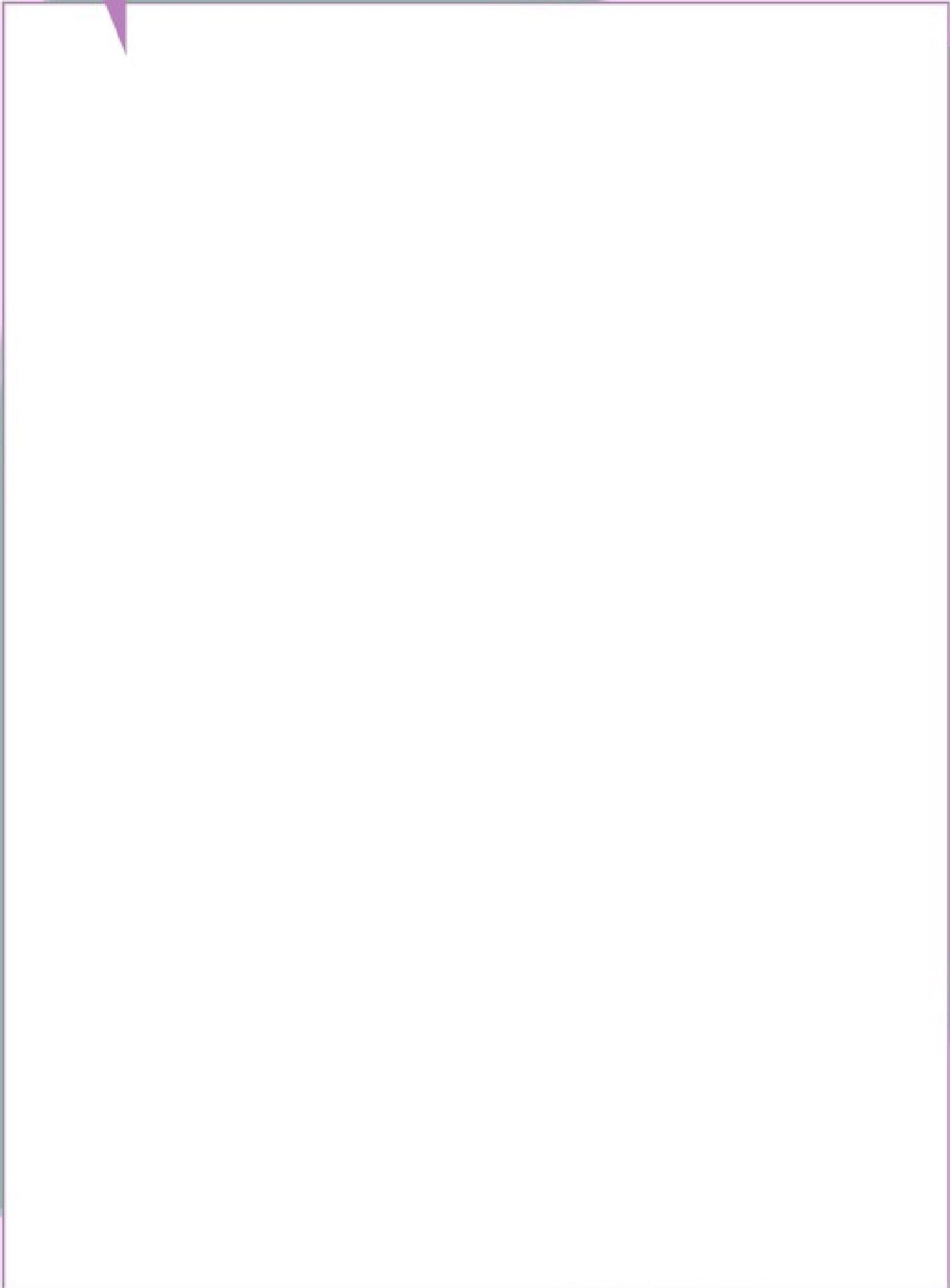
Etre capable de développer des documents HTML. Explorer l'ensemble des possibilités offertes par HTML.

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 élémentaires. 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
1440 €

5 au 7 mars
14 au 16 mai

8 au 10 octobre
19 au 21 novembre

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	Modes 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.

Web 2.0 - (X)HTML, CSS, Javascript

Durée: 5 jours
2300 €

27 au 31 janvier

24 au 28 mars

14 au 18 avril

23 au 27 juin

15 au 19 septembre

17 au 21 novembre

8 au 12 décembre

Public:

Analystes, développeurs d'applications.

Objectifs:

Concevoir et développer un site web utilisant les standards XHTML, CSS et Javascript.

Connaissances préalables nécessaires:

Connaissance des bases d'un langage de programmation.

Programme:

La démarche de développement d'un site web

Le choix de la solution technique : différences entre web statique et web dynamique, javascript, Flash, Ajax.
La conception du site : mise en page, navigation.
La charte graphique et les règles ergonomiques.

Le langage HTML et XHTML

Présentation générale d'HTML 4.0

De HTML à XHTML.

Les tags de présentation.

Les liens.

Les inclusions multimédias et les formats d'images GIF, JPEG et PNG.

Les tableaux.

Travaux pratiques : concevoir la mise en page d'un site web et en définir les règles ergonomiques.

Le découpage des pages HTML

Les balises FRAME et IFRAME.

Les feuilles de style (CSS)

Les concepts de base.

Les propriétés de style.

L'externalisation de la charte graphique.

Les propriétés de positionnement (CSS P)

Les concepts de base

Les spécificités liées aux navigateurs (Internet Explorer, Firefox, ...)

Travaux pratiques : créer une feuille de style pour externaliser la charte graphique du site. Développer les pages statiques du site.

Les formulaires

La balise FORM

Les balises INPUT, SELECT, TEXTAREA

Les contrôles de saisie : initiation à Javascript.

Présentation des techniques web dynamiques

ASP .Net, servlets JSP, PHP

Les applets Javascript Les composants ActiveX

Travaux pratiques : mettre en oeuvre un formulaire appelant un script serveur existant.

Le langage Javascript

L'objectif de Javascript

L'intégration aux pages HTML

Web 2.0 - (X)HTML, CSS, Javascript

Eléments syntaxiques	La structure générale d'un programme de scripts Les données, les opérateurs, les instructions de contrôle. Les fonctions et l'appel de scripts.
Les objets	Les différentes catégories d'objets du navigateur. Les objets Window, Frame, et Document. Les objets Link, Anchor, et Location. Les objets spécifiques à Javascript.
Le contrôle des formulaires	Les objets Button, Text, Select, Option , Checkbox et Radio. Travaux pratiques : développer une page HTML intégrant un formulaire contrôlé par Javascript.
Le dynamic HTML	Présentation du concept DHTML. Les principes et utilisations. Modification des styles CSS en Javascript. pratiques : développer une page HTML intégrant des boîtes de contenus repliables.
Du Web au Web 2.0 : Ajax	Présentation de Asynchronous Javascript And XML La personnalisation de l'interface : Drag and Drop Effets visuels Services web Le rafraîchissement partiel de la page : iFrame, XMLHttpRequest
HTML5 et CSS3	Présentation de la structure d'une page HTML5. Quelques propriétés CSS3. Impacts sur les développements actuels.

CMS : mise en oeuvre de Joomla

Durée: 3 jours

Prix et dates: nous consulter

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

Google Apps Utilisation

Durée: 1 jour
480 €

24 janvier
23 mai

3 octobre
5 décembre

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 oeuvre 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 oeuvre 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
1640 € HT

du 3 au 5 février
du 19 au 21 mai

du 8 au 10 septembre
du 3 au 5 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 oeuvre 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
1565 € HT

13 au 15 janvier
28 au 30 avril

15 au 17 juillet
30 septembre au 1er octobre
24 au 26 novembre

Public:

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

Objectifs:

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

Connaissances préalables nécessaires:

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

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 oeuvre : 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

Réseau social d'entreprise : techniques et outils

Durée: 2 jours
1040 €

13 au 14 mars
19 au 20 juin

2 au 3 octobre
11 au 12 décembre

Public:

Les chefs de projet, architectes, responsables méthodes et outils, intéressés par la mise en place d'un réseau social d'entreprise.

Objectifs:

Comprendre quelles sont les technologies utilisées dans le cadre des réseaux sociaux, savoir quelles sont les solutions disponibles pour créer un RSE.

Connaissances préalables nécessaires:

Notions sur les réseaux, bases de TCP/IP et des technologies internet.

Programme:

Présentation des RSE

Fonctionnalités :

courrier électronique, messagerie instantanée, zones de rencontre,

échange d'informations, de documents, galerie photos, calendrier partagé, gestion des rendez-vous,

flux d'activité, groupes, forums, blogs, sondages, enquêtes, ...

Positionnement par rapport aux outils de l'entreprise :

évolutions de l'extranet, enrichissement des outils de messagerie, des solutions de travail collaboratif...

Technologies

Les techniques mises en oeuvre dans le RSE :

Le web 2.0, cloud, SaaS...

APIs mises à disposition par les réseaux sociaux publics

Solutions

Panorama des outils disponibles :

twistranet, jaliOS, zimbra, liferay, drupal, bluewiki, statusnet, eXo Social, BuddyPress....

Principales caractéristiques de ces outils :

fonctionnalités, architecture, modèle économique (SaaS, ou solution interne),

Mise en oeuvre

Exemple de création d'un RSE en interne à l'entreprise avec les fonctionnalités standards :

gestion de groupes, d'agendas, de micro-blogging...

Intégration dans le système d'information

Liens avec les autres outils : messagerie, la gestion de documents...

Les risques

Confidentialité des données et informations de l'entreprise, l'excès d'informations, doublons dans les fonctionnalités entre le système d'informations et le RSE, manque d'adhésion...

Conseils pour bien aborder un projet de RSE.

Filières Virtualisation et Cloud

Virtualisation
Linux
2 j

Virtualisation avec
Xen
2 j

Virtualisation avec
kvm
2 j

Virtualisation avec
lxc
2 j

Administration
de machines virtuelles
2 j

Cloud :
technologies et enjeux
1 j

Architecture cloud
d'entreprise
3 j

Eucalyptus :
configuration cloud d'entreprise
2 j

CloudStack :
mise en oeuvre
2 j

Utilisation OpenStack
2 j

OpenStack : administration
3 j

Virtualisation Linux

Durée: 2 jours
1070 € HT

13 au 14 janvier
10 au 11 mars
19 au 20 mai

15 au 16 juillet
29 au 30 septembre
1er au 2 décembre

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 éditions.
Configuration des machines virtuelles en XML.

Ixc

Linux Containers
Objectifs du projet
Isolation et contrôle des ressources.
Mise en oeuvre.

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 Xen

Durée: 2 jours
1070 €

16 au 17 janvier
13 au 14 mars
22 au 23 mai

17 au 18 juillet
2 au 3 octobre
4 au 5 décembre

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

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.

Installation

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.

Domaines utilisateurs

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.

Supports de stockage

Déclaration des espaces de stockage accessibles au domU : périphériques blocs, partitions physiques, ou volumes logiques.

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 oeuvre 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 avec KVM

Durée: 2 jours
1070 €

20 au 21 janvier
7 au 8 avril

30 juin au 1er juillet
6 au 7 octobre
8 au 9 décembre

Public:

Administrateurs, chefs de projet et toute personne souhaitant mettre en oeuvre 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 oeuvre	<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 oeuvre : la commande migrate Paramètres (bande passante) Migration vers un fichier : sauvegarde puis restauration</p>

Virtualisation avec KVM

Administration

Les outils de gestion de machines virtuelles kvm :
UVMM, virsh, virt-manager.
Travaux pratiques avec libvirt
Présentation de proxmox et mise en oeuvre :
gestion de machines virtuelles, création de clusters proxmox.
Méthode de migration.

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 switchs VDE.
Configuration des switchs avec Unixterm.

Virtualisation avec lxc

Durée: 2 jours
1070 €

23 au 24 janvier
10 au 11 avril

3 au 4 juillet
9 au 10 octobre
11 au 12 décembre

Public:

Administrateurs, chefs de projet et toute personne souhaitant mettre en oeuvre la virtualisation avec lxc.

Objectifs:

Comprendre les principes des linux containers et savoir les mettre en oeuvre.

Connaissances préalables nécessaires:

Une bonne connaissance des systèmes Linux est nécessaire.

Programme:

Introduction

Les différentes techniques de virtualisation.

Définitions : conteneurs, machines virtuelles, hyperviseurs, virtualisation matérielle.

Présentation de lxc : Linux containers, historique, principe de fonctionnement. L'isolation de ressources, création d'un environnement utilisateur.

Positionnement par rapport aux autres solutions de virtualisation.

Cgroup

Fonctionnement de Control Group. Vérification de la configuration du noyau.

Activation des Cgroups.

Les outils LXC

Site de référence pour le téléchargement.

Installation de LXC par rpm, urpmi, yum ou apt-get install.

Présentation des différents outils pour vérifier la configuration du noyau, créer, détruire, gérer les conteneurs, et les tâches associées : lxc-checkconfig, lxc-console, lxc-create, lxc-start, lxc-stop etc...

Gestion des conteneurs

Configuration, création, démarrage. Utilisation des templates pour créer des conteneurs standards.

Choix des systèmes de fichiers.

Configuration du réseau

Les différentes méthodes : interface physique, pont/commutateur virtuel, vlan.

Mise en oeuvre. Configuration d'un point par brctl.

Administration de machines virtuelles

Durée: 2 jours
1070 € HT

13 au 14 mars
27 au 28 mai

17 au 18 juillet
11 au 12 septembre
4 au 5 décembre

Public:

Administrateurs responsables d'un parc de machines virtuelles.

Objectifs:

Connaître les techniques et outils de gestion d'un parc de machines virtuelles.

Connaissances préalables nécessaires:

Connaissance de l'administration système.

Programme:

Introduction

Le besoin : gestion du cycle de vie des machines virtuelles, sauvegardes/restaurations des configurations, des données, gestion des versions logicielles, déploiements, supervision, suivi des ressources (inventaire, suivi de consommation...), optimisation des performances, planification, réservation de ressources, orchestration...

Technologies

Présentation des techniques mises en oeuvre :
hyperviseurs, protocoles snmp pour les actions d'inventaire, nfs, pour les accès fichiers, interfaces comme libvirt...
Produits de gestion des environnements virtualisés :
Proxmox Virtual Environment, pour la gestion de machines virtuelles kvm ou openVZ ;
ZenWorks, pour l'orchestration ;
vSphere, en environnement VMware ;
ganeti, pour gérer des machines virtuelles Xen ou kvm, virt-manager, oovirt, interfaces d'administration basées sur libvirt...et des outils non spécifiques aux environnements virtualisés, mais nécessaires pour la gestion d'un nombre important de serveurs :
puppet, gestionnaire de configuration ;
nagios et shinken, pour la supervision ;
OCS/Glpi, pour l'inventaire et la télédistribution...

Administration de machines virtuelles

Mise en oeuvre

Travaux pratiques avec Proxmox Virtual Environment:
Architecture du produit, fonctionnement en grappes.
Installation de la plate-forme Proxmox, connexion depuis un navigateur,
Fonctionnalités disponibles :
gestion des images, suivi des capacités des serveurs, gestion des zones openVz, des grappes, etc ...
Stockage des images de VMs : iScsi/NFS
Installation d'un serveur d'images : iScsi ou NFS
Ajout de zones de stockage
Les migrations, différentes possibilités :
internes à Proxmox, entre deux noeuds ;
depuis ou vers une machine externe de type Xen, Vmware, qemu ;
migration de conteneurs openVZ.
Les clusters Proxmox, gestion des grappes.

Positionnement par rapport aux autres architectures

La proximité avec les outils de cloud.
Etude de quelques cas pratiques.

Cloud : technologies et enjeux

Durée: 1 jour
535 € HT

11 février
22 avril
16 juin

13 octobre
15 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 oeuvre.

Connaissances préalables nécessaires:

Connaissance générale des systèmes d'informations.

Programme:

Introduction

Définition : cloud computing,
Les types de cloud: 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, openNebula, cloudStack, 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 aspects sécurité : évaluation des risques, présentation des solutions.
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
1585 € HT

12 au 14 février
23 au 25 avril
17 au 19 juin

14 au 16 octobre
16 au 18 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 oeuvre. Savoir mener un projet de création d'un cloud privé.

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.

Architecture cloud d'entreprise

Cloud d'entreprise	<p>Cloud privé : infrastructure non partagée et dédiée à une entreprise ou une organisation.</p> <p>Les avantages d'une solution de cloud privé : sécurité, localisation des données.</p> <p>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.</p> <p>Exemples de produits de cloud privé : Eucalyptus, Xen Cloud Platform, VMWare VSphere, Enomaly, cloudstack, openstack, openNebula</p> <p>Fonctionnalités : déploiement sur des clusters, déploiement de composants sur différents serveurs, possibilité de maintenance "on-line".</p>
Mise en oeuvre	<p>Travaux pratiques avec OpenStack et Eucalyptus.</p> <p>Présentation.</p> <p>Architecture des produits.</p> <p>Travaux pratiques : installation depuis les packages RPM.</p> <p>Configuration de l'hyperviseur (xen ou kvm).</p> <p>Lancement d'une première machine virtuelle.</p>
configuration d'un cloud privé	<p>Mise en oeuvre du réseau entre machines virtuelles.</p> <p>Les différents modes de gestion du réseau par Eucalyptus.</p> <p>Outils d'administration d'eucalyptus : interface web d'administration, ligne de commande (gestion des noeuds, des utilisateurs...)</p> <p>Lien avec un cloud public comme Amazon.</p>
La sécurité	<p>Evaluation des risques, présentation des outils et solutions</p> <p>Aspects juridiques, protection des données, traçabilité, chiffrement, génération de clés...</p>

Eucalyptus : configuration cloud d'entreprise

Durée: 2 jours
1060 € HT

20 au 21 février
5 au 6 mai

5 au 6 septembre
27 au 28 novembre

Public:

Architecte, chef de projet, et toute personne souhaitant installer une infrastructure de cloud avec Eucalyptus.

Objectifs:

Comprendre le fonctionnement d'Eucalyptus, et savoir l'installer et le configurer.

Connaissances préalables nécessaires:

Connaissance générale des systèmes d'informations, systèmes et réseaux IP.

Programme:

Introduction

Définition : cloud computing,
Les types de clouds : SaaS, PaaS, IaaS. Clouds privés et publics.
Présentation d'Eucalyptus :
"Elastic Utility Computing Architecture for Linking Your Programs To Useful Systems".
Fonctionnalités :
réseau virtuel, partage de données, stockage distant (services EBS),
mise à disposition de services (SNS, SQS...), applications,
pilotage distant de systèmes locaux,
mise à disposition de ressources à la demande.
Caractéristiques techniques : hyperviseurs supportés, compatibilité Amazon.
Notion de cloud hybride.
Différences entre la version opensource et Enterprise Edition.

Architecture d'Eucalyptus

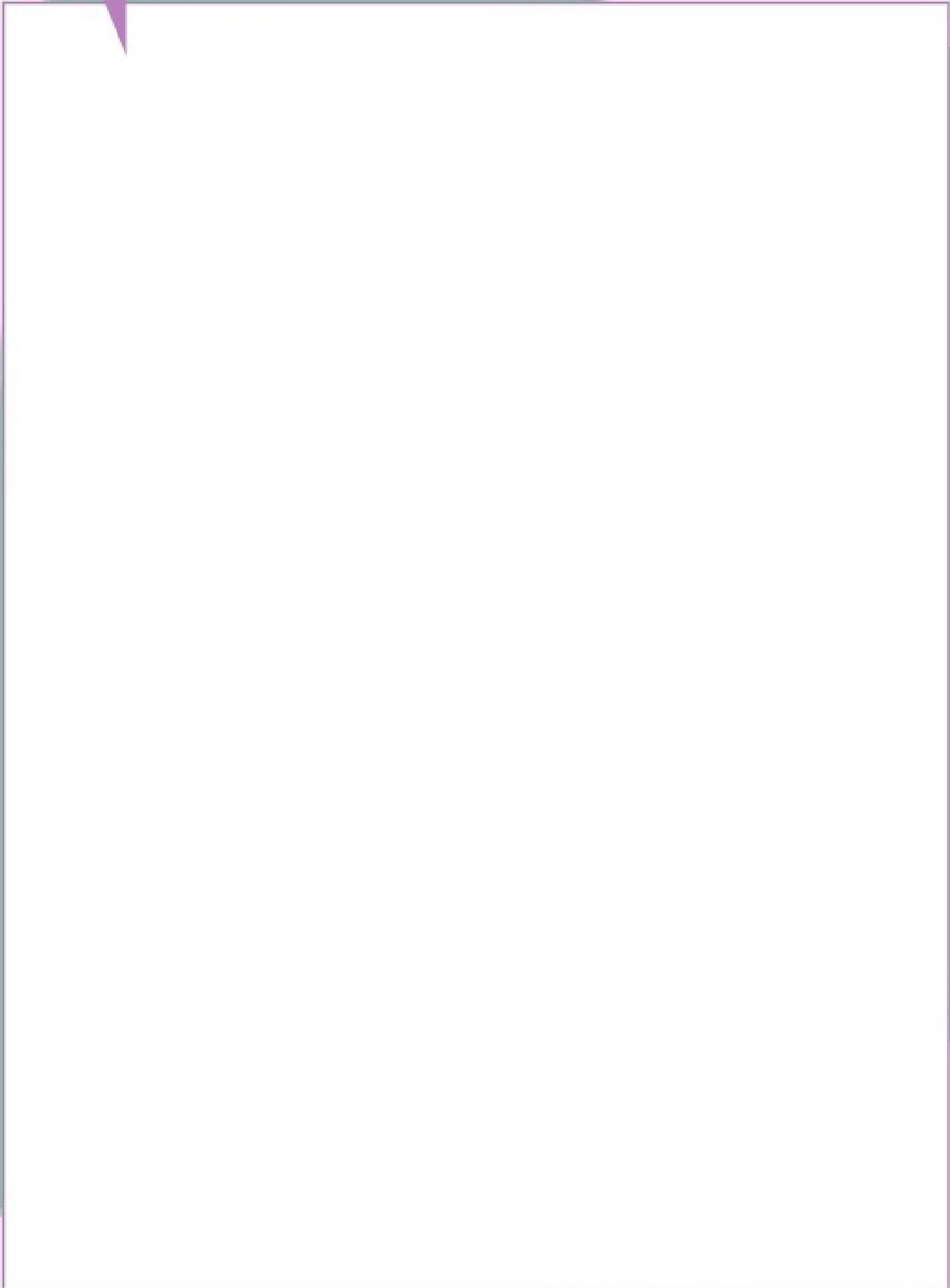
Organisation en web services : Cloud Controller (CLC): le contrôleur de cloud
Walrus, gestionnaire de persistance ; Cluster Controller (CC), contrôleur de grappe ;
Storage Controller (SC), contrôleur de stockage ; Node Controller (NC), contrôleur de noeud, et VMware Broker.

Installation et configuration.

Installation des différents composants :
le noeud frontal hébergeant le CLC et Walrus, les noeuds frontaux de grappes, et les noeuds pilotant les machines virtuelles.
Configuration de l'infrastructure :
niveau de journalisation, interfaces réseau, modes réseau (gestion des adresses Mac)...
Définition des machines virtuelles.
Console d'administration : interface de gestion des utilisateurs et des groupes, gestion des comptes pour une facturation en fonction des ressources utilisées.

Compatibilité avec Amazon

Notion de cloud hybride.
API d'interfaçage avec Amazon.



CloudStack : mise en oeuvre

Durée: 2 jours
1060 € HT

27 au 28 février
28 au 29 avril

25 au 26 septembre
13 au 14 novembre

Public:

Architecte, chef de projet, et toute personne souhaitant installer une infrastructure de cloud avec CloudStack

Objectifs:

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

Connaissances préalables nécessaires:

Connaissance générale des systèmes d'informations, systèmes et réseaux IP.

Programme:

Introduction	Définition : cloud computing, Les types de clouds : SaaS, PaaS, IaaS. Clouds privés et publics. Présentation de CloudStack : historique, license. Fonctionnalités : fourniture d'une infrastructure IaaS, outils d'orchestration de cloud, fourniture de stockage élastique, gestion d'images virtuelles...
Caractéristiques techniques	Hyperviseurs supportés. VMWare, Oracle VM, KVM, XenServer, Xen Cloud Platform. Interfaces d'administration : interface web, ligne de commande, API RESTful. Architecture : Management Server pour la gestion des machines virtuelles, des utilisateurs, de l'adressage IP, des ressources disques... Terminologie CloudStack : zone, pod, cluster, hôte, stockage primaire, stockage secondaire...
Installation et configuration	Prérequis matériel et logiciel pour le CloudStack Management Server et pour l'hyperviseur. Travaux pratiques : installation du Management Server, de la base MySQL et du partage NFS. Choix et installation d'un hyperviseur. Configuration d'une infrastructure cloud par la console web : création de zones, pods, clusters, hôtes, et stockage primaire et secondaire. Initialisation et création des premières instances du cloud.
Administration du cloud	Gestion des utilisateurs. Configuration du réseau. Exploitation des machines virtuelles : création, migration, suppression, Utilisation de projets pour regrouper des utilisateurs partageant les mêmes ressources. Mise en oeuvre des templates.

OpenStack : configuration et administration

Durée: 3 jours
1625 € HT

19 au 21 février
14 au 16 mai

17 au 19 septembre
19 au 21 novembre

Public:

Architecte, chef de projet, et toute personne souhaitant installer une infrastructure de cloud avec OpenStack

Objectifs:

Comprendre le fonctionnement d'OpenStack, et savoir l'installer et le configurer.

Connaissances préalables nécessaires:

Connaissance générale des systèmes d'informations, systèmes et réseaux IP.

Programme:

Introduction	Présentation d'OpenStack : historique, acteurs, licence Fonctionnalités : outils d'orchestration de cloud, stockage élastique, gestion d'images virtuelles,...
Caractéristiques techniques	Hyperviseurs supportés. Architecture : les projets OpenStack Nova, Cinder, Glance, Swift, Quantum, Keystone, Horizon L'accès par des clients webservice Dashboard, l'interface de gestion, pilotage, suivi.
Installation et configuration	Prérequis matériel et logiciel. Les composants à installer : authentification avec Keystone, serveur d'images Glance, stockage d'objets avec Swift, serveur de calcul Nova Définition des tenants. Préparation de l'installation. Travaux pratiques : installation à partir de scripts, configuration de la base de données, du service de messages, du service keystone : gestion des utilisateurs, groupes, accès; configuration du réseau, création et lancement des instances.
Extension	Ajout d'un noeud de calcul. Configuration. Visualisation de la capacité supplémentaire. Gestion. Migration d'un noeud de calcul.
Gestion des volumes	Présentation de Cinder. Architecture, locale, NAS, SAN. Mise en oeuvre : démarrage du service, installation de volumes iScsi. Manipulation de disques virtuels : création, attachement, formatage, suppression.

OpenStack : configuration et administration

Gestion du réseau

Définitions : adresses mobiles, adresses fixes.

Les modes réseaux : Flat, FlatDHCP, VLAN.

Quantum : fonctionnalités, architecture. Mise en oeuvre sur un noeud autonome.

Création d'un réseau privé virtuel. Ajout de routeurs virtuels et liaisons avec les instances.

Création de stocks d'adresses IP mobiles. Affectation.

Définition d'une architecture globale.

OpenStack : utilisation

Durée: 2 jours
1060 € HT

17 au 18 février
12 au 13 mai

15 au 16 septembre
17 au 18 novembre

Public:

Exploitant et tout utilisateur d'OpenStack.

Objectifs:

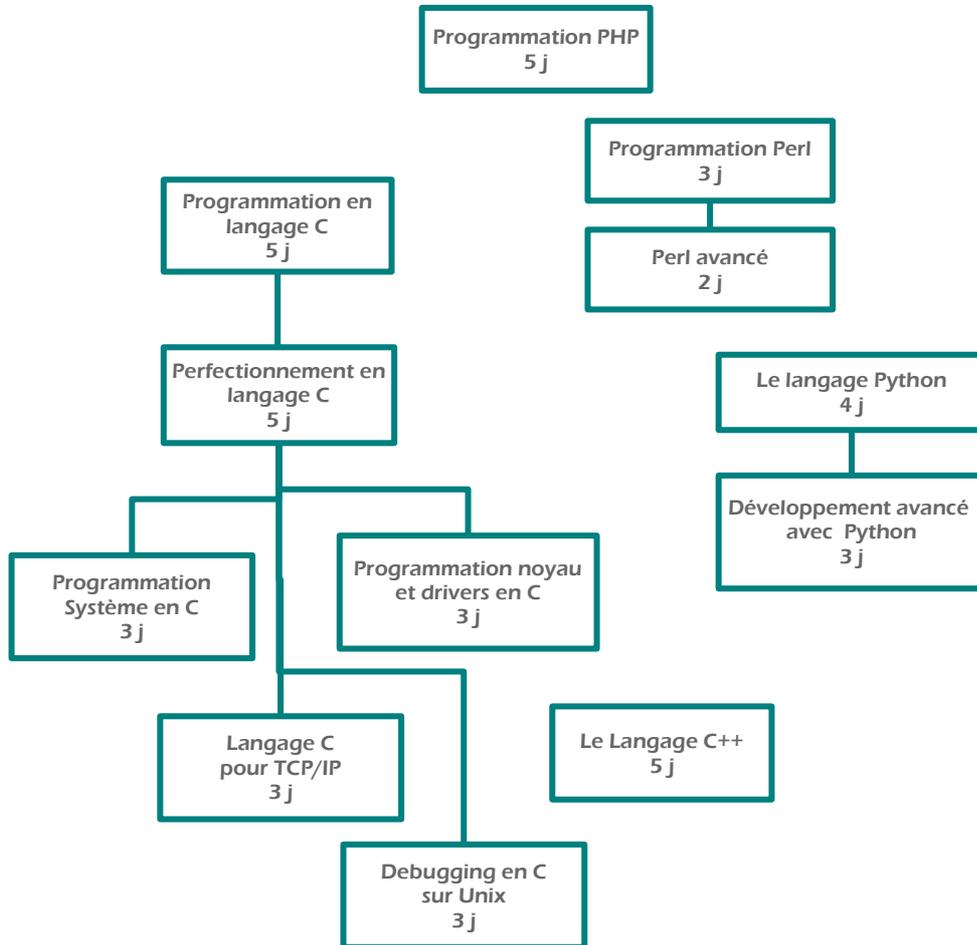
Savoir utiliser OpenStack pour gérer des machines virtuelles, des ressources dans un cloud.

Connaissances préalables nécessaires:

Connaissance générale des systèmes d'informations, systèmes et réseaux IP.

Programme:

Introduction	Présentation d'OpenStack : historique, acteurs, licence Fonctionnalités : fourniture d'une infrastructure IaaS, outils d'orchestration de cloud, stockage élastique, gestion d'images virtuelles,...
Caractéristiques techniques	Hyperviseurs supportés Architecture : les projets OpenStack Nova, Cinder, Glance, Swift, Quantum, Keystone, Horizon L'accès par des clients webservices Dashboard, l'interface de gestion, pilotage, suivi.
Utilisation	Présentation de la console web. Notions de projets, utilisateurs, clés Gestion des images. Gestion des machines virtuelles. Groupes de sécurité. Utilisation de la console VNC intégrée.
Exploitation	Présentation des outils. Commandes de base Mise en oeuvre de keystone, glance, nova Statistiques/consommations. Reprises en cas de panne d'un noeud.
Interfaces EC2	Présentation des EC2 Tools. Mise en oeuvre. Centralisation de l'exploitation.
Intégration	Interfaçage avec Amazon EC2. Tests avec Amazon, Rackspace, Trystack Supervision avec Nagios



Programmation en langage C

Durée: 5 jours
2055 € HT

10 au 14 février
7 au 11 avril
23 au 27 juin

1er au 5 septembre
24 au 28 novembre

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, strlwr, 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 oeuvre. 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
2100 € HT

3 au 7 mars
12 au 16 mai

7 au 11 juillet
15 au 19 septembre
8 au 12 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 oeuvre 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 oeuvre. 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 oeuvre 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 oeuvre de fprintf, fscanf, fgets, fputs.

Programmation système en C sur Unix/Linux

Durée: 3 jours
1500 € HT

24 au 26 mars
26 au 28 mai

29 septembre au 1er octobre
17 au 19 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 oeuvre 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 oeuvre de la primitive `msgget()`,

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

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

Segments partagés :

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

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

Sockets BSD :

mise en oeuvre des prises réseaux pour la communication interprocessus.

Exemple avec des liens locaux. Extension aux liens distants.

Communications inter-machines.

Les processus et la parallélisation

Création de processus.

Définition et mise en oeuvre 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 oeuvre des mutex.

Coopération de traitements entre threads.

Mise en oeuvre des conditions variables. Gestion des signaux dans un thread.

Ordonnancement de threads.

Programmation noyau et drivers en C sur Linux

Durée: 3 jours
1550 € HT

6 au 8 janvier
1er au 4 avril

29 septembre au 1er octobre
26 au 28 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	<p>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.</p>
Compilation noyau	<p>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.</p>
Programmation de modules	<p>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).</p>

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

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

Prix et dates: nous consulter

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 oeuvre	<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

Développement d'applications Web avec PHP

Durée: 5 jours
2250 €

17 au 21 février
31 mars au 4 avril
16 au 20 juin

22 au 26 septembre
17 au 21 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.

Développement d'applications Web avec 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
1235 €

3 au 5 février
7 au 9 avril
30 juin au 2 juillet

6 au 8 octobre
15 au 17 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) Plates-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
Fichiers	Entrées-sorties standards Manipulation de fichiers Les redirections Opérateurs de test de fichiers

Programmation Perl

Exécution

différentes méthodes d'exécution
compilation , exécution

Perl avancé

Durée: 2 jours
975 €

du 6 au 7 février
du 10 au 11 avril

du 3 au 4 juillet
du 9 au 10 octobre
du 18 au 19 décembre

Public:

Tout développeur souhaitant améliorer sa connaissance de Perl et acquérir les mécanismes avancés de programmation.

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 DBI	Présentation et installation. Exemples d'utilisation.
Programmation web avec CGI	Le module CGI. Méthodes fournies par CGI.pm Import du module complet ou de groupes de méthodes.

Le langage Python

Durée: 4 jours
1990 €

17 au 20 mars
19 au 22 mai

8 au 11 septembre
17 au 20 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

Les fichiers

Méthodes d'accès aux fichiers :
ouverture (accès en mode lecture, écriture, ajout, ...),
fermeture, le mot clé with;
lecture dans un fichier

Le langage Python

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
1515 €

26 au 28 mars
10 au 12 juin

24 au 26 septembre
24 au 26 novembre

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 : PyQt, 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 Gadgetfly, interfaçage MySQL avec MySQLdb, accès à PostgreSQL avec les modules PyGreSQL ou 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 oeuvre 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.
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, ...

XML Développement

Durée: 3 jours
1550 €

5 au 7 février
2 au 4 avril
26 au 28 mai

15 au 17 juillet
29 septembre au 1er octobre
12 au 14 novembre

Public:

Développeurs, concepteurs, chefs de projet.

Objectifs:

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

Connaissances préalables nécessaires:

Connaissances générales des systèmes d'information.

Programme:

Introduction au standard
XML Schéma Objectifs et
définition

Les limites de HTML
Les apports de XML.
Les champs d'application.
Les standards et outils associés (Open Document, RSS, SVG, ...)
Les langages du Web : XHTML, WML, XFORM, XUL ...

Les documents XML

La structure et la définition du document.
Les balises, les éléments et les attributs.
Commentaires, Texte et CDATA.
Document bien formé versus document valide.
Le Document Type Definition (DTD).
Les espaces de noms (namespaces)
Travaux pratiques : définir des DTD internes et externes.
Mettre en oeuvre la modularisation des DTD.

XML Schema

XML Schema : rôle et intérêt par rapport aux DTD.
Les types de base : integer, string, date, decimal, ...
La syntaxe des schémas : les listes, les unions, les séquences, les choix, ...
Les dérivations : par restriction et extension
Travaux pratiques : concevoir, à partir de spécifications, des XML Schéma utilisant les notions de base et avancées (dérivation)

XPath et la navigation

XPath : le langage de navigation.
Les axes nodaux.
Les types d'expression de sélection.
Les expressions : booléennes, numériques, chaînes de caractères.

Les liens XML (XLL)

XML Linking Language : rôle et intérêt
Xlink : les liens simples et étendus.
Xpointer : l'adressage de documents.

XML Développement

Les feuilles de style

XML et les feuilles de styles.
L'intégration des CSS.
XSL : rôle et positionnement.
XSLT versus XSL-FO
XSLT : le langage de tranformation.
Les documents réguliers et non réguliers.
Les règles de transformation et es expressions de sélection.
Les traitements sur les noeuds : génération, copie, dénombrement et tri.
Les instructions : test, itération, fusion.
Travaux pratiques : mettre en oeuvre les différents concepts XSLT par l'écriture de feuilles XSLT générant du code HTML à partir d'une source XML.

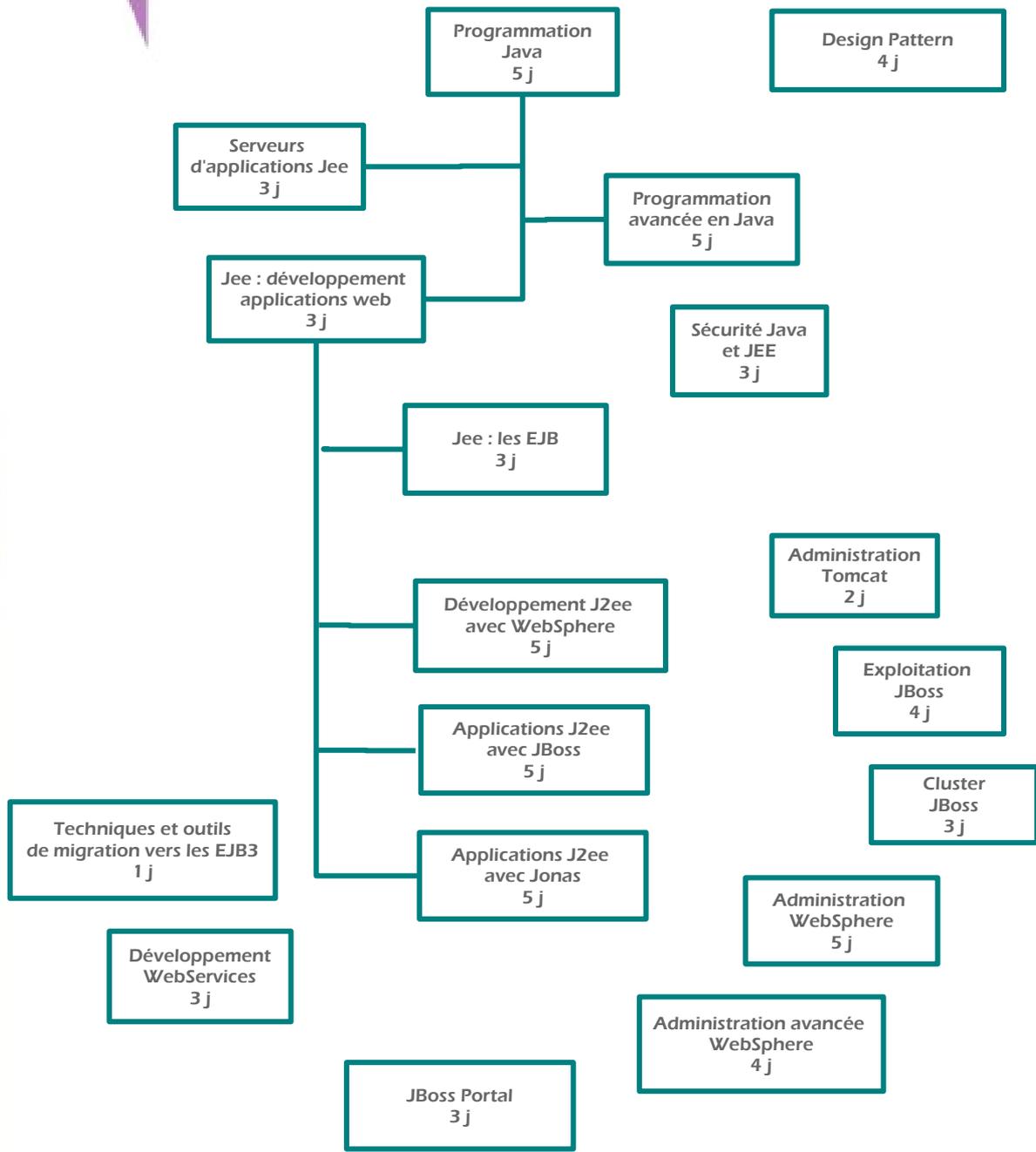
Les outils XML

Les outils d'édition : XMLSpy, oXygen, CookTop...
Les parseurs WSLT : MSXML, Saxon, Xalan, ...

Les modèles d'architecture

Le traitement sur le navigateur (XSLT, DOM...)
Le traitement sur le serveur web (XSLT, DOM, SAX...)
Les webs services (SOAP, WSDL)
Le traitement sur le serveur de données (Xquery, XUpdate, ...)

Filières Langages Java et JEE



Concepts Objets et programmation Java

Durée: 5 jours
2250 €

3 au 7 février
24 au 28 mars
2 au 6 juin

15 au 19 septembre
20 au 24 octobre
1er au 5 décembre

Public:

Développeurs, ingénieurs logiciels et architectes d'applications.

Objectifs:

Apprendre le langage Java et assimiler les concepts objet. Utiliser les outils du JDK et les principales API de la Standard Edition 6. Ce cours s'appuie sur le développement d'une application complète mettant en oeuvre une architecture à 3 niveaux (présentation, métier et persistance).

Connaissances préalables nécessaires:

Avoir une pratique de la programmation dans des langages structurés (C, Pascal, Basic, Fortran, Cobol, ...)

Programme:

Les concepts objet	Programmation objet, les réutilisables. Principe de l'encapsulation. Attributs et méthodes. Accesseurs. Différence entre objet et classe. Instanciation. Conventions de nommage.
Introduction à Java	Philosophie de conception sous-jacente à Java. Les différentes éditions. Les API de la SE 6. Les fichiers sources, le byte-code et la JVM. Première application.
Syntaxe java	Les règles d'écritures. Types primitifs. Types abstraits. Déclaration des variables. Opérateurs. Structures de contrôle. Tableaux.
Les packages	Rôle des packages. Définir ses propres packages.
Les classes	Déclaration de classes, d'attributs et de méthodes. Constructeurs. Instanciation. Destruction des objets : le garbage collector. Accès aux attributs et méthodes. Les références : this et null. Surcharge des noms de méthodes. Membres et méthodes de classe : static. Les classes composées d'objets. Contrôle d'accès aux membres.
Les énumérés	Définition. Exemples.
Les interfaces	Définition et déclaration. Utilisation des interfaces.
L'héritage	Mécanisme d'héritage. Recherche de méthodes pour une classe dérivée. Héritage et instanciation. Conversions standards dans l'héritage. Le polymorphisme. Classes et méthodes abstraites.
Les classes internes	Définition. Caractéristiques principales. Déclaration. Exemples.
Les exceptions	Définition. Graphes d'héritage. Gestion des exceptions.
Les structures de données	La classe Vector. La classe Stack. L'interface Enumeration. Structures de données ordonnées. Les collections.

Concepts Objets et programmation Java

Les génériques

Définition. Exemples.

Auto Boxing et Auto
UnBoxing

Objectif. Exemples.

Les annotations

Définition. Annotations standards. Exemples.

Présentation des principaux
packages

Java.lang et java.util. Internationalisation (i18n). Présentation
des autres packages.
Les entrées/sorties. Mode octet ou mode caractère. Canaux et
filtres. Formatage. Sérialisation.

Le multi-threading.

Thread et runnable. Priorité et état. Gestion de la concurrence
et synchronisation. L'API de concurrence.

Présentation de JDBC

Drivers JDBC. Les classes et interfaces en jeu. Les transactions.
Le support de SQL.

Programmation avancée en Java

Durée: 5 jours
2250 €

31 mars au 2 avril

22 au 24 septembre
8 au 10 décembre

Public:

Développeurs Java, ingénieurs logiciels et architectes d'applications.

Objectifs:

Approfondir la connaissance de Java notamment dans les domaines de la programmation multi-tâches, des tests et du logging.

Connaissances préalables nécessaires:

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

Programme:

Le multi-threading	Fonctionnement. Ordonnancement et priorité. Exclusion mutuelle. Synchronisation. Thread démon. Communication par flux "pipe".
L'API de concurrence.	Les exécuteurs de tâches. Les queues. Les maps atomiques. La représentation du temps et de ses unités. Les synchroniseurs. Les traitements asynchrones anticipés. Les variables atomiques. Les verrous "haute performance".
Les annotations.	Objectif. l'API Reflection. Annotations standards. Les méta-annotations. Fabriquer ses annotations. Annotation Processing Tool (APT)
Les nouvelles I/O	La gestion des flux standards : l'API Scanner.
La gestion et la supervision de la JVM	L'API de management JMX
Les tests	Objectif. Le framework JUnit.
Traçabilité des applications	Objectif. L'API Java Logging.
Java Management eXtension	Supervision avec JMX Principe des MBeans, et exemples de MBeans standards fournis à partir du jdk 1.5
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.
Sécurité	Introduction à la sécurité Java. Protection du système vis à vis des applications tierces : SecurityManager, ClassLoader. JCE. Chiffrement, chiffrement asymétrique, hachage

Programmation avancée en Java

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
1490 €

3 au 5 mars
5 au 7 mai

25 au 28 août
3 au 5 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 oeuvre, 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 oeuvre du SecurityClassLoader
Protection de l'exécution	Exécution protégée : SecurityManager, ClassLoader. Surcharge des méthodes d'accès s: 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 SHA1withDSA, MD5withRSA. 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 fabrication 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 oeuvre

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 oeuvre des certificats avec JEE.

JEE : Développement d'applications web

Durée: 3 jours
1440 €

10 au 12 février

7 au 9 avril

23 au 25 juin

29 septembre au 1er octobre

15 au 17 décembre

Public:

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

Objectifs:

Mettre en place une application web dynamique à l'aide de servlets, JSP, Taglibs et des JvaBeans en respectant le modèle MVC. Assurer la persistance en utilisant JDBC.

Connaissances préalables nécessaires:

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

Programme:

L'API JDBC

Objectif de JDBC. Les types de drivers. Les architectures applicatives.

Les classes et interfaces en jeu. Connexion.

La gestion des transactions et l'isolation transactionnelle. Interrogation et mise à jour. Appel d'une procédure stockée. Les types de données. Les pools de connexion. Les Rowset. La libération des ressources. La gestion des exceptions.

Présentation de JPA (Java Persistence API).

Architecture en couches

Présentation, Métier et Persistance. Couplage fort, couplage faible. Les Design Patterns nécessaires : Singleton, Factory, Façade, Iterator. Notion de composant. Découpage du composant en 3 couches (service, donnée, persistance). Mapping opérationnel pour la persistance. L'implémentation de la persistance avec JDBC.

La couche Présentation

Servlet, JSP et Taglib. Design Pattern MVC

Servlet (le contrôleur)

Objectif. Le protocole HTTP. L'API Servlet. Cycle de vie d'une servlet.

Gestion de contexte. Gestion de la requête client. Gestion des cookies.

Redirection côté client et côté serveur. Configuration et déploiement.

Java Server Page (la vue)

Objectif. Cycle de vie d'une page JSP. JSP dans le MVC. Les différents tags.

JSP et l'intégration des JavaBeans.

Les bibliothèques de Tags et JNDI

Objectifs. Utilisation et conception. La JSTL.

Objectif de JNDI. Enregistrement (Binding. Lookup)

JEE : les EJB

Durée: 3 jours
1550 €

20 au 22 janvier

17 au 19 mars

5 au 7 mai

16 au 18 juin

8 au 10 septembre

20 au 22 octobre

1er au 3 décembre

Public:

Développeurs Java, concepteurs, chefs de projet.

Objectifs:

Mettre en oeuvre 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
1600 €

10 au 12 février

7 au 9 avril

10 au 12 juin

6 au 8 octobre

1er au 3 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. Les EJB3 et annotations Java pour créer des webservices.
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. Mise en oeuvre de jUDDI Publication d'un Webservice.
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.
WebServices et Axis	Présentation Axis. Principe de fonctionnement. Mise en oeuvre. Maintien de session avec la méthode setMaintainSession() Gestion des attachements. Sérialisation personnalisée avec typeMapping.
Axis 2	Présentation, historique Fonctionnalités. Travaux pratiques : installation, lancement du serveur, tests.

Développement Web Services

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.
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 oeuvre de webservices RESTful avec Jersey.

EJB3

Apports des EJB3.
Génération de webservices à l'aide des annotations.

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 oeuvre 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
970 €

6 au 7 mars
15 au 16 mai

28 au 29 août
6 au 7 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 oeuvre 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
2420 €

24 au 28 mars
2 au 6 juin

1er au 5 septembre
15 au 19 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 7 et 8. 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. Les nouveautés de la version WebSphere Application Server V8.
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 : <code>ejbCreate()</code> , <code>ejbActivate()</code> , <code>setSessionContext()</code> , <code>ejbStore()</code> ,... Fichiers de description en XML. Les fichiers spécifiques de WebSphere: <code>ibm-application-bnd/ext/ext-pme.xmi</code> , <code>ibm-ejb-jar- *.xmi</code> , <code>ibm-web-*.xmi</code> .
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	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.
Intégration	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.
Introduction aux EJB3	Apport du jdk1.5. Annotations et génériques. Principes de mise en oeuvre 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.
EJB Session en version 3	Reprise de l'EJB Caddie. Migration en version 3. Dépendance d'injection. Intercepteurs de méthodes métier.
EJB Entité en version 3	Les fichiers de persistance. Méthodes d'accès : FIELD/PROPERTY. Mapping par défaut. Relations en EJB3. Héritage. Synchronisation avec la base de données, détachement/attachement.

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
2365 €

10 au 14 mars
16 au 20 juin

8 au 12 septembre
17 au 21 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 oeuvre 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	<p>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.</p>
JBoss présentation	<p>Historique, présentation des différentes versions et de leurs caractéristiques, de JBOss AS V 5, 6, 7 à WildFly. L'architecture Jboss : le noyau, la couche services, la couche présentation, la couche application, le principe JMX et les Mbeans. Installation, configuration. Premiers pas dans la console JBoss.</p>
Développement	<p>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.</p>
Servlets, pages JSP	<p>Cycle de vie des servlets, servlets thread-safe Définition des JSP. Liaison JSP/JavaBean.</p>
JNDI	<p>Inroduction, présentation de l'API. JNDI pour JEE.</p>

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 oeuvre 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 oeuvre 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 oeuvre 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 oeuvre 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 oeuvre 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 oeuvre (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 oeuvre	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
2420 €

27 au 31 janvier
31 mars au 4 avril

30 juin au 4 juillet
22 au 26 septembre
17 au 21 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 7 et 8.

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. Les nouveautés de la version WebSphere Application Server V8.
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, noeuds, 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 oeuvre 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 ldap. 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
960 €

20 au 21 février
17 au 18 avril
5 au 6 juin

18 au 19 septembre
6 au 7 novembre

Public:

Toute personne souhaitant configurer et administrer une application avec Tomcat.

Objectifs:

Comprendre le fonctionnement de Tomcat, et savoir le mettre en oeuvre, 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 postgresQL. 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 oeuvre 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
1995 € HT

20 au 23 janvier
24 au 27 mars
19 au 22 mai

29 septembre au 2 octobre
24 au 27 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>JBoss présentation Historique, présentation des différentes versions et de leurs caractéristiques, de JBoss AS V 5, 6, 7 à WildFly. 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. Présentation du noyau JBoss. Présentation du micro-conteneur. Installation, configuration, arborescence des fichiers. Les services disponibles. Adaptation de la configuration : ajout/suppression de services Visualisation dans les consoles de JBoss.</p>
Applicatif	<p>Terminologie: application web, container, sources de données, pilote JDBC, hôtes virtuels. Configuration des services essentiels : JBossWeb, JNDI, JBossMQ, ... Configuration du scanner de déploiement d'applications. Applications pliées, dépliées.</p>
Lien avec Tomcat	<p>Installation de Tomcat et modification des paramètres de base Activation du service Tomcat dans la configuration Jboss. Le fichier server.xml. Administration du service web. Mise en place d'hôtes virtuels.</p>
Gestion des ressources	<p>Fournisseurs d'URL, fournisseurs JMS, sessionMail. Mécanismes de connexion aux bases de données : JDBC, pool de connexions. Modification de la base de données cible. Exemple avec postgresql. Connexion interapplicatives avec JCA.</p>
Sécurité avec JAAS et SecurityManager	<p>Intégration de la sécurité dans le serveur d'applications. Mise en place d'une politique de sécurité lors du déploiement de l'application. Sécurité: JAAS, rôles, groupes de permissions.</p>

Administration JBoss

Automatisation	<p>Exploitation par scripts avec twiddle, JBoss-cli. Parcours de ressources.</p> <p>Ecriture d'un script de visualisation de statistiques sur un cluster: taux d'utilisation mémoire, nombres de requêtes, connexions JDBC.</p> <p>Scripts d'automatisation.</p>
Intégration	<p>Clustering JBoss : définitions, mise en oeuvre.</p> <p>Configuration de JBoss en multi-instances.</p> <p>Accès depuis un serveur apache, mise en place d'une liaison ajp à l'aide de mod_jk, mod_cluster.</p> <p>Réplication sous JBoss.</p> <p>Sessions Http distribuées. Réplication d'EJB.</p> <p>Ecriture d'un service JBoss permettant de simuler une application batch.</p>
Journalisation	<p>Mise en oeuvre de la journalisation.</p> <p>Analyse des logs</p> <p>Fichiers de logs, résolution d'incidents.</p> <p>Configuration de la journalisation. Modification des niveaux de journalisation.</p> <p>Différents types ventilations. Centralisation des logs vers une machine syslog externe.</p>
Suivi	<p>Gestion des performances. Suivi du garbage collector.</p> <p>Installation du MBean Jboss Profiler.</p> <p>Suivi du ramasse miettes (garbage collector) avec jconsole.</p> <p>Paramètres de la JVM pour gérer au mieux l'espace mémoire.</p> <p>Travaux pratiques: mise sous stress avec jmeter, suivi de la charge et amélioration de la configuration. Optimisation et allègement des configurations JBoss.</p> <p>Utilisation du MBean SNMP sur version < 7.</p> <p>Envoi de trap SNMP à une console de supervision.</p> <p>Lien vers la supervision avec Nagios via SNMP.</p> <p>Automatisation d'alertes en cas d'incident.</p>

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 7 et 8.

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 noeuds Installation du gestion de noeuds. Installation d'un noeud piloté. Transformation d'un serveur autonome en noeud piloté. Fonctionnement des agents locaux. Gestion des serveurs.
Cluster	Gestion de cluster : ajout de noeud, 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 noeuds Utilisation de Jython. Positionnement par rapport à JACL. 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. Chiffrage 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. Chiffrage. 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 oeuvre d'un système de surveillance d'un pool de connexion JDBC.

Administration WebSphere avancée

Extensions

Présentation de la version WebSphere 8 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</p> <p>L'architecture JOnAS : le noyau, la couche services, la couche présentation, la couche application, le principe JMX et les Mbeans.</p> <p>Présentation du noyau JOnAS.</p> <p>Présentation de l'architecture OSGi.</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 JOnAS.</p>
Applicatif	<p>Terminologie : application web, container, sources de données, pilote JDBC, hôtes virtuels.</p> <p>Configuration des services essentiels : tomcat6, JNDI, quartz, ...</p> <p>Déploiement d'applications : mode manuel, mode automatique.</p> <p>Assemblage d'une application.</p> <p>Présentation de GenIC.</p>
Lien avec Tomcat	<p>Installation de Tomcat et modification des paramètres de base</p> <p>Activation du service Tomcat dans la configuration JOnAS.</p> <p>Le fichier tomcat6-server.xml.</p> <p>Administration du service web. Mise en place d'hôtes virtuels.</p>
Gestion des ressources	<p>Fournisseurs d'URL, fournisseurs JMS, sessionMail.</p> <p>Présentation de Joram et speedo.</p> <p>Mécanismes de connexion aux bases de données : JDBC, pool de connexions.</p> <p>Modification de la base de données cible.</p> <p>Travaux pratiques : 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>Travaux pratiques :</p> <p>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 oeuvre 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 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. Présentation du noyau JBoss. Présentation du micro-conteneur. Installation, configuration, arborescence des fichiers. Les services disponibles. Adaptation de la configuration : ajout/suppression de services Visualisation dans la console JMX de JBoss.</p>
Introduction à JBoss Portal	<p>Rappels sur les portlets : normes JSR-168 Portlets specifications (1.1) et JSR-286 : Portlets specifications (1.2) Fonctionnalités du portail : personnalisation, single sign on et agrégation de contenus Terminologie : portlets container, page, instance Positionnement de JBoss Portal et JBoss AS dans l'architecture JEMS.</p>
Installation et configuration	<p>Les prérequis techniques. Installation et configuration de base. Démarrage de JBoss Portal. Configuration du conteneur de portlets</p>
Développement	<p>l'API portlet, spécifications ; Etats et modes ; Descripteurs de déploiement; bibliothèque de tags des portlets, gestion du single sign on : 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
1540 €

17 au 24 mars
26 au 28 mai

13 au 15 octobre
3 au 5 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.

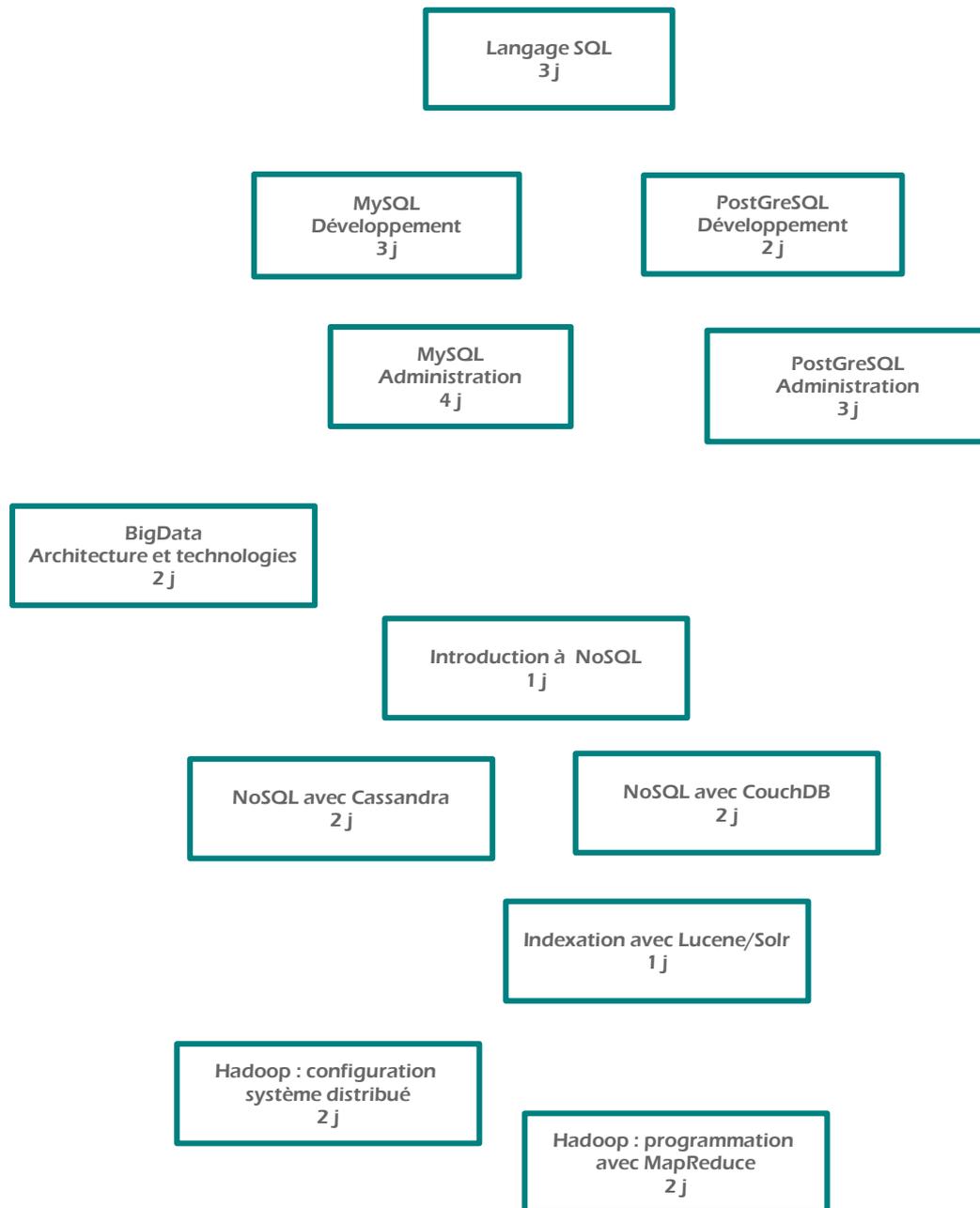
Programme:

JBoss présentation	Historique, présentation des différentes versions et de leurs caractéristiques, de JBoss AS V 5, 6, 7 à WildFly. 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: noeud, 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, configuration du réseau, fichiers de configuration.
Déploiement d'applications	Principe des applications réparties. Le service de déploiement. Travaux pratiques : création d'une application simple, déploiement sur un cluster JBoss. Tests : accès à l'application depuis chaque noeud de la partition.

Clusters JBoss

JGroups	<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 noeuds 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>
Applications	<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>
Routage HTTP	<p>Distribution de sessions HTTP. Routage de niveau 7. Les outils. Mise en place de pound, mod_jk, haproxy et mod_cluster. Travaux pratiques : mise en oeuvre de solutions pour la répartition de charge.</p>
Architecture	<p>Principe d'un référentiel central d'applications. Architecture d'exploitation pour les applications. Intégration des fichiers journaux. Regroupement vers une machine syslog. Travaux pratiques : Liaison d'une grappe avec une base de données répliquée.</p>
Automatisation	<p>Le langage de scripts. Les objets accessibles. Les commandes Twiddle ou JBoss-cli. 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>

Filières Bases de Données et BigData



Langage SQL

Durée: 3 jours
1440 €

4 au 6 février
25 au 27 mars
3 au 5 juin

16 au 18 septembre
21 au 23 octobre
2 au 4 décembre

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
1440 €

10 au 12 février
31 mars au 2 avril
10 au 12 juin

22 au 24 septembre
12 au 14 novembre

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 oeuvre des requêtes SQL.
Programmation serveur	L'intérêt de la programmation côté serveur, et les limites Mise en oeuvre 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
1910 € HT

17 au 20 février
7 au 10 avril
23 au 26 juin

29 septembre au 2 octobre
17 au 20 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 oeuvre de la réplication

PostgreSQL : Développement

Durée: 2 jours
945 €

24 au 25 février
14 au 15 avril

30 juin au 1er juillet
6 au 7 octobre
15 au 16 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	<p>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</p>
Installation et configuration	<p>Installation de PostgreSQL Installation avec un système de paquetage</p>
Interagir avec le serveur	<p>Le concept du client serveur Le client PostgreSQL Utiliser les clients graphiques</p>
Spécificités SQL de PostgreSQL	<p>Spécificités DDL (Définition des objets) Spécificités DML (Manipulation des données) Optimisation des requêtes</p>
Programmation coté serveur	<p>Importation et exportation de données L'intérêt de la programmation côté serveur et ses limites Mise en oeuvre de déclencheurs (triggers) et programmation de procédures stockées Déclaration et utilisation de variables Gestion des curseurs Mise en oeuvre avec PL/pgsql et présentation des autres outils disponibles</p>

PostgreSQL : Administration

Durée: 3 jours
1400 €

26 au 28 février
16 au 18 avril

2 au 4 juillet
8 au 10 octobre
17 au 19 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 requêtes 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 \ psql : installation, configuration.
Exploitation	Sauvegardes/Sauvegardes 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

BigData Architecture et technologies

Durée: 2 jours
1060 € HT

20 au 21 janvier
14 au 15 avril

8 au 9 septembre
6 au 7 novembre

Public:

Chefs de projets, architectes, et toute personne souhaitant connaître les outils et solutions pour mettre en place une architecture BigData.

Objectifs:

Comprendre les concepts du BigData et savoir quelles sont les technologies implémentées.

Connaissances préalables nécessaires:

Il est demandé aux participants d'avoir une bonne culture générale sur les systèmes d'information.

Programme:

Introduction	<p>Le besoin : volumes importants de données, traitements optimisés de flux de données au fil de l'eau, liés aux nouvelles technologies et aux nouveaux usages. Domaines concernés : recherche scientifique, médical, e-commerce, sécurité, ... Développement des techniques sur différents aspects : stockage, indexation/recherche, calcul. Définition ETL : Extract Transform Load. Les acteurs.</p>
Stockage	<p>Caractéristiques NoSQL : Structure de données proches des utilisateurs, développeurs Données structurées et non structurées, documents, images, fichiers XML, JSON, CSV, ... Les différents modes et formats de stockage. Stockage réparti : réplication, sharding, gossip protocol, hachage, Systèmes de fichiers distribués : GFS, HDFS, BigTable, .. Les bases de données. Quelques exemples de produits et leurs caractéristiques : cassandra, MongoDB, CouchDB, DynamoDB.</p>
Indexation et recherche	<p>Moteurs de recherche. Principe de fonctionnement. Méthodes d'indexation. Exemple de Lucene, et mise en oeuvre avec solr. Recherche dans les bases de volumes importants : Exemples de produits et comparaison : dremel, drill, elasticsearch, MapReduce,</p>
Calcul et restitution, intégration	<p>Différentes solutions : calculs en mode batch, ou en temps réel, sur des flux de données ou des données statiques. Les produits : langage de calculs statistiques, R Statistics Language Outils de calcul sur des volumes importants : storm en temps réel, hadoop en mode batch. Zoom sur Hadoop : complémentarité de HDFS et MapReduce.</p>

BigData Architecture et technologies

Evolutions

Les offres Saas BigData comme Google BigQuery.
Les limites. Les nouveautés annoncées

Introduction à NoSQL

Durée: 1 jour
630 € HT

31 janvier
30 avril

3 octobre
5 décembre

Public:

Experts en bases de données, chefs de projet et toute personne souhaitant comprendre le fonctionnement et les apports des bases NoSQL.

Objectifs:

Connaître les caractéristiques techniques des bases de données NoSQL, les différentes solutions disponibles. Identifier les critères de choix.

Connaissances préalables nécessaires:

Connaissance générale des systèmes d'informations et des bases de données.

Programme:

Introduction	origine des bases de données, les notions de transaction, les SGBD, la standardisation SQL, l'arrivée de nouveaux besoins : volumes importants liés aux technologies et aux nouveaux usages, traitements optimisés de flux de données au fil de l'eau. Développement des techniques sur différents aspects : stockage, indexation/recherche, calcul. Définition ETL : Extract Transform Load.
Caractéristiques NoSQL	Structure de données proches des utilisateurs, développeurs: sérialisation, tables de hachage, JSON. Priorité au traitement du côté client. Protocoles d'accès aux données, interfaces depuis les langages classiques. Données structurées et non structurées, documents, images, Stockage réparti : réplication, sharding, gossip protocol, hachage,.. Parallélisation des traitements : implémentation de MapReduce. Cohérence des données et gestion des accès concurrents : "eventual consistency" et multi-version concurrency control.
Principaux acteurs	Les solutions NoSQL et leurs choix techniques : CouchDB, MongoDB, Cassandra, HBase (Hadoop), Elasticsearch, .. Démonstrations avec Cassandra et couchDB. Critères de choix.
Mise en oeuvre	Points à vérifier : méthode d'utilisation des données format de stockage JSON, XML, choix de la clé, notion de clé composite, ... aspects matériels, besoins en mémoire, disques, répartition, .. import des données : outils et méthodes selon les moteurs NoSQL

Base de données NoSQL avec Cassandra

Durée: 3 jours
1590 € HT

22 au 24 janvier
16 au 18 avril

10 au 12 septembre
12 au 14 novembre

Public:

Chefs de projet, gestionnaires de bases de données.

Objectifs:

Connaître les apports de Cassandra, savoir l'installer et le configurer.

Connaissances préalables nécessaires:

Connaissances générales sur les bases de données.

Programme:

Introduction	Historique, fonctionnalités de Cassandra, licence Format des données, "key-value", traitement de volumes importants, haute disponibilité, système réparti de base de données, ...
Installation et configuration	Prérequis. Plate-formes supportées. Etude du fichier de configuration : conf/cassandra.yaml Répertoire de travail, de stockage des données, gestion de la mémoire. Démarrage d'un noeud et test de l'interface cassandra-cli.
Cassandra-cli	Commandes de base : connexion au système de base de données, création de colonnes, insertion, modification recherche, Le CQL : Cassandra Query Language. Exécution de scripts.
Clusters	Principe. Préparation du premier noeud : adresse d'écoute. Configuration de nouveaux noeuds. Notion de bootstrapping et de token. Paramètres listen_address et rpc_address. Réplication : topologie du réseau et EndpointSnitch. Stratégie de réplication. Ajout de noeuds, suppression. Cassandra dans un cloud. Mise en oeuvre avec OpenStack.
Exploitation	Sauvegardes. Import/export au format JSON. Supervision avec nodetool cfstats, ou export JMX vers des outils de supervision comme Nagios.
Support Hadoop	Principe de MapReduce. Implémentation Hadoop. Mise en oeuvre depuis Cassandra.

Base de données NoSQL avec CouchDB

Durée: 3 jours
1590 € HT

27 au 29 janvier
5 au 7 mai

29 septembre au 1er octobre
1er au 3 décembre

Public:

Chefs de projet, gestionnaires de bases de données.

Objectifs:

Connaître les apports de Couchdb, savoir l'installer et le configurer.

Connaissances préalables nécessaires:

Notions générales sur les bases de données.

Programme:

Introduction	Historique, fonctionnalités de CouchDB : stockage de données, traitement de flux de données, haute disponibilité, système distribué, ... Caractéristiques techniques : Format des données : documents JSON, gestion des accès concurrents par MVCC, architecture web, Licence.
Installation et configuration	Plates-formes supportées et prérequis. Installation depuis les packages. Démarrage du service. Tests depuis un client http.
Interfaces	Utilitaire en ligne de commande, curl : interrogation, création, suppression de bases de données. Aperçu du mode debug. Console d'administration web : Futon. Exécution de la chaîne de tests. Création de bases de données, de documents. Ecriture de fonctions de subdivision (map), et d'agrégation (reduce) en javascript pour générer des vues.
Les vues	Principe. Stockage sous forme de "design documents". Création de vues avec Javascript. Indexation des vues.
Exploitation	Sécurité : création d'utilisateurs, administrateurs, gestion des droits, chiffrement des mots de passe, fonctions de validation des mises à jour, authentification par cookie. Réplication : principes, utilisation de l'API REST standard, réplication incrémentale. Mise en oeuvre depuis Curl ou Futon ou un client http standard.

Indexation avec Lucene - Solr

Durée: 1 jour
510 € HT

30 janvier
23 mai

2 octobre
4 décembre

Public:

Chefs de projets, développeurs, et toute personne souhaitant utiliser Solr pour l'indexation de volumes importants de données.

Objectifs:

Comprendre le fonctionnement de Solr et savoir le mettre en oeuvre.

Connaissances préalables nécessaires:

Bonnes connaissances des systèmes d'exploitation.

Programme:

Introduction	Présentation de Solar, historique, fonctionnalités. Architecture technique. Lien avec Lucene.
Installation et configuration	Prérequis techniques. Configuration de la webapp solr. Mise en oeuvre avec tomcat. Etude du fichier solrconfig.xml
Indexation de documents	Différentes méthodes possibles : avec XML ou json, ou des fichiers CSV. Travaux pratiques de création, mise à jour de solr.
Requêtes de recherche	Les "RequestHandler" fournis par Solr. Mise en forme du résultat avec les "Response Writer". Optimisation des requêtes.
Exploitation	Interface d'administration Présentation de CloudSolr

Hadoop : configuration système distribué

Durée: 2 jours
1060 € HT

3 au 4 février
10 au 11 juin

6 au 7 octobre
8 au 9 décembre

Public:

Chefs de projet, administrateurs et toute personne souhaitant mettre en oeuvre un système distribué avec Hadoop.

Objectifs:

Connaître les principes du framework Hadoop et savoir l'installer et le configurer.

Connaissances préalables nécessaires:

Connaissance des commandes des systèmes unix/linux.

Programme:

Introduction	Les fonctionnalités du framework Hadoop Le projet et les modules : Hadoop Common, HDFS, YARN, MapReduce
Architecture	Terminologie : NameNode, DataNode, ResourceManager, NodeManager. Rôle des différents composants.
Installation et configuration	Première installation en local depuis les RPMs. Mise en oeuvre avec un seul noeud Hadoop. Configuration de l'environnement, étude des fichiers de configuration : core-site.xml, hdfs-site.xml, mapred-site.xml, yarn-site.xml et capacity-scheduler.xml Création des users pour les daemons hdfs et yarn, droits d'accès sur les exécutables et répertoires. Démarrage des composants : hdfs, hadoop-daemon, yarn-daemon, etc ..
Sécurité	Activation de la sécurité avec Kerberos dans core-site.xml, et dans hdfs-site.xml pour les NameNode et DataNode. Sécurisation de yarn avec la mise en oeuvre d'un proxy et d'un Linux Container Executor.
Exploitation	Supervision des éléments par le NodeManager. Visualisation des alertes en cas d'indisponibilité d'un noeud. Configuration des logs avec log4j.

Hadoop : développement avec MapReduce

Durée: 2 jours
1060 € HT

5 au 6 février
12 au 13 juin

8 au 9 octobre
10 au 11 décembre

Public:

Chefs de projets, développeurs, et toute personne souhaitant comprendre les techniques de développement avec MapReduce dans l'environnement Hadoop.

Objectifs:

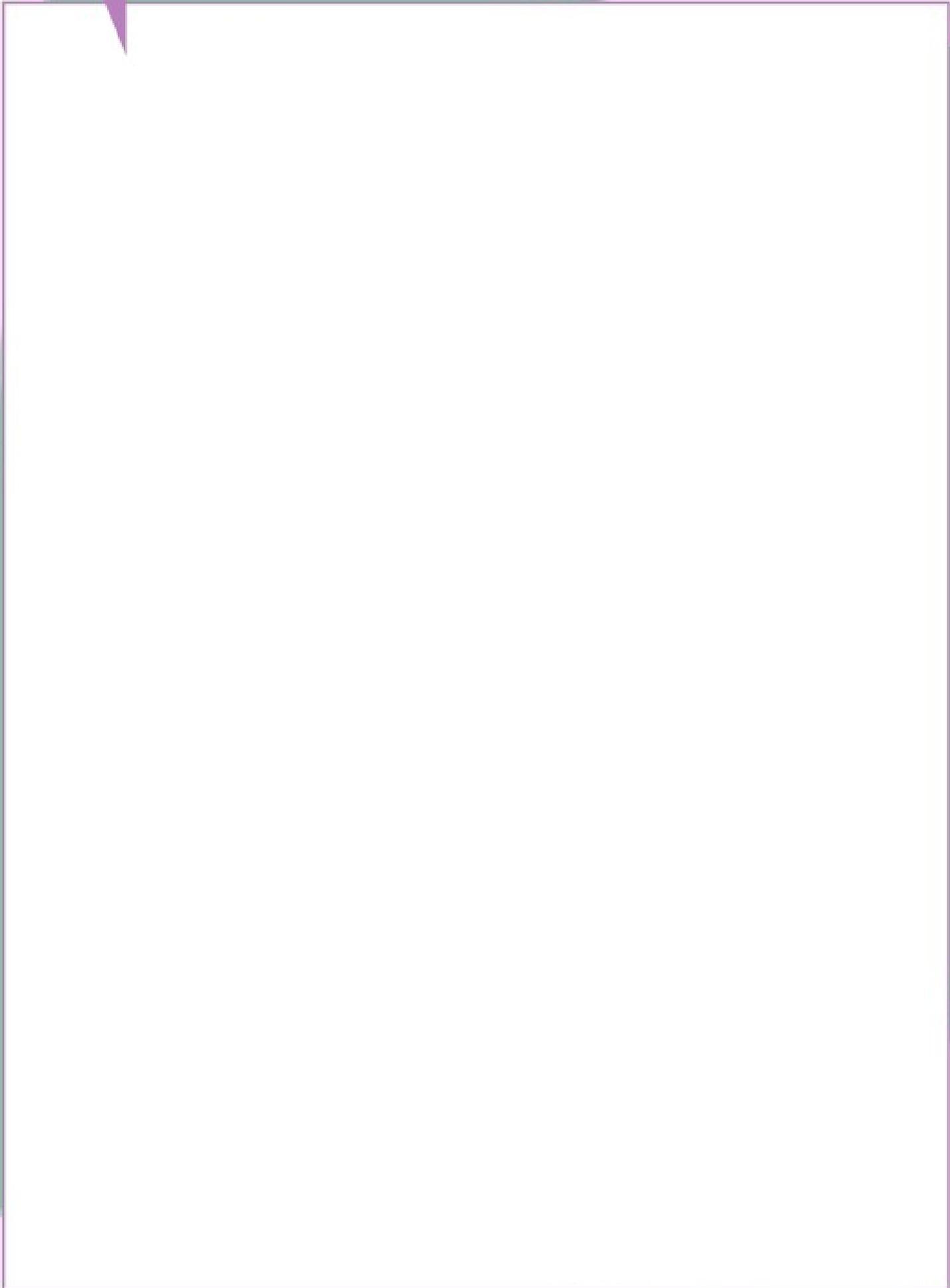
Connaître les principes du framework Hadoop et savoir utiliser la technologie MapReduce pour paralléliser des calculs sur des volumes importants de données.

Connaissances préalables nécessaires:

Connaissance d'un langage de programmation objet comme Java.

Programme:

Introduction	Les fonctionnalités du framework Hadoop Le projet et les modules : Hadoop Common, HDFS, YARN, MapReduce
MapReduce	Principe et objectifs du modèle de programmation MapReduce. Fonctions map() et reduce(). Couples (clés, valeurs). Implémentation par le framework Hadoop. Etude de la collection d'exemples. Rédaction d'un premier programme et exécution avec Hadoop.
Programmation	Les interfaces principales : mapper, reducer, partitioner, reporter, outputcollector. Format des entrées et sorties d'un job MapReduce : InputFormat et OutputFormat. Configuration des jobs avec jobconf. Interaction avec le JobTracker par JobClient.
Optimisation	Gestion de la mémoire, des accès disque. Utilisation de la JVM. Mise en oeuvre du cache distribué.
Securité	Configuration des ACLs. Gestion de l'authentification.



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

Catalogue des formations 2014

Pythagore F.D. : Apprendre à Apprendre		
Filières Systèmes		
Filières Systèmes Linux		
Linux/unix introduction	UX100	6
Le Shell	UX002	8
Shell avance	UX004	9
Administration UNIX multi-plateformes	UX011	11
Administration AIX	UX018	13
Administration AIX avancée	UX019	15
Administration Solaris 11	UX040	17
Administration avancée Solaris	UX041	19
Introduction au logiciel libre	UX107	20
Administration Linux	UX111	21
Les services réseaux Linux	UX112	24
Haute disponibilité Linux	UX115	26
Linux système sécurisé	UX117	28
Linux sécurité des accès	UX118	30
Linux : optimisation performances métrologie	UX119	32
Administration avancée Linux	UX140	33
Administration Ubuntu Server	UX142	36
Preparation certification LPI101	UC101	39
Preparation certification LPI102	UC102	40
Préparation certification LPI201	UC201	41
Preparation certification LPI202	UC202	42
Filières Systèmes embarqués et mobilité		
Linux temps réel	UX121	44
Systèmes Linux embarqués	UX125	45
openWrt	UX126	47
Développement d'applications Android	UX128	49
Industrialisation Android	UX131	51
Initiation au developpement iPhone/iPad	MC030	52
Expertise Developpement iPhone iPad	MC040	54
Filières Réseaux et TCP/IP		
Introduction aux réseaux	RS001	57
TCP/IP : protocoles et mise en oeuvre	RS003	59
Administration réseaux IP	RS007	61
Routage IP avancé	RS009	62
IP: Migration vers IPv6	RS014	63
Messagerie	RS006	65
Sécurité TCP/IP	RS018	66
Mise en oeuvre du protocole SNMP	RS022	67
Voix sur IP	RS023	69
Performances et optimisation des réseaux	RS025	71
Annuaire LDAP	RS122	72
Zimbra : installation et configuration	RS132	74
Filière Production et supervision		
Supervision nagios : utilisation	RS129	76
Administration Nagios	RS130	78
Supervision avec shinken	RS135	81
Reporting avec Cacti	RS134	82
Gestion de Parc avec OCS et GLPI	UX124	83
Sauvegardes avec Bacula	UX160	85
Puppet : administration centralisée	UX161	86
Filières Internet et Web		
Serveur WEB : apache	RS104	88
Administration serveur Nginx	RS105	90
Serveur Proxy Squid	RS114	92
Sécurisation des services internet	RS115	93
HTML et CSS	RS102	95
Web Dynamique avec JavaScript	RS106	97
Web 2.0 - (X)HTML, CSS, Javascript	RS128	98
CMS : mise en oeuvre de Joomla	RS127	100
Google Apps Utilisation	AW002	101
Google App Engine	SY200	102
Développement Web avec GWT	LW001	103

Réseau social d'entreprise : techniques et outils	RS136	104
Filières Virtualisation et Cloud		
Virtualisation Linux	SY011	106
Virtualisation avec Xen	SY004	107
Virtualisation avec KVM	SY007	109
Virtualisation avec lxc	SY008	111
Administration de machines virtuelles	SY010	112
Cloud : technologies et enjeux	SY100	114
Architecture cloud d'entreprise	SY101	115
Eucalyptus : configuration cloud d'entreprise	SY110	117
CloudStack : mise en oeuvre	SY112	119
OpenStack : configuration et administration	SY111	120
OpenStack : utilisation	SY131	122
Filières Développement		
Programmation en langage C	LC001	124
Perfectionnement en langage C	LC002	126
Programmation système en C sur Unix/Linux	LC010	128
Programmation noyau et drivers en C sur Linux	LC011	130
Debugging en C sur Unix	UX015	131
Le langage C++	LC004	132
Introduction à Objective-C	LC005	133
Développement d'applications Web avec PHP	LH001	134
Initiation au langage Pascal	PA001	136
Programmation Perl	LP001	137
Perl avancé	LP002	139
Le langage Python	LY001	140
Développement avancé avec Python	LY002	142
XML Développement	AS120	143
Filières Langages Java et JEE		
Concepts Objets et programmation Jav	LJ001	146
Programmation avancée en Java	LJ002	148
Sécurité Java et JEE	LJ005	150
JEE : Développement d'applications web	AS004	152
JEE : les EJB	AS005	153
Développement Web Services	AS006	155
Techniques et outils de migration vers les EJB3	AS007	157
Serveurs d'application JEE	AS011	158
Développement JEE avec WebSphere	AS101	160
Développement JEE avec JOnAS	AS103	162
Applications JEE avec JBoss	AS104	164
Conception JEE avec les Design Patterns	AS122	166
Administration WebSphere	AS131	167
Administration Tomcat	AS133	169
Administration JBoss	AS134	171
Administration WebSphere avancée	AS135	173
Exploitation JOnAS	AS136	175
JBoss Portal	AS137	177
Clusters JBoss	AS144	179
Filières Bases de Données et BigData		
Langage SQL	BD001	182
MySQL : Développement	BD010	183
MySQL : Administration	BD011	184
PostgreSQL : Développement	BD020	185
PostgreSQL : Administration	BD021	186
BigData Architecture et technologies	CB001	187
Introduction à NoSQL	CB002	189
Base de données NoSQL avec Cassandra	CB010	190
Base de données NoSQL avec CouchDB	CB011	191
Indexation avec Lucene – Solr	CB012	192
Hadoop : configuration système distribué	CB013	193
Hadoop : développement avec MapReduce	CB014	194