

Catalogue des formations 2016

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 et Linux, et les applicatifs Apache, Openldap, Squid, Nagios, Zabbix, OCS/GLPI, puppet , chef...
- la virtualisation avec xen, kvm, lxc, et le cloud : cloudstack et openstack
- TCP/IP (IPv6, snmp, Architecture, Sécurité, Administration de réseaux IP, VoIP, ...)
- Développement (langage C, Java, Jee, technologies Jee, JBoss, WebSphere, WebServices, PHP, Perl, Python , ...) et le développement sur mobiles android
- les bases de données et le BigData avec NoSQL, Cassandra, MongoDB, Hadoop, ...

Sur chacun de ces domaines, notre équipe possède un excellent niveau d'expertise couvrant l'ensemble du domaine , des basiques aux outils les plus complexes : nos formations vont de l'introduction à Linux, à la Sécurité, la Haute Disponibilité, des concepts NoSQL à la programmation MapReduce

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 2016 :

Les technologies BigData sont entrées dans une phase opérationnelle, et les questions de production, de fiabilisation, sont d'actualité : parmi nos nouveautés 2016, la « **Programmation R pour Hadoop** » pour utiliser les techniques de calculs statistiques dans un environnement Hadoop, ou « **L'optimisation de calculs avec Spark** » répondent à ces préoccupations.

Du côté de l'administration , toujours en production, des outils comme chef, ou puppet, docker, permettent de gérer des parcs systèmes et applicatifs de manière industrialisée et automatisée : nos nouveaux stages comme la « **Gestion de configuration avec Chef** » ou « **Docker, mise en oeuvre** » montrent comment déployer rapidement et en toute sécurité, des applications et systèmes.

Enfin, du côté de l'internet des objets, nous proposons un stage pour aborder l'ensemble des technologies mises en oeuvre dans les **objets connectés** : depuis les systèmes embarqués, jusqu'au stockage des données collectées dans le cloud, et l'analyse par les outils de BigData.

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 physiques de 6 à 10 places chacune,
- 10 salles de formation à distance accessible partout à l'aide de la solution de classes virtuelles CALIAMIS
- 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;
- serveurs de données (Hadoop 7 To pour les travaux pratiques BigData, grappes OpenStack
- 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 (120 Mbps 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.

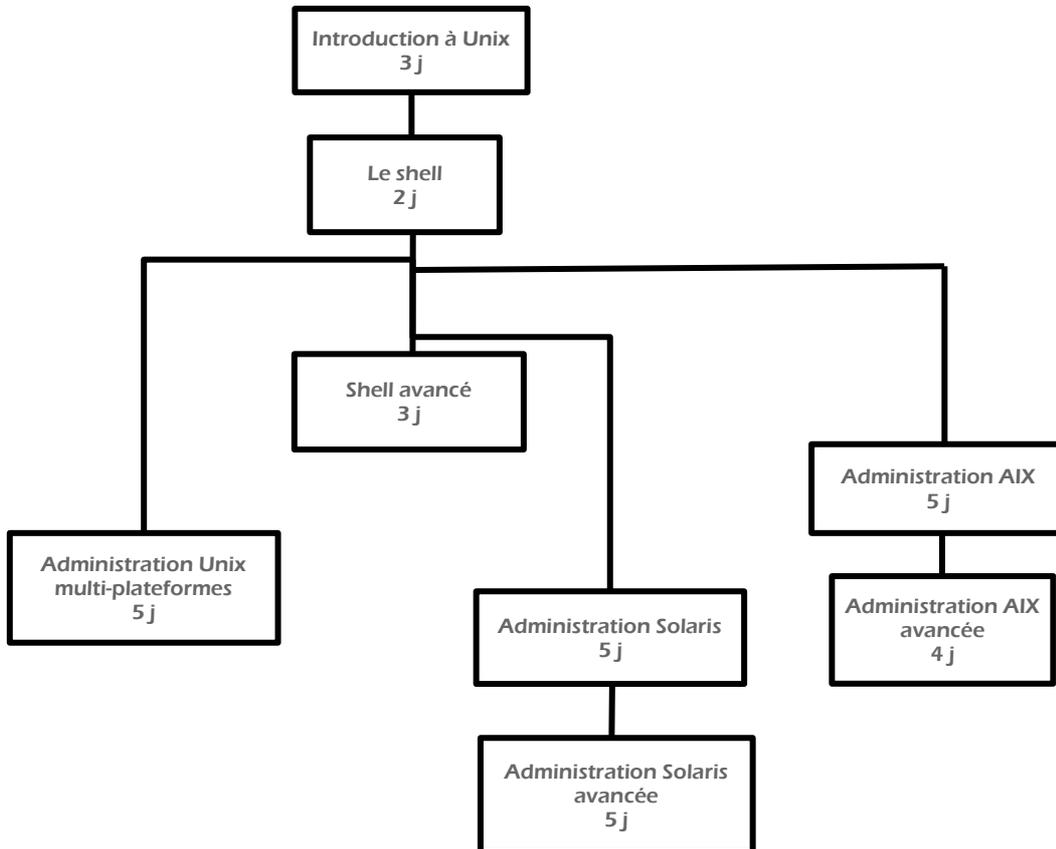
Les participants à ces stages sont très heureux de pouvoir bénéficier des avantages des formations en centre, tout en évitant les déplacements.

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

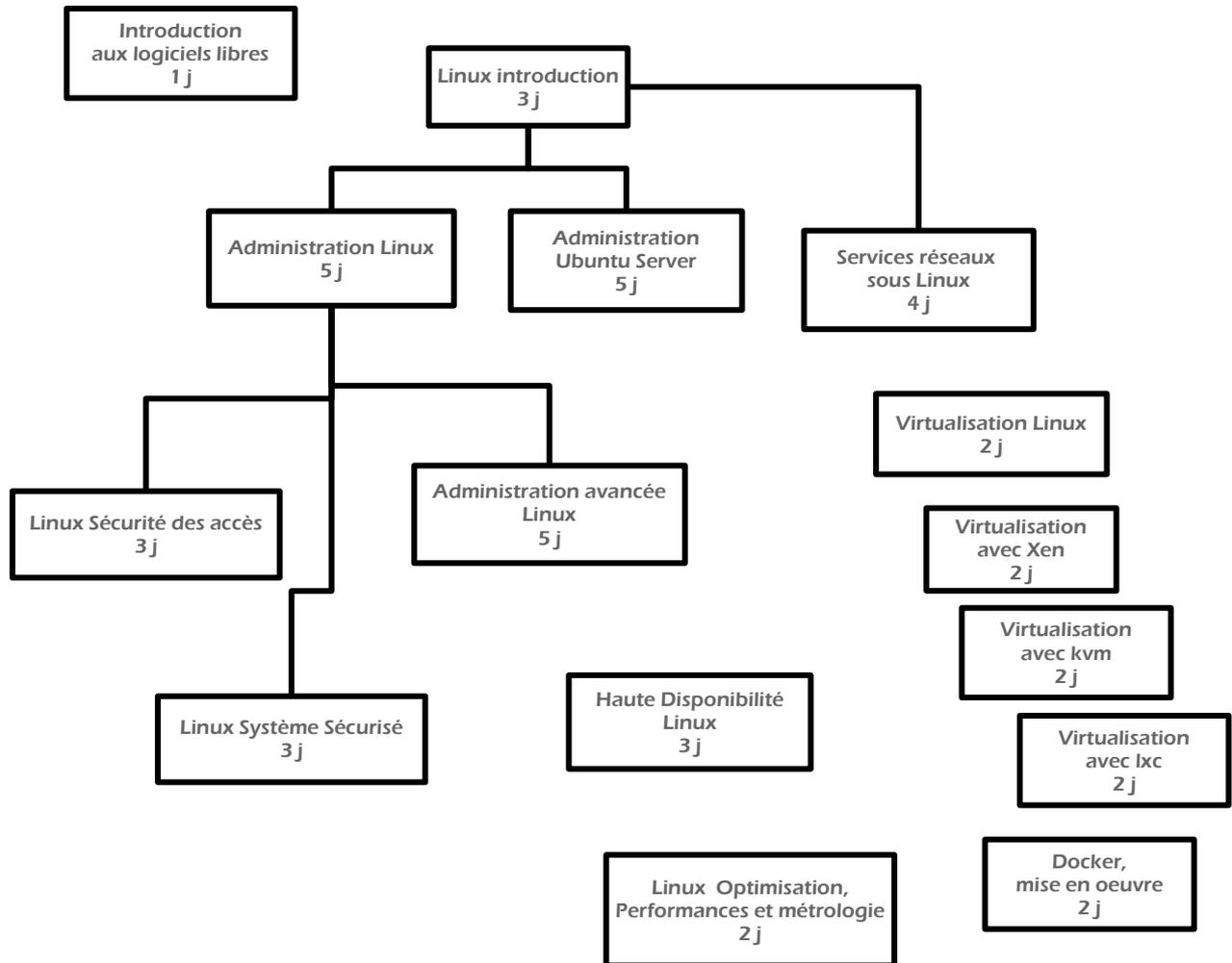
Si vous avez des besoins particuliers dans l'un de nos domaines d'expertise, n'hésitez pas à faire appel à nos services : nous pouvons élaborer des plans de formation et des supports de cours sur mesure.

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

Filières Systèmes



Filières Systèmes Linux



Linux/unix introduction

Durée: 3 jours
1505 €

25 au 27 janvier
7 au 9 mars
23 au 25 mai

29 au 31 août
17 au 19 octobre
21 au 23 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	Organisation des données sur un serveur Unix Structure des disques Le système de 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

Linux/unix introduction

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.
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
1030 €

28 au 29 janvier
10 et 11 mars
26 au 27 mai

1er et 2 septembre
20 au 21 octobre
24 au 25 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

Atelier : Shell avance

Durée: 3 jours
1535 €

8 au 10 février
17 au 19 mai

5 au 7 septembre
28 au 30 novembre

Public:

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

Objectifs:

Appliquer les techniques avancées du shell. Connaître les outils disponibles et les bonnes pratiques concernant la programmation de scripts. Ce stage est réalisé sous la forme d'un atelier de travaux pratiques

Connaissances préalables nécessaires:

Connaissance des principes de base du système UNIX/Linux et de la programmation shell de base.

Programme:

Rappels techniques	<ul style="list-style-type: none"> Différents modes d'exécution des scripts : nohup, exec, at, ... Interruption des scripts : les signaux Le suivi de consommation : time eval : réinterprétation d'une commande select : gestion de menus (ksh) getopts : décoder les options d'un script Programmation parallèle Programmation événementielle Gestion des alertes Le debugging et l'optimisation
Programmation m4	<ul style="list-style-type: none"> utilisation de la protection, encapsulation
Les outils	<ul style="list-style-type: none"> grep, awk, sed. Les variables disponibles avec awk,; les fonctions. opérateurs, le contrôle d'exécution Utilisation des expressions régulières dans sed Les sous-expressions
Mise en oeuvre	<ul style="list-style-type: none"> Développement en shell d'un superviseur Unix/Linux et réseau Fonctionnalités : console centrale de supervision Visualisation de l'état de chaque poste supervisé. Journalisation des opérations et états Gestion des utilisateurs, ressources (mémoire, cpu, disques). Supervision de processus.

Administration UNIX multi-plateformes

Durée: 5 jours
2230 €

1^{er} au 5 février
4 au 8 avril
27 juin au 1^{er} juillet

3 au 7 octobre
28 novembre au 2 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

Administration UNIX multi-plateformes

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
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
2230 €

15 au 19 février
30 mai au 3 juin

10 au 14 octobre
12 au 16 décembre

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

Administration AIX

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

Administration AIX avancée

Durée:4 jours

Prix et dates: nous consulter

Public:

Exploitants et administrateurs de systèmes AIX.

Objectifs:

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

Connaissances préalables nécessaires:

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

Programme:

ODM	Les bases ODM. Administration ODM. Gestion des informations. Classes et objets. Fichiers de configurations. Création d'objets : odm_create_class, odm_add_obj, odmcreate, odmdrop. Création d'une nouvelle base. Les commandes en mode ligne : odmdelete, odmadd, odmchange, odmshow,odmget.
Installation AIX	Lancement. Options d'installation. Menu post-install. Fichiers d'installation.SMS/AIX.
Installation réseau	Installation AIX en réseau, principe de NIM Mise en place d'un maître NIM : nim_master_setup, nimconfig. Informations sur l'environnement NIM.Clients NIM.NIM avec wsm. nimol. Mise en 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
2230 €

8 au 12 février
9 au 13 mai

12 au 16 septembre
5 au 9 décembre

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 rôles (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

Administration Solaris 11

Ordonnancement	At, cron, fichier crontab
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

Prix et dates: nous consulter

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. Rappels sur la gamme de machines Sparc : processeurs M5, M6, T5, Sparc64.
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. LDoms/Oracle VM Server for SPARC : architecture, hyperviseur, vswitch, vnet. Affectation de ressources: processeur, disque, réseau.
Configuration réseau avancée	Configuration ip : nwadm, ipadm, dladm Agrégation de liens avec IPMP. groupes, le démon in.mpathd, ippmpstat. Architecture, mécanismes de basculement. Agrégation de machine à l'aide de VRRP. Mise en place d'un routeur VRRP. Mise en évidence du basculement en cas de panne. Répartition de charge avec ILB, algorithmes, direct, DNAT, NAT complet.

Administration avancée Solaris

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
510 €

4 mars
13 mai

1er septembre
25 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.

Introduction au logiciel libre

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
2230 €

15 au 19 février

18 au 22 avril

6 au 10 juin

19 au 23 septembre

28 novembre au 2 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).
Les apports de systemd

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

Systèmes de fichiers

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.

Utilisateurs

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.

Processus

Les processus. Les threads. Gestion des priorités.
 Utilisation des pseudo-processus /proc: stat, cpuinfo...

Sécurité des données

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.

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

Prix et dates: nous consulter

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 Tcpdump 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.

Les services réseaux Linux

Messagerie	Les protocoles, POP3 et IMAP4 Structure des messages. Présentation de Postfix, installation et configuration. Fichiers master.cf, main.cf
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
1560 €

3 au 5 février
2 au 4 mai

29 au 31 août
14 au 16 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, ldirectord, 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 ldirectord</p> <p>Configuration de Pacemaker pour la gestion du cluster.</p>

Haute disponibilité Linux

IPVS

Présentation : IP Virtual Server.
Répartition de charge.
Contrainte au niveau du noyau.
Travaux pratiques :
préparation d'un noyau IPVS, configuration passerelle.
Mise en place d'un cluster.

Idirectord

Présentation : Linux director daemon.
Fonctionnalités.
Travaux pratiques :
installation et configuration de Idirectord

Applications

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.

Données

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.

Linux système sécurisé

Durée: 3 jours
1560 €

7 au 9 mars
20 au 22 juin

19 au 21 septembre
30 novembre au 2 décembre

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.

Linux système sécurisé

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

Sécurisation traitements

Les risques : le déni de service, exemples de virus sur un système Linux.

Travaux pratiques :

exploitation d'un débordement de pile.

Les moyens de détection, la surveillance, les traces :

syslog, l'accounting.

L'audit de sécurité.

Méthodes de protection : démarche sur les systèmes Linux.

Sécurité du noyau

Les différentes approches de sécurisation du noyau.

Présentation de GrSecurity et SELinux.

Travaux pratiques avec GrSecurity :

installation, configuration du noyau, paramétrage du niveau de sécurité.

Administration avec grAdm2.

Génération d'une politique : learning mode.

Mise en place des règles d'ACL.

L'ACL GrSec.

Restrictions d'accès aux appels systèmes. Masquage de processus.

Visibilité du répertoire /proc.

Restrictions chroot.

SELinux : principe, configuration du noyau, options du noyau.

Travaux pratiques :

définition d'une politique de sécurité.

Installation et activation de la politique de sécurité dans le fichier /etc/selinux/config.

Linux système sécurisé

Sécurité des données

Contrôle de la cohérence du système de fichiers : fsck.
Procédure de vérification.
Sauvegardes : définitions
Commandes et outils standards.
Utilisation des sauvegardes pour la disponibilité des données.
Outils sauvegarde/archivage/compression :
gzip, zip, tar, dump, restore, dd, cpio, rsync
Service d'urgence pour Linux :
en cas de problème au démarrage du système,
utilisation d'un système tiers : "systemRescue CD"
Travaux pratiques :
création de CD de secours.

Sécurité système de fichiers

Sécurité: mise en place des contrôles d'accès
ACL : principe des listes de contrôle d'accès POSIX.
Travaux pratiques : mise en place des ACL sur xfs
Les quotas : principe, mise en place dans le fichier /etc/fstab.
La commande edquota pour l'édition, et le paramétrage, et la
commande quota pour la visualisation.
Travaux pratiques : mise en place des quotas

Linux sécurité des accès

Durée: 3 jours
1560 €

14 au 16 mars
27 au 29 juin

26 au 28 septembre
5 au 7 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 (Opensource Vulnerability Assessment Scanner) : principe de fonctionnement, installation. Travaux pratiques : réalisation d'un audit réseau avec openVAS.

Linux sécurité des accès

Sécurisation des accès
réseau

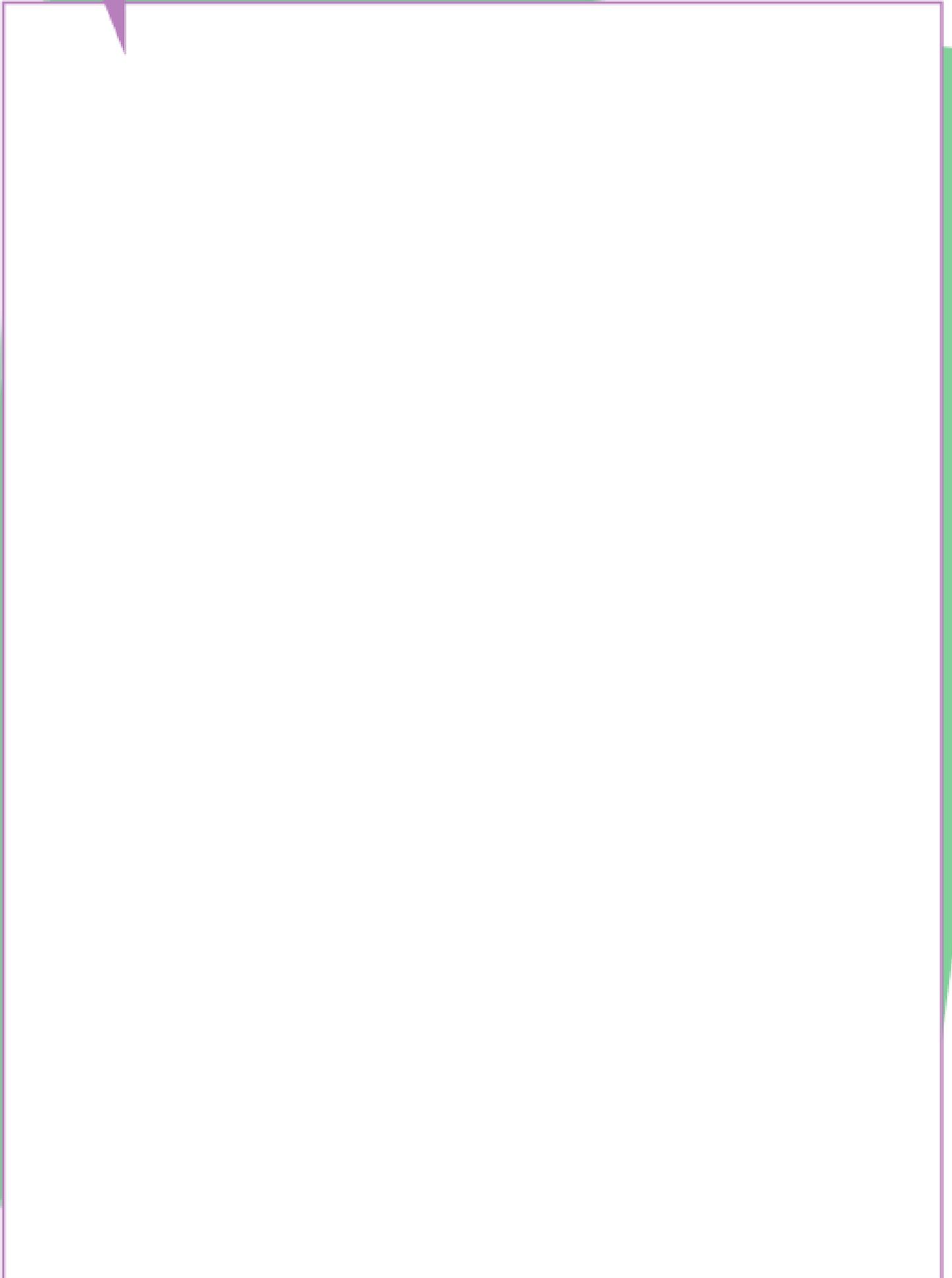
Protection de services réseaux au travers de xinetd.
Les tcp-wrappers: telnet, tftp, snmp, ftp, pop3s, imap4s
Les contrôles d'accès :
Etude des fichiers /etc/hosts.allow et /etc/hosts.deny
Les accès réseaux : sftp, les r-commandes (rlogin, rsh)
Sécurisation des transferts de fichiers avec vsftp
Présentation 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
1095 €

10 au 11 mars
23 au 24 juin

22 au 23 septembre
8 au 9 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. Utilisation des pseudo-systèmes /proc et /sys: stat, cpuinfo, ... Utilisation des processus système: kswapd, swpctl, rsyslogd Commandes : vmstat, lscpu, chcpu TP : création d'un utilitaire d'extraction des informations système.
Outils	Présentation des outils oprofile, sysstat, dstat, tuned, tuned-adm TP avec oprofile.
Systèmes de fichiers	Les différents types de systèmes de fichiers. Les systèmes natifs : ext2, ext3, ext4, xfs, Gestion de la fragmentation, pagination. Les systèmes émulés : vfat, ntfs. Les systèmes distribués : nfs, cifs Options : rsize, wsize, timeo, retrans,... TP : outil de mesure des accès.
Utilisateurs	accounting, quotas, fichiers de logs. ulimit, prlimit.
Réseau	Commandes : iptraf, nstat, rtacct Exploitation des éléments statistiques produits Gestion des ressources, qos avec tc, ifstat. Outils mrtg, rrdtool, SystemTap, DTrace, Phoronix test suite, TSung
Cgroups	Gestion des performances. Limitations des ressources affectées à un ou plusieurs processus. Introduction au cloisonnement.

Administration avancée Linux

Durée: 5 jours
2265 €

29 février au 4 mars

25 au 29 avril

13 au 17 juin

3 au 7 octobre

Public:

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

Objectifs:

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

Connaissances préalables nécessaires:

Connaître les techniques d'administration d'un système Unix ou Linux.

Programme:

Distribution

Présentation : RedHat Package Manager.

Les distributions qui utilisent les rpm.

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

Détails de la commande rpm, et de ses options.

Travaux pratiques :

mise en 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

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.

RAID

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.

Authentification en production

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

Administration avancée Linux

Performances

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.

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, cpubinfo, ...</p>
Sauvegardes	<p>Outils sauvegarde/archivage/compression: gzip, zip, tar, dd, cpio Sauvegarde du système, création de CD de secours. Travaux pratiques : sauvegarde par cpio, archivage par tar Commandes de sauvegardes en réseau : rsync, clonezilla. Présentation de l'outil backuppc.</p>
Impressions	<p>Les services d'impression, démarrage/arrêt des services d'impression. Présentation de CUPS : Common Unix Printing system Définitions : classes d'imprimantes, classes implicites, destination, filtres, backends. Travaux pratiques : installation d'une imprimante, modification d'un pilote : utilisation de lpr, cups, printtool, system-config-printer.</p>

Administration Ubuntu Server

Programmation de tâches

Le besoin, l'automatisation des tâches systèmes
Exécution différée avec at.
Programmation de tâches avec cron.
Etude du fichier crontab.

Réseau IP

Les objets à configurer :
les interfaces réseaux, les routes, le DNS.
Principe de la configuration dynamique ou statique.
Configuration, nommage/activation des interfaces réseau, drivers.
Etude des fichiers /etc/hosts, /etc/nsswitch, /etc/resolv.conf.
Travaux pratiques :
création d'une interface réseau, visualisation,
configuration de plusieurs adresses IP sur la même interface physique,
ajout d'une route, d'un hôte, d'un serveur DNS, et tests.
Les utilitaires ssh, clients windows (Putty, WinSCP).
Sécurisation avec netfilter et iptables : principes de base.

Exploitation

Configuration de syslog
Journaux : /var/log/messages
surveillance des logs avec logwatch
Vérification d'intégrité du système de fichiers avec AIDE

Virtualisation Linux

Durée: 2 jours

Prix et dates: nous consulter

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.
lxc	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

Prix et dates: nous consulter

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.

Virtualisation avec Xen

Supports de stockage	Déclaration des espaces de stockage accessibles au domU : périphériques blocs, partitions physiques, ou volumes logiques.
Systèmes 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
1115 €

18 au 19 février
12 au 13 mai

22 au 23 septembre
24 au 25 novembre

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>

Virtualisation avec KVM

Migration d'images

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

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 switches VDE.
Configuration des switches avec Unixterm.

Virtualisation avec lxc

Durée: 2 jours
1115 €

24 au 25 mars
26 au 27 mai

8 au 9 septembre
7 au 8 novembre

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	<p>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.</p>
Cgroup	<p>Fonctionnement de Control Group. Vérification de la configuration du noyau. Activation des Cgroups.</p>
Les outils LXC	<p>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...</p>
Gestion des conteneurs	<p>Configuration, création, démarrage. Utilisation des templates pour créer des conteneurs standards. Choix des systèmes de fichiers.</p>
Configuration du réseau	<p>Les différentes méthodes : interface physique, pont/commutateur virtuel, vlan. Mise en oeuvre. Configuration d'un point par brctl.</p>

Docker : mise en oeuvre

Durée: 2 jours

1030 € HT

du 18 au 19 février

du 2 au 3 juin

du 20 au 21 octobre

du 1er au 2 décembre

Public:

Administrateurs, chefs de projet et toute personne souhaitant mettre en oeuvre Docker pour déployer ses applications.

Objectifs:

Comprendre et savoir mettre en oeuvre Docker, et les produits associés.

Connaissances préalables nécessaires:

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

Programme:

Introduction

Présentation docker, principe, fonctionnalités

Besoins : packaging d'applications, déploiement rapides, coexistence de plusieurs versions d'une application sur un même serveur.

Principe et architecture

Les containers lxc.

Présentation de lxc : Linux containers, historique, principe de fonctionnement.

Les Cgroups.

L'isolation de ressources, création d'un environnement utilisateur.

Positionnement par rapport aux autres solutions de virtualisation.

Apports de Docker : Docker Engine pour créer et gérer des containers Dockers.

Plate-formes supportées.

Produits complémentaires : Docker Machine, Docker Compose, Kitematic, Docker Swarm, Docker Registry

Installation et configuration

Prérequis techniques et travaux pratiques d'installation sur Linux
Création d'un groupe Docker.

Gestion des images et des containers

Utilisation de DockerFile pour créer de images :

principales instructions (RUN, FROM, ENV, EXPOSE, etc ...)

Recommandations et bonne pratiques d'écriture de DockerFile.

Gestion des containers : création, affichage, sauvegarde de l'état

Volumes de données

Initialisation des volumes de données lors de la création d'un container.

Ajout de volumes de données, contrôle avec la commande Docker inspect

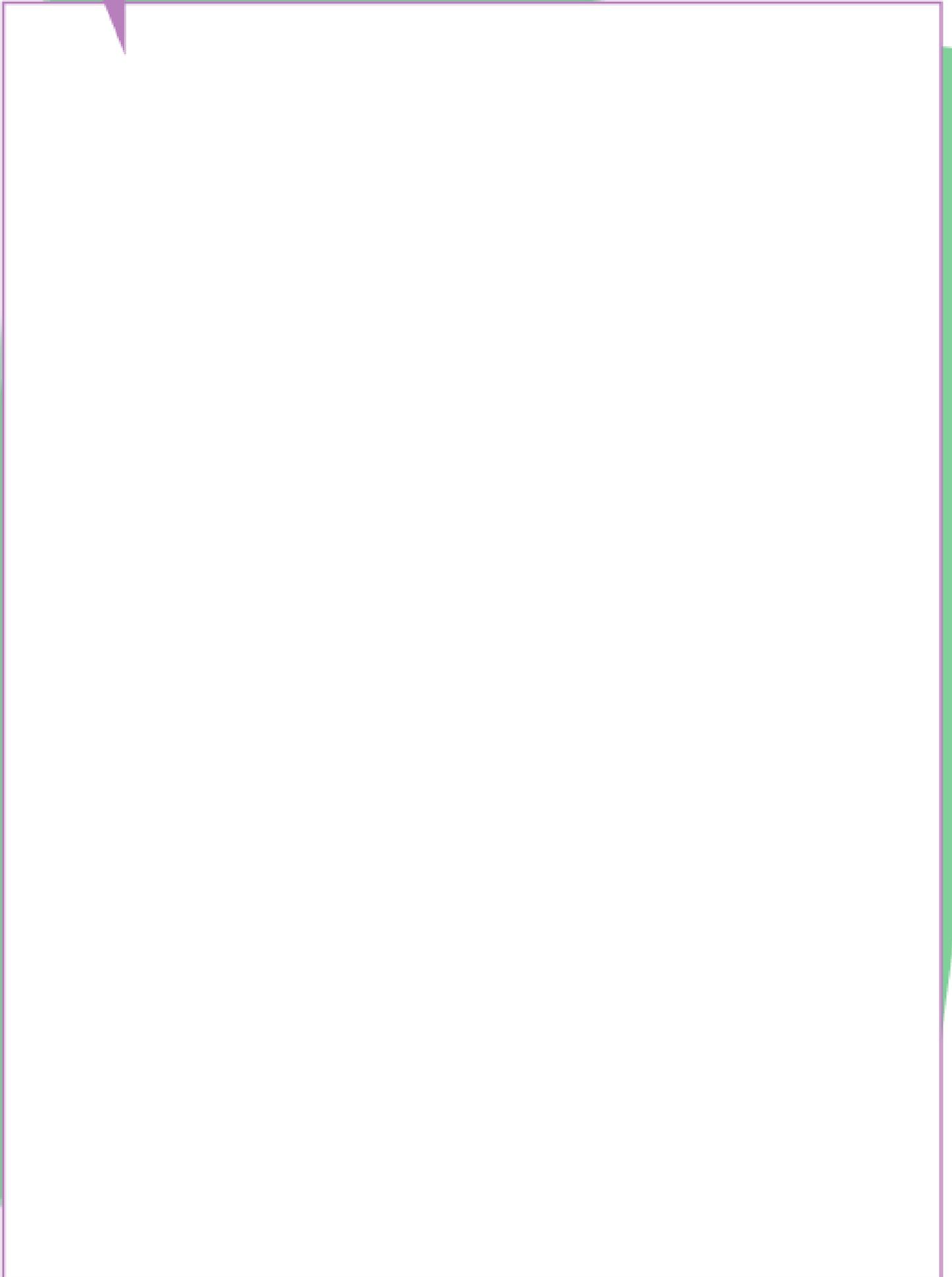
Sauvegarde, migration, restauration de volumes

Création de containers de volume de données

Docker : mise en oeuvre

Administration

Applications multi-containers avec Compose
Présentation de Swarm pour le clustering
Configuration réseau et sécurité dans Docker.





Filières Systèmes embarqués et mobilité

Linux Embarqué
2 j

Développement
Applications Android
4 j

Systèmes Linux embarqués

Durée: 2 jours

Prix et dates: nous consulter

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, PCduino3, CubieBoard2, Raspberry Pi2)
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.

Systèmes Linux embarqués

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.

Développement

Ajout de modules au noyau.

Travaux pratiques : Portage de net-snmp.

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.

Travaux pratiques : 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
1915 €

29 février au 3 mars
17 au 20 mai

10 au 13 octobre
12 au 15 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.

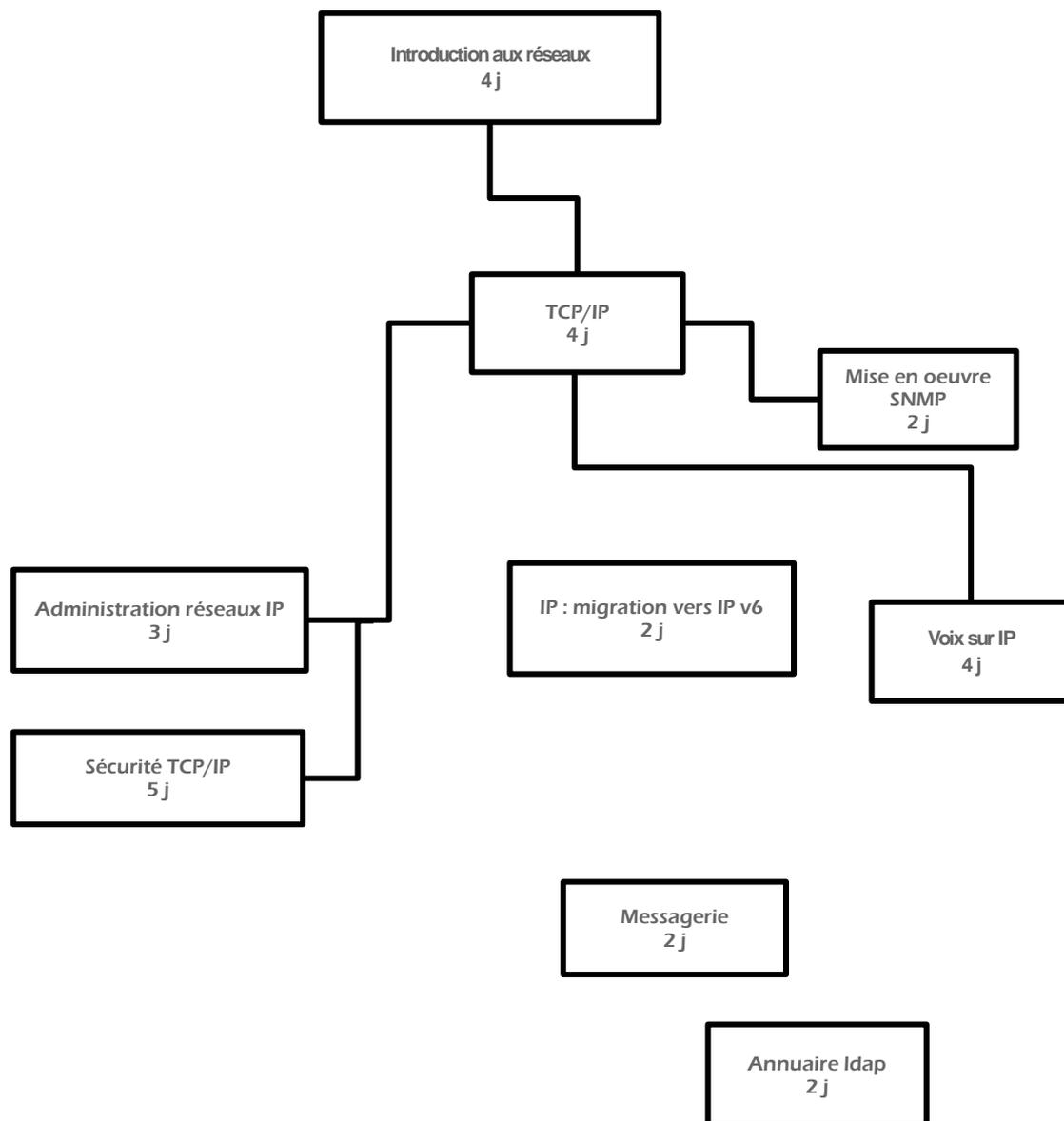
Programme:

Introduction	Présentation du système d'exploitation Android et historique des versions : de la v1 jusqu'à la v4. Les apports de la version Kit Kat. 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. Boites 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

Filières Réseaux et TCP/IP



Introduction aux réseaux

Durée: 4 jours
2025 €

1er au 4 février
17 au 20 mai

12 au 15 septembre
17 au 20 octobre
28 novembre au 1er 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

Introduction aux réseaux

Utilisation du Wifi	Points d'accès Modes de fonctionnement, mode répéteur, Mode pont Alignement d'antennes, supervision de 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 QoS
IPV6	Besoin, fonctionnalité La trame IPV6, adressage
Sécurisation	VPN et tunnels: Objectif, fonctionnement DMZ et Pare-feux : Définition, serveur Proxy, fonctionnement pare-feux et tunneling Filtrage: les iptables, politique par défaut, état des connexions, traduction d'adresses, traduction de ports, connexion à internet
Voix sur IP	Commutation de paquets Avantages de la voix sur IP Les protocoles : H323, SIP Introduction RTP : définition et applications, RTP et Nat Utilisation du registrar SIP avec Asterisk Création des comptes téléphones, du dialplan, vérification et tests Enregistrements SRV : serveurs DNS et Asterisk Transport de données Bande passante et qualité de service (QoS)
Evolutions	L'adressage IP, la sécurité, les réseaux de stockage.

TCP/IP : protocoles et mise en oeuvre

Durée: 4 jours
1770 €

15 au 18 février
30 mai au 2 juin

26 au 29 septembre
12 au 15 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.
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.

TCP/IP : protocoles et mise en oeuvre

IPv6

Adressage actuel, attribution des adresses.

Le travail de l'IETF (Bradner/Mankin). Plan d'adressage sur 128bits.

Agrégateurs : découpage TLA/NLA/SLA/IID. Intégration des Regional Registries

Fonctionnement : Surcharge d'entêtes. Structures des trames.

Les nouveaux mécanismes: fragmentation: MTU universelle, DHCPv6, dynamic DNS, renumérotation simplifiée d'un plan d'adressage

Sécurité

Ipssec (IP Security Protocol)

TP de mise en oeuvre

IP: Migration vers IPv6

Durée: 2 jours
1035 €

10 au 11 février
26 au 27 mai

3 au 4 octobre
19 au 20 décembre

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

Mise en oeuvre du protocole SNMP

Durée: 2 jours
1035 €

8 au 9 février
2 au 3 mai

22 au 23 septembre
28 au 29 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).

Mise en oeuvre du protocole SNMP

Les alertes	Création d'un serveur d'alertes avec snmptradd. Définition des conditions d'alertes pour chaque objet.
Sécurité	Authentification Protection du contenu
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.

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.

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

Voix sur IP

Durée: 4 jours
2010 €

8 au 11 février
28 juin au 1er juillet

17 au 20 octobre
5 au 8 décembre

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.

Voix sur IP

Mise en pratique	<p>Configuration de téléphones SIP depuis un navigateur : choix du codec, renseignement de l'adresse SIP, indication d'un serveur DNS, ...</p> <p>Installation et configuration d'un registrar Asterisk.</p> <p>Création des comptes des téléphones et du dialplan (plan de numérotation).</p>
Les enregistrements SRV	<p>Communication entre des réseaux différents.</p> <p>Utilisation des serveurs DNS.</p> <p>Travaux pratiques : Création de deux domaines DNS, mise à jour des enregistrements SRV pour la localisation des services sip/udp.</p> <p>Configuration des softphones, et tests.</p> <p>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.</p> <p>Notions de contexte, extensions, priorités, et applications.</p> <p>Configuration d'appels stateful et stateless.</p>
La sécurité	<p>Critères de sécurité : Disponibilité, Intégrité, Confidentialité.</p> <p>Mise en évidence des risques pour chaque critère.</p> <p>Contraintes légales concernant les appels d'urgence.</p> <p>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.</p> <p>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.</p> <p>Avantages : gestion d'un nombre important de téléphones (logiciels ou matériels).</p> <p>Principe : distribution d'adresses IP dynamiques par un serveur BOOTP/DHCP, mise à disposition des fichiers de configuration pour chaque téléphone.</p> <p>Travaux pratiques : configuration du provisioning pour des terminaux SIP (Linksys de Cisco).</p>

Annuaire LDAP

Durée: 2 jours
1020 €

10 au 11 mars
16 au 17 juin

29 au 30 septembre
9 au 10 novembre

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 ldif.
Méthodes et commandes de consultation dans un annuaire.
Travaux pratiques :
recherche composée dans un annuaire.
Consultation depuis un client de messagerie.
Gestion des permissions.
Outils graphiques de consultation.

Sécurité

Authentification, contrôle d'accès, chiffrement des transactions
Annuaire et PKI

Architecture

Distribution, réplication d'annuaires.
Le besoin de synchronisation, les méthodes.
Travaux pratiques :
création d'un serveur esclave, avec réplication de toutes les
informations du serveur maître.

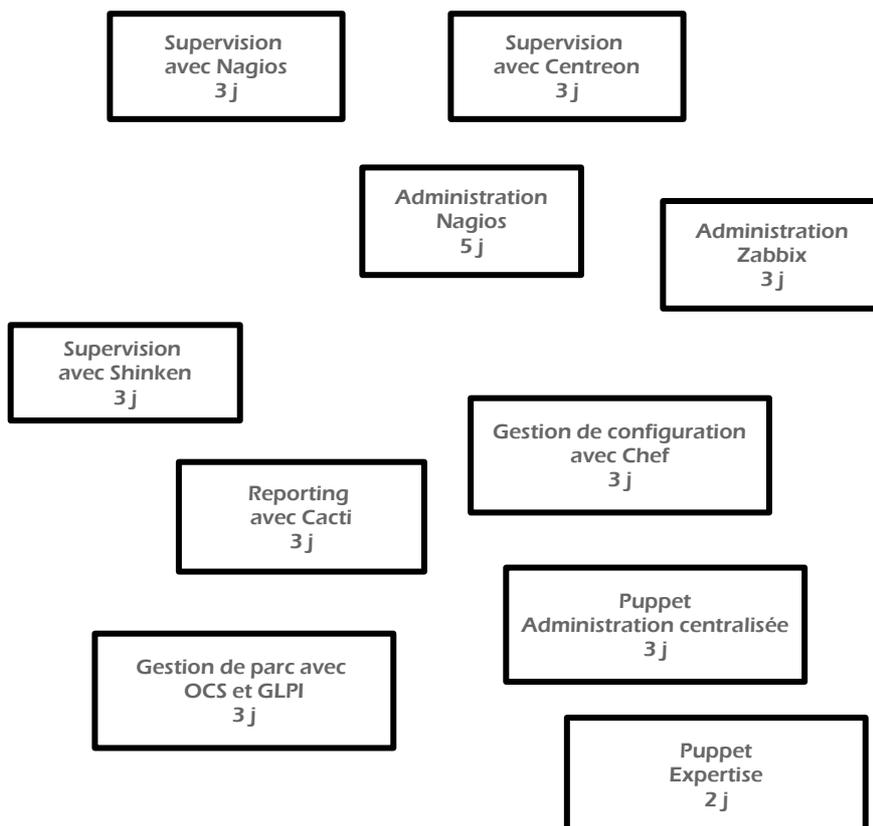
Intégration

Dans le réseau d'entreprise :
Exemples : pam/ldap, samba/ldap.
Travaux pratiques :
création d'un annuaire ldap pour samba,
configuration d'un module pam-ldap,
Mise en oeuvre pour un serveur de messagerie.

Le marché

Présentation des principaux annuaires.

Filière Production et supervision



Supervision nagios : utilisation

Durée: 3 jours
1505 €

14 au 16 mars

23 au 25 mai

8 au 10 juin

12 au 14 septembre

14 au 16 novembre

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.

Supervision nagios : utilisation

Déploiement	Sur les hôtes, principes de NRPE, NSCA. Travaux pratiques : écriture de scripts de déploiement. (NRPE)
Les plugins	Principe de fonctionnement. Quelques plugins courants
L'interface Centreon	Les fonctionnalités, les sites de référence, L'architecture Nagios/Centreon. 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
2400 €

14 au 18 mars
23 au 27 mai

12 au 16 septembre
14 au 18 novembre

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'ordonnancement. Délai entre chaque test. Entrelacement des services. Tests concurrents. Fréquence de récupération.</p>
Contrôle et débogage	<p>Analyse des fichiers de logs. Commandes de contrôle. Mode d'exécution des plugins. Options détaillées.</p>
Les plugins	<p>Principe de fonctionnement. Mise en 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>

Administration Nagios

Centralisation NDO

Fonctionnalités et composants.
Travaux pratiques :
Mise en oeuvre de NDO

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
1505 €

2 au 4 mars
9 au 11 mai

26 au 28 septembre
7 au 9 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
1505 €

21 au 23 mars
18 au 20 mai

19 au 21 septembre
21 au 23 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.

Supervision avec Centreon

Durée: 3 jours
1505 €

7 au 9 mars
17 au 19 mai

3 au 5 octobre
28 au 30 novembre

Public:

Exploitants et utilisateurs d'un système de supervision Centreon.

Objectifs:

Connaître les fonctionnalités de Centreon, 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 Centreon	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. Comparaison avec Nagios. Architecture : Centreon Engine et Centreon Broker.
Installation et configuration	Configuration des hôtes, services, commandes, périodes temporelles, contacts, groupes, catégories, modèles Travaux pratiques : exemples de supervision de services
Configuration avancée	macros, méta-services, escalades, traps SNMP Le collecteur de supervision L'ordonnanceur, Centreon Broker, Centreontrapd
Extensions	Présentation des produits complémentaires

Zabbix administration

Durée: 3 jours
1505 €

2 au 4 mars
9 au 11 mai

26 au 28 septembre
7 au 9 novembre

Public:

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

Objectifs:

Connaître les fonctionnalités de Zabbix. Savoir installer, configurer et administrer le produit. Savoir mettre en oeuvre la supervision dans un environnement hétérogène.

Connaissances préalables nécessaires:

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

Programme:

Présentation Zabbix

Historique du produit, version, licence.
Systèmes supportés.

Les fonctionnalités de Zabbix :

Supervision réseau et serveurs

(état des services, charge processeur, disques, ...)

système de configuration d'alertes,

interface de supervision,

reporting et visualisation des données collectées par zabbix,

gestion des ressources (capacity planning)

Architecture

Quelques définitions :

notion d'hôte, de groupe d'hôtes, item, trigger,

event, action, escalation, media, notification, template

Les composants de l'architecture :

Zabbix server, agent, proxy

Java gateway pour la supervision JMX.

Installation

Configuration requise. Plate-formes supportées.

Site de référence.

Travaux pratiques :

Installation depuis les packages,

Initialisation de la base de données.

Démarrage du serveur Zabbix.

Configuration depuis l'interface PHP

Configuration

Gestion des hôtes et groupes d'hôtes,
des items, triggers, event.

Gestion des notifications sur événements.

Visualisation.

Création de templates

Les utilisateurs : configuration, groupes d'utilisateurs,
droits d'accès

Zabbix administration

Applications types

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

Travaux pratiques :
utilisation de l'interface web

Supervision distribuée

Principe des proxy Zabbix.
Mise en oeuvre d'une architecture distribuée.

Gestion de configuration avec Chef

Durée: 3 jours

1505 € HT

du 8 au 10 février

du 2 au 4 mai

du 12 au 14 septembre

du 14 au 16 novembre

Public:

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

Objectifs:

Comprendre le fonctionnement de Chef, 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/Linux et un langage de développement de scripts.

Programme:

Introduction

Présentation de Chef, fonctionnalités
Gestion automatisée de l'infrastructure systèmes et réseaux (serveurs, machines virtuelles, containers, équipements réseaux)

Architecture

Chef server, Chef Analytics pour le suivi,
Chef management Console : interface web d'administration
Chef-client sur les noeuds
clients d'administration (workstation) et Chef Development Kit.
Définition de la notion de noeud, et des attributs du noeud, des cookbooks

Installation et configuration

Chef Server :prérequis techniques,
différents modes d'installation (standalone, cluster, ..)
Clients d'administration (workstation):
utilisation de knife pour synchroniser les données avec chef-server.
Installation de chef-client sur un noeud,
Etapes de l'exécution d'un client :
Récupération des données sur les noeuds, authentification auprès du Chef-Server
Création de la "run-list", exécution, mise à jour du noeud.

Cookbooks

Principe. Les cookbooks disponibles en opensource
Exemples : apache2 et nginx pour configurer un serveur apache et nginx,
chef-client, pour gérer le fichier de configuration clien.rb et chef-client service
Modification, développement de cookbook.

Gestion de configuration avec Chef

Chef Analytics et la console

Installation et configuration.

Principe de Chef Analytics.

Collecte de données, sur les noeuds, les actions exécutées,...

Visualisation des données

Console web d'administration des clients,
cookbooks, noeuds, rapports, rôles, etc..

Puppet : administration centralisée

Durée: 3 jours
1505 €

du 21 au 23 mars
du 6 au 8 juin

du 26 au 28 septembre
du 12 au 14 décembre

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. Ce stage est illustré par de nombreux travaux pratiques sur le développement des manifests puppet.

Connaissances préalables nécessaires:

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

Programme:

Introduction	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.
Architecture	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.
Installation et configuration	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.

Puppet : administration centralisée

Le langage puppet

Introduction : présentation des manifests, modules, templates.
Les manifests. Dépendance entre instructions.
Variables, facts.
Déclarations conditionnelles : if, case, selectors
Les classes et modules
Installation de modules
Les templates.
Paramètres de classes.
Types de ressources définis.
Mise en oeuvre sur de nombreux travaux pratiques.
Utilisation de la documentation.
Organisation du site .pp.
Particularités windows.

Bonnes pratiques

Retours d'expériences et méthodes d'organisation et de développement des scripts puppet.
Présentation des patterns pour puppet

Puppet : expertise

Durée: 2 jours
1030 €

du 24 au 25 mars
du 9 au 10 juin

du 29 au 30 septembre
du 15 au 16 décembre

Public:

Administrateurs, exploitants souhaitant approfondir leurs connaissances de Puppet pour la gestion centralisée des configurations logicielles.

Objectifs:

Comprendre le fonctionnement de la base hiera, savoir rechercher et créer des modules et connaître les outils complémentaires disponibles autour de puppet.

Connaissances préalables nécessaires:

Il est indispensable de connaître les bases de puppet.

Programme:

La base hiera

Intérêt de Hiera :
gestion des paramètres en dehors des manifests.
Fonctionnement, mise en oeuvre .
configuration avec le fichier hiera.yaml
Préparation de la hiérarchie
Utilisation des données hiera depuis puppet.

Les outils complémentaires

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.

Les modules

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.
Exemple : lien avec Nagios par le module thias/nagios

Gestion de Parc avec OCS et GLPI

Durée: 3 jours

1510 € HT

23 au 25 février

18 au 20 mai

31 août au 2 septembre

23 au 25 novembre

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.

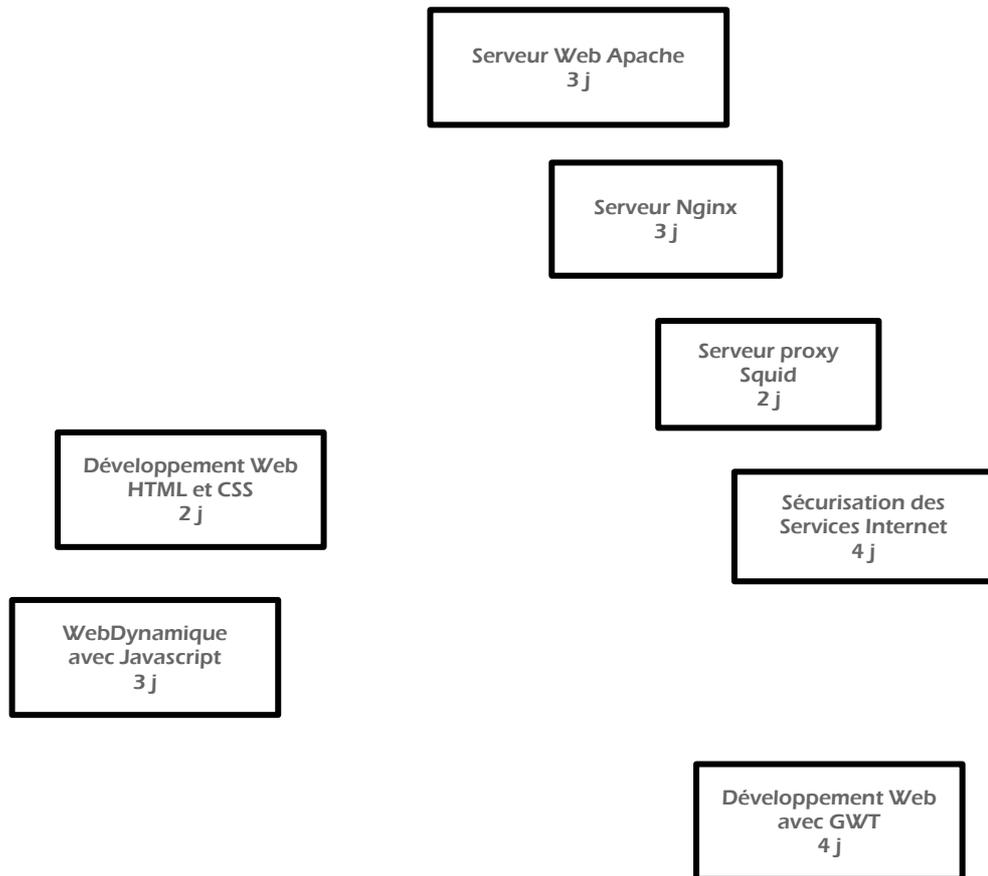
Programme:

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

Filières Internet et Web



Serveur WEB : apache

Durée: 3 jours
1505 €

8 au 10 février
11 au 13 avril
20 au 22 juin

26 au 28 septembre
2 au 4 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
1505 €

7 au 9 mars
13 au 15 juin

10 au 12 octobre
19 au 21 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 <code>/etc/nginx/nginx.conf</code> : 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.

Administration serveur Nginx

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.

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
2025 €

29 février au 3 mars
17 au 20 mai

5 au 8 septembre
7 au 10 novembre

Public:

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

Objectifs:

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

Connaissances préalables nécessaires:

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

Programme:

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

Sécurisation des services internet

SSH et GPG	<p>SSH : Secure SHell Présentation du protocole de connexion SSH Utilisation de la cryptographie. Authentification par clé. Travaux pratiques : configuration, sécurisation de l'ouverture du port. GPG : GNU Privacy Guard Transmission de messages chiffrés et signés.</p>
VPN et tunnels	<p>Objectifs. Fonctionnement. Mise en oeuvre : tunnels ssh, stunnel. Travaux pratiques avec openVPN. Principe d'IPsec. Implémentation avec FreeS/WAN.</p>
Proxy	<p>Introduction : la fonction d'un serveur proxy. Les multi-serveurs proxys. Mise en 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.</p>
Services reseaux	<p>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.</p>
Messagerie	<p>Sécurisation. Notion de relais ouvert. Outils de test du serveur. Travaux pratiques : mise en oeuvre avec postfix. Apport de Cyrus.</p>
Serveur http	<p>Restriction d'accès. Authentification http. Sécurité avec SSL et HTTPS : principe, configuration par défaut. Génération de clés SSL</p>
Acces reseaux	<p>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.</p>

Sécurisation des services internet

Détection de failles

Outils : snort, openvas.

Travaux pratiques :

mise en oeuvre et réalisation d'un audit de sécurité de l'environnement de test.

Développement Web HTML et CSS

Durée: 2 jours
1055 €

29 février au 1er mars
9 au 10 mai

5 au 6 septembre
14 et 15 novembre

Public:

Utilisateurs de l'informatique, contributeurs, informaticiens (chefs de projet, concepteurs, réalisateurs), webmestres amenés à participer à l'élaboration de documents en vue d'une publication sur le Web

Objectifs:

Etre capable de développer des documents HTML. Explorer l'ensemble des possibilités offertes par HTML. Améliorer la présentation et la maintenance des pages web d'un site grâce aux feuilles de style.

Connaissances préalables nécessaires:

Connaissances générales sur Internet.

Programme:

Introduction	<ul style="list-style-type: none"> Qu'est-ce que le web ? Les langages et leur utilité Les navigateurs et leurs outils Caractéristiques du HTML : langage de balises Les nouveautés du HTML5
Le document HTML5 minimum	<ul style="list-style-type: none"> Le doctype La balise HTML L'en-tête du document Le titre du document L'encodage (charset) du document Le corps du document Valider le code HTML5
Gestion du texte	<ul style="list-style-type: none"> Les balises du texte Les commentaires Les caractères spéciaux Formatage du texte
Structuration du document	<ul style="list-style-type: none"> Les titres Les paragraphes Les citations Les séparateurs horizontaux Les listes Les divisions de page

Développement Web HTML et CSS

Les liens	Concepts Liens internes Liens externes Infobulle Nouveautés HTML5
Gestion des tableaux	Créer un tableau Gérer l'alignement du tableau, des cellules du tableau Fusionner des cellules, des colonnes Formater et structurer le tableau
Les balises démantiquées et d'organisation	Concepts Les balises header, nav, footer et aside Les balises section et article Nouveautés HTML5
Les formulaires	La déclaration de formulaire Ligne et zone de texte La liste déroulante Les boutons de choix unique (radio), de choix multiples (checkbox) Les boutons d'envoi, d'annulation, de commande Formulaires cachés, transfert de fichiers, formulaires de mot de passe, organisation des champs de formulaires Gestion des formats Les curseurs
Gestion du multimédia	Les images et arrière-plans L'insertion de fichiers audio et video
Introduction aux feuilles de style	Concepts et utilité des feuilles de style Notions de base des CSS Gestion des polices de caractères, de texte Les listes et tableaux Les arrière-plans Propriété de l'impression Autres feuilles de style CSS3

Web Dynamique avec JavaScript

Durée: 3 jours
1505 €

2 au 4 mars
11 au 13 mai

7 au 9 septembre
16 au 18 novembre

Public:

Développeurs web, architectes web, chefs de projets, webmestres

Objectifs:

Maîtriser l'utilisation des balises et feuilles de style HTML5/CSS3. Utiliser JavaScript pour gérer l'interactivité d'un site web. Savoir construire son site en Responsive Design

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:

Rappels HTML	Création d'une page web Document HTML minimum Modèles de sites HTML5
Feuilles de style CSS3	Les nouveaux sélecteurs CSS3 Les pseudo-classes
La mise en forme CSS3 des boîtes	Les nouvelles propriétés CSS3 Gérer le débordement, l'arrière-plan, les bordures Ombres et dégradés Générateurs CSS3 en ligne
Gestion du texte	Des nouvelles mises en forme de texte Gestion des polices La typographie avancée
Javascript	Introduction Structure du langage Variables et affectation Les types de variables Les opérateurs Les structures de contrôle Javascript et les formulaires Javascript et le CSS
Gestion des formulaires	Rappels Les nouveaux champs La validation des formulaires Les expressions régulières L'aide Des pseudo-classes pour les formulaires La validation des saisies Gestion des boutons

Web Dynamique avec JavaScript

Les transformations CSS3	La transformation, déplacement, mise à l'échelle, rotation, déformation Appliquer toutes les transformations Les générateurs en ligne
Les transitions CSS3	Mettre en place les transitions Effectuer une transition de déplacement Les autres propriétés des transitions Les générateurs en ligne
Les animations CSS3	La construction d'une animation Exemples d'animation : déplacement, rotation, transparence
Le multimédia	Balises et codecs audio et video
Le canevas de dessin	La création bitmap en javascript Les différentes techniques Les transformations La manipulation des objets
Le design Web adaptatif (Responsive Design)	Les requêtes de média La taille des écrans Les grilles fluides Des images adaptatives Créer du design adaptatif

Développement Web avec GWT

Durée: 4 jours
2140 €

15 au 18 février
23 au 26 mai

19 au 22 septembre
21 au 24 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.

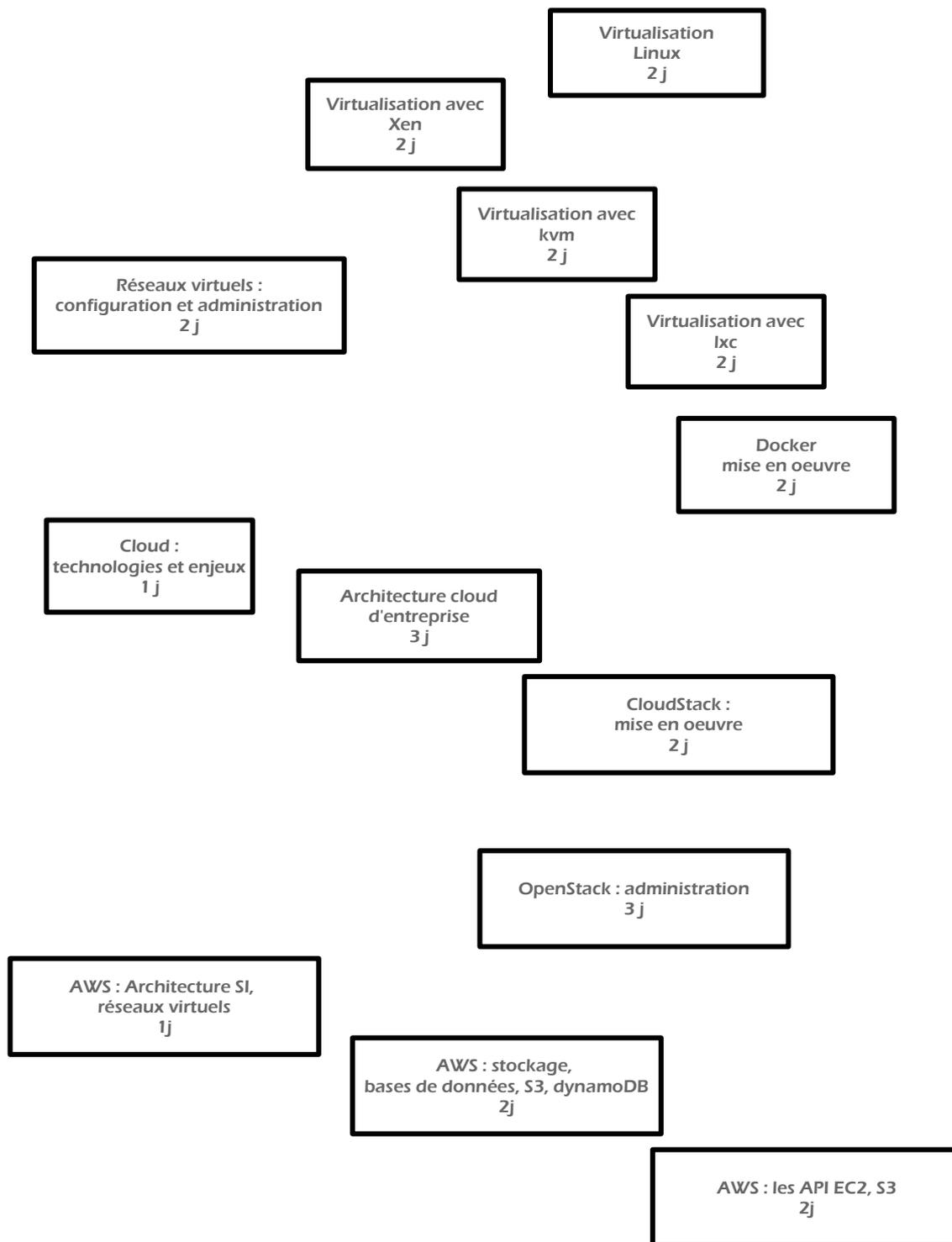
Connaissances préalables nécessaires:

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

Filières Virtualisation et Cloud



Virtualisation Linux

Durée: 2 jours

Prix et dates: nous consulter

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.
lxc	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

Prix et dates: nous consulter

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.

Virtualisation avec Xen

Supports de stockage	Déclaration des espaces de stockage accessibles au domU : périphériques blocs, partitions physiques, ou volumes logiques.
Systèmes 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
1115 €

18 au 19 février
12 au 13 mai

22 au 23 septembre
24 au 25 novembre

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>

Virtualisation avec KVM

Migration d'images

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

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 switches VDE.
Configuration des switches avec Unixterm.

Virtualisation avec lxc

Durée: 2 jours
1115 €

24 au 25 mars
26 au 27 mai

8 au 9 septembre
7 au 8 novembre

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	<p>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.</p>
Cgroup	<p>Fonctionnement de Control Group. Vérification de la configuration du noyau. Activation des Cgroups.</p>
Les outils LXC	<p>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...</p>
Gestion des conteneurs	<p>Configuration, création, démarrage. Utilisation des templates pour créer des conteneurs standards. Choix des systèmes de fichiers.</p>
Configuration du réseau	<p>Les différentes méthodes : interface physique, pont/commutateur virtuel, vlan. Mise en oeuvre. Configuration d'un point par brctl.</p>

Docker : mise en oeuvre

Durée: 2 jours

1030 € HT

du 18 au 19 février

du 2 au 3 juin

du 20 au 21 octobre

du 1er au 2 décembre

Public:

Administrateurs, chefs de projet et toute personne souhaitant mettre en oeuvre Docker pour déployer ses applications.

Objectifs:

Comprendre et savoir mettre en oeuvre Docker, et les produits associés.

Connaissances préalables nécessaires:

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

Programme:

Introduction

Présentation docker, principe, fonctionnalités

Besoins : packaging d'applications, déploiement rapides, coexistence de plusieurs versions d'une application sur un même serveur.

Principe et architecture

Les containers lxc.

Présentation de lxc : Linux containers, historique, principe de fonctionnement.

Les Cgroups.

L'isolation de ressources, création d'un environnement utilisateur.

Positionnement par rapport aux autres solutions de virtualisation.

Apports de Docker : Docker Engine pour créer et gérer des containers Dockers.

Plate-formes supportées.

Produits complémentaires : Docker Machine, Docker Compose, Kitematic, Docker Swarm, Docker Registry

Installation et configuration

Prérequis techniques et travaux pratiques d'installation sur Linux
Création d'un groupe Docker.

Gestion des images et des containers

Utilisation de DockerFile pour créer de images :

principales instructions (RUN, FROM, ENV, EXPOSE, etc ...)

Recommandations et bonne pratiques d'écriture de DockerFile.

Gestion des containers : création, affichage, sauvegarde de l'état

Volumes de données

Initialisation des volumes de données lors de la création d'un container.

Ajout de volumes de données, contrôle avec la commande Docker inspect

Sauvegarde, migration, restauration de volumes

Création de containers de volume de données

Docker : mise en oeuvre

Administration

Applications multi-containers avec Compose
Présentation de Swarm pour le clustering
Configuration réseau et sécurité dans Docker.

Réseaux virtuels: administration

Durée: 2 jours

1090 € HT

10 au 11 mars

2 au 3 juin

6 au 7 octobre

1er au 2 décembre

Public:

Administrateurs responsables d'un parc de machines virtuelles.

Objectifs:

Connaître les techniques et outils de configuration et administration de réseaux virtuels.

Connaissances préalables nécessaires:

Connaissance de l'administration système et des réseaux.

Programme:

Introduction

Le besoin de réseaux virtuels :
multiples machines virtuelles sur un seul hôte, cloud
Outils et leurs caractéristiques : VDE, Switch Light, OpenVswitch

Réseaux virtuels et clouds

Exemples dans OpenStack, et Amazon AWS.
Création d'interface réseaux, configuration du routage,
mise en place des services réseaux (DNS, DHCP, ...)

Open vSwitch

Présentation : fonctionnalités (commutateur virtuel, sécurité, QoS, ..)
architecture, protocoles supportés.
Installation : packages et partie noyau.
Configuration par ovs-vsctl.

Administration réseaux virtuels

Travaux pratiques avec Open vSwitch
Création des interfaces réseaux.
Activation des interfaces et du routage.
Supervision réseaux, analyse des flux.
Configuration de la QoS pour limiter les flux.

Cloud : technologies et enjeux

Durée: 1 jour
555 € HT

14 mars
17 mai

12 septembre
14 novembre

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
1650 €

15 au 17 mars
18 au 20 mai

13 au 15 septembre
15 au 17 novembre

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. Les avantages d'une solution de cloud privé : sécurité, localisation des données. Solutions disponibles : un environnement privé mis à disposition par un fournisseur de cloud standard, des produits permettant d'installer un cloud interne à l'entreprise. Exemples de produits de cloud privé : Eucalyptus, Xen Cloud Platform, VMWare VSphere, Enomaly, cloudstack, openstack, openNebula 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. Présentation. Architecture des produits. Travaux pratiques : installation depuis les packages RPM. Configuration de l'hyperviseur (xen ou kvm). Lancement d'une première machine virtuelle.</p>
configuration d'un cloud privé	<p>Mise en oeuvre du réseau entre machines virtuelles. Les différents modes de gestion du réseau. Outils d'administration : interface web d'administration, ligne de commande (gestion des noeuds, des utilisateurs...) Lien avec un cloud public comme Amazon.</p>
La sécurité	<p>Evaluation des risques, présentation des outils et solutions Aspects juridiques, protection des données, traçabilité, chiffrement, génération de clés...</p>

CloudStack : mise en oeuvre

Durée: 2 jours
1115 €

du 10 au 11 mars
du 3 au 4 mai

du 29 au 30 août
du 3 au 4 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.

CloudStack : mise en oeuvre

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
1695 €

3 au 6 février

11 au 13 avril

27 au 29 juin

19 au 21 septembre

5 au 7 décembre

Public:

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

Objectifs:

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

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, Neutron, Keystone, Horizon L'accès par des clients webservices 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. Déploiement en masse. Scripts d'automatisation.
Extension	Ajout d'un noeud de calcul. Configuration. Visualisation de la capacité supplémentaire. Gestion. Migration d'un noeud de calcul. Passer à l'échelle. Les mises à jour. Analyser les problèmes.

OpenStack : configuration et administration

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.

Stratégie pour le stockage.

Gestion du réseau

Définitions : adresses mobiles, adresses fixes.

Les modes réseaux : Flat, FlatDHCP, VLAN.

Neutron : fonctionnalités, architecture. Mise en oeuvre de Neutron.

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

Création de stocks d'adresses IP mobiles. Affectation.

Scripts SDN (Software Defined Network).

Définition d'une architecture globale.

AWS : architecture SI, réseaux virtuels

Durée: 1 jour
540 €

15 février
25 avril

29 août
17 octobre

Public:

Architecte, chef de projet, et toute personne souhaitant connaître les possibilités du cloud Amazon.

Objectifs:

concevoir une architecture applicative avec AWS.

Connaissances préalables nécessaires:

Connaissance générale des systèmes d'informations et de la virtualisation.

Programme:

Introduction	Présentation du système AWS. Positionnement par rapport aux autres offres de cloud. Etude des fonctionnalités accessibles avec AWS Management Console: Ressources de calcul et réseaux Stockage Bases de données Déploiement et supervision Services applicatifs, services pour mobiles, ..
Stockage	Présentation des différentes options de stockage : RedShift, S3, dynamoDB Mise en oeuvre de Amazon Simple Storage Service (S3)
Calcul et réseaux	Utilisation d'Amazon EC2 (Elastic Compute Cloud) et Amazon Elastic Block Storage (EBS) Création d'un Amazon Virtual Private Cloud (VPC)
Sécurité	Authentification et autorisation dans le cloud. Présentation AWS Identity et Access Management

AWS : stockage,S3,dynamodb..

Durée:2 jours
1090 €

16 au 17 février
26 au 27 avril

30 au 31 août
18 au 19 octobre

Public:

Architecte, chef de projet, et toute personne souhaitant connaître les solutions de stockage et de gestion des données avec AWS.

Objectifs:

Comprendre les différentes solutions de stockage et des gestion des données et savoir les mettre en oeuvre.

Connaissances préalables nécessaires:

Connaissance des fonctionnalités de base d'AWS, et notions de bases de données classiques.

Programme:

Introduction	Caractéristiques et rôle de chaque composant : Stockage avec CloudFront, Glacier, S3, Storage Gateway Gestion de données avec DynamoDB, ElastiCache, RDS, Redshift
Mise en oeuvre S3	Création d'espaces de stockage distants. définition des droits d'accès, de règles de cycles de vie (destruction, sauvegarde et archivage avec Glacier, ...) Upload de données, ..
Bases de données	DynamoDB : fonctionnalités (NoSQL, architecture distribuée, réplication, ... Exemples de mise en oeuvre de dynamoDB : création de tables, exportation / importation de données, outils de supervision. utilisation d'IAM pour le contrôle d'accès. Intégration avec Redshift Redshift : principe des clusters Redshift, définition de snapshots, gestion de la sécurité, groupes, ElastiCache : zone de cache évolutive et distribuée. Terminologie : cache node, cluster, etc ... Mise en oeuvre d'ElastiCache dans un environnement multi-noeuds. Paramétrage des droits d'accès avec IAM Configuration des clients.

AWS : les API EC2 et S3

Durée: 2 jours
1090 €

18 au 19 février
28 au 29 avril

1^{er} au 2 septembre
20 au 21 octobre

Public:

Architecte, chef de projet, développeur souhaitant connaître les possibilités d'interopérabilité offertes par les API EC2 et S3

Objectifs:

Savoir mettre en oeuvre les API EC2 et S3 et mettre en évidence la compatibilité des scripts et outils.

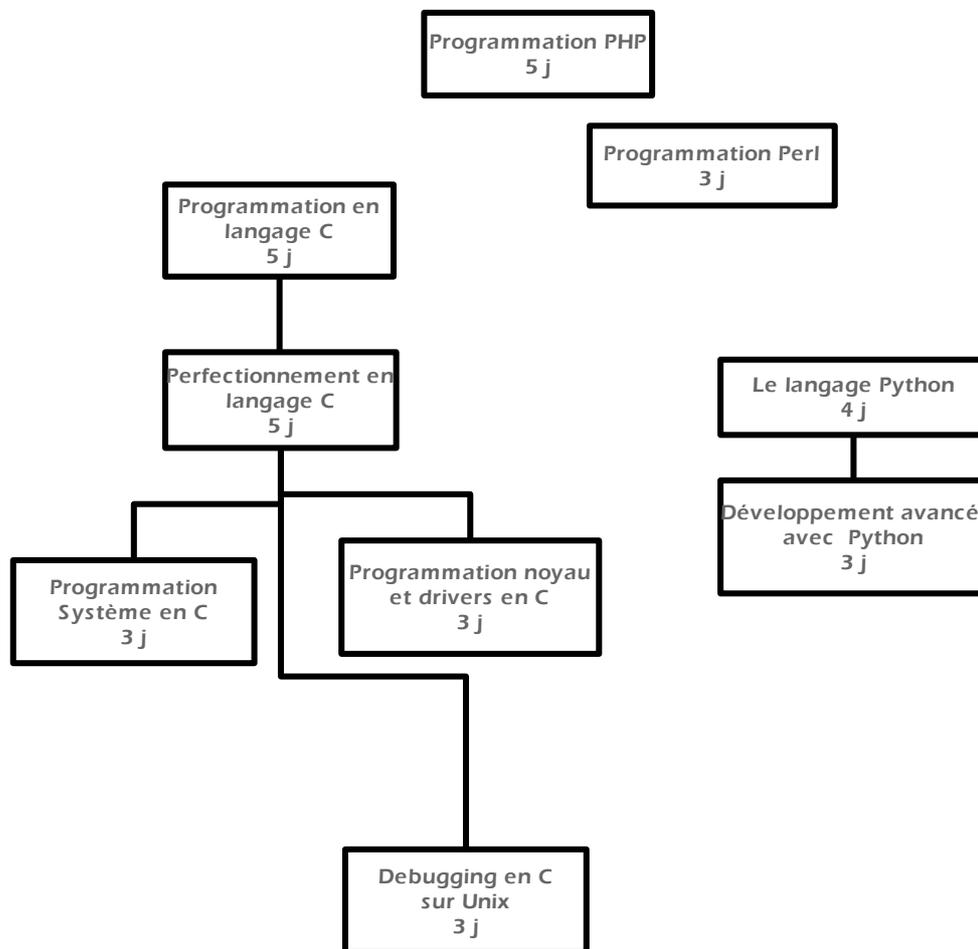
Connaissances préalables nécessaires:

Connaissance des fonctionnalités d'AWS, des notions de virtualisation et de cloud, et d'un langage de script.

Programme:

Introduction	Principe de la comptabilité EC2 et S3. Le positionnement des grands acteurs du cloud.
Exemples de scripts	Procédures automatisées de gestion de serveurs de calcul ou d'espaces de stockage. Méthodes pour créer les outils compatibles EC2 et S3. Mise en évidence de la compatibilité sur des environnements OpenStack et AWS.
Les kits SDK AWS	Besoin : accès aux services AWS depuis des applications ou plate-formes clientes. Différentes plate-formes supportées : Android, Java, PHP, python, ruby, Node.js, .net Exemples : portage des scripts de procédures automatisées sur android, interaction avec les ressources dans un cloud OpenStack, et AWS.
L'API EC2	Elastic Compute Cloud. Fonctions disponibles : gestion des instances, des zones géographiques, des tables de routage, ..
L'API S3	Amazon Simple Storage Service Fonctions disponibles avec l'API Amazon S3 REST

Filières Développement



Programmation en langage C

Durée: 5 jours

2300 € HT

1er au 5 février

23 au 27 mai

19 au 23 septembre

14 au 18 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 en langage C

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.
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

2350 € HT

8 au 12 février

30 mai au 3 juin

26 au 30 septembre

21 au 25 novembre

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
1540 €

15 au 17 février
6 au 8 juin

3 au 5 octobre
5 au 7 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.

Programmation système en C sur Unix/Linux

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
1610 €

29 février au 2 mars
13 au 15 juin

10 au 12 octobre
12 au 14 décembre

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

Développement d'applications Web avec PHP

Durée: 5 jours
2350 €

8 au 12 février

4 au 8 avril

13 au 17 juin

19 au 23 septembre

21 au 25 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. SOLLite et MySQL. Connexion et déconnexion. Lire et mettre à jour les données. Gérer les erreurs. Requêtes préparées.

Développement d'applications Web avec PHP

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.

PHP et le système de fichiers.

La sécurité sur les fichiers.

Les fonctions de manipulation d'images.

Création de graphiques (JPGRAPH, ARTICHOV).

Génération de fichiers Excel.

Les concepts de sécurité, l'authentification.

Programmation Perl

Durée: 3 jours
1390 €

15 au 17 février
17 au 19 mai

5 au 7 septembre
7 au 9 novembre

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

Programmation Perl

Fichiers

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

Exécution

différentes méthodes d'exécution
compilation , exécution

Le langage Python

Durée: 4 jours
2035 €

7 au 10 mars
23 au 26 juin

19 au 22 septembre
5 au 8 décembre

Public:

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

Objectifs:

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

Connaissances préalables nécessaires:

La connaissance d'un langage de programmation sera appréciée.

Programme:

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

Le langage Python

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

Développement avancé avec Python

Durée:3 jours

Prix et dates: nous consulter

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.

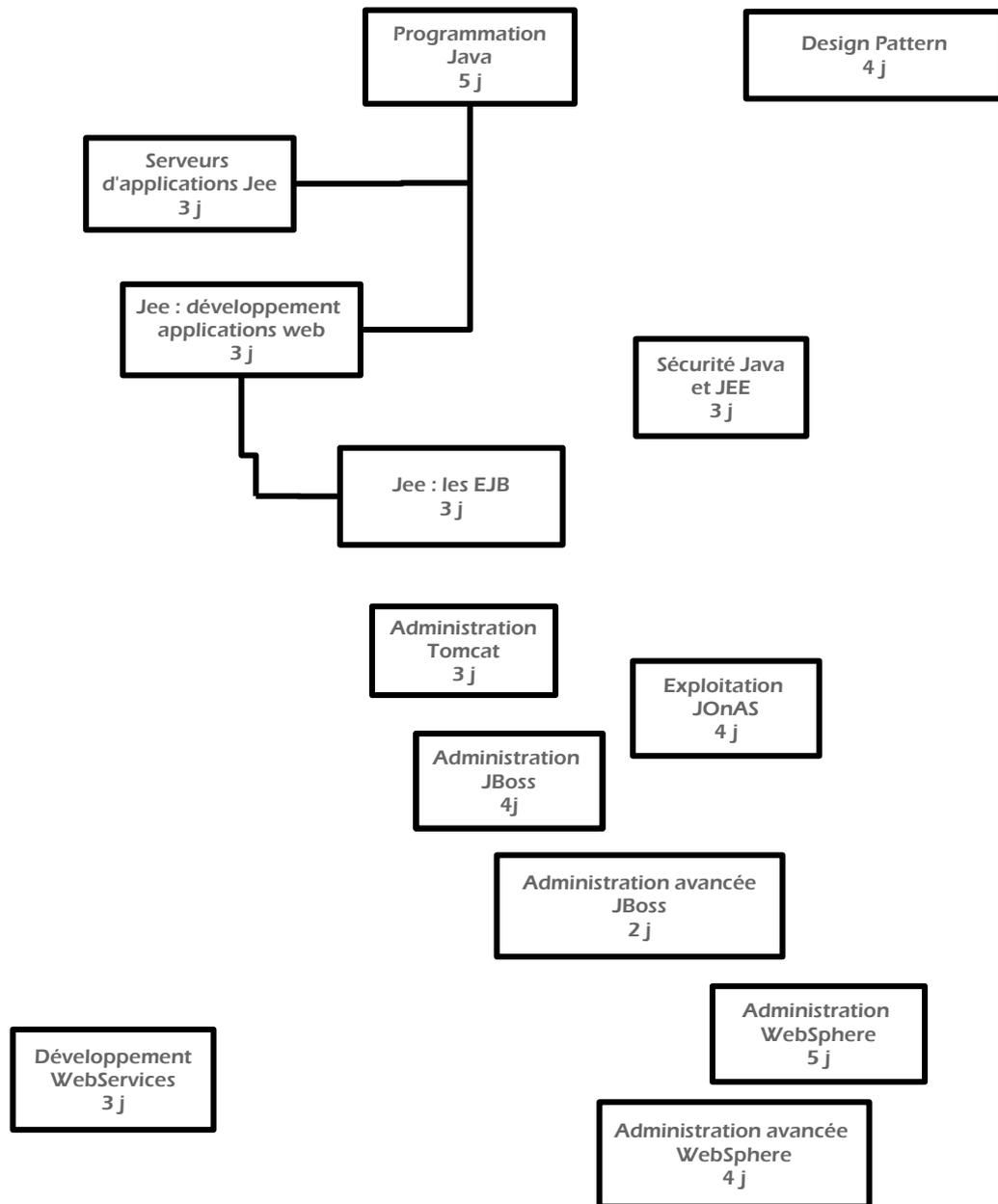
Connaissances préalables nécessaires:

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 Gadfly, interfaçage MySQL avec MySQLdb, accès à postgresSQL 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, ...

Filières Langages Java et JEE



Concepts Objets et programmation Java

Durée: 5 jours
2350 €

1^{er} au 5 février

4 au 8 avril

6 au 10 juin

5 au 9 septembre

14 au 18 novembre

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.

Concepts Objets et programmation Java

Les structures de données	La classe Vector. La classe Stack. L'interface Enumeration. Structures de données ordonnées. Les collections.
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.

Sécurité Java et JEE

Durée: 3 jours
1555 €

29 février au 2 mars
17 au 19 mai

21 au 23 septembre
2 au 4 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 : 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 fabrique de certificats Java. Les certificats de modification X509.
Contrôle	Rappel sur les ACL. Le paquetage java.security.acl. Ajout d'entrée, vérification d'accès.
Obfuscation	Principe Techniques d'obfuscation Solutions commerciales
JAAS	Présentation Fonctionnement et mise en 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
1505 €

7 au 9 mars
20 au 22 juin

3 au 5 octobre
12 au 14 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 (le contrôleur)	Servlet, JSP et Taglib. Design Pattern MVC 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

Prix et dates: nous consulter

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.

JEE : les EJB

Gestion de la sécurité

Sécurité Jee avec JAAS.
Sécurité par programmation.
Sécurité déclarative.

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
1670 €

15 au 17 février
2 au 4 mai

12 au 14 septembre
7 au 9 novembre

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. Maintenance de session avec la méthode setMaintainSession() Gestion des attachements. Sérialisation personnalisée avec typeMapping.

Développement Web Services

Axis 2	Présentation, historique Fonctionnalités. Travaux pratiques : installation, lancement du serveur, tests.
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.

Serveurs d'application JEE

Durée: 2 jours

Prix et dates: nous consulter

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.

Serveurs d'application JEE

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.

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.

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 Tomcat

Durée: 3 jours
1505 €

9 au 10 février

18 au 20 avril

27 au 29 juin

17 au 19 octobre

5 au 7 décembre

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. Ce module s'appuie sur des travaux pratiques.

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.

Administration Tomcat

Console d'administration	<p>Travaux pratiques : installation et configuration des droits d'accès dans le fichier tomcat-users.xml</p> <p>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.</p>
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>

Administration Tomcat

Journalisation

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.

Exploitation

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

Administration JBoss

Durée: 3 jours
1590 €

1er au 3 février
9 au 11 mai

5 au 7 septembre
14 au 16 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

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.

Applicatif

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.

Lien avec Tomcat

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.

Gestion des ressources

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.

Administration JBoss

Sécurité avec JAAS et SecurityManager

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.

Automatisation

Exploitation par scripts avec twiddle, JBoss-cli. Parcours de ressources.
Ecriture d'un script de visualisation de statistiques sur un cluster: taux d'utilisation mémoire, nombres de requêtes, connexions JDBC.
Scripts d'automatisation.

Intégration

Clustering JBoss : définitions, mise en oeuvre.
Configuration de JBoss en multi-instances.
Accès depuis un serveur apache, mise en place d'une liaison ajp à l'aide de mod_jk, mod_cluster.
Réplication sous JBoss.
Sessions Http distribuées. Réplication d'EJB.
Ecriture d'un service JBoss permettant de simuler une application batch.

Journalisation

Mise en oeuvre de la journalisation.
Analyse des logs
Fichiers de logs, résolution d'incidents.
Configuration de la journalisation. Modification des niveaux de journalisation.
Différents types ventilations. Centralisation des logs vers une machine syslog externe.

Suivi

Gestion des performances. Suivi du garbage collector.
Installation du MBean Jboss Profiler.
Suivi du ramasse miettes (garbage collector) avec jconsole.
Paramètres de la JVM pour gérer au mieux l'espace mémoire.
Travaux pratiques: mise sous stress avec jmeter, suivi de la charge et amélioration de la configuration. Optimisation et allègement des configurations JBoss.
Utilisation du MBean SNMP sur version < 7.
Envoi de trap SNMP à une console de supervision.
Lien vers la supervision avec Nagios via SNMP.
Automatisation d'alertes en cas d'incident.

Administration avancée JBoss

Durée: 2 jours
1090 €

4 au 5 février
12 au 13 mai

8 au 9 septembre
17 au 18 novembre

Public:

Administrateurs d'applications JEE souhaitant configurer, optimiser des applicatifs JBoss et savoir mettre en oeuvre le clustering..

Objectifs:

Savoir optimiser la configuration d'applicatifs fonctionnant avec JBoss. 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, Installation, configuration.
Cluster	Définitions : noeud, serveur, application. Architecture HA JBoss. Fonctionnement en cluster ou en grappes de serveurs. Mode domain, mode standalone. Travaux pratiques : Configuration d'un cluster, configuration du réseau, fichiers de configuration. standalone-ha.xml, domain.xml.
JGroups	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.
Applications	Principe des applications réparties. Mise à jour des applications. Réplication de sessions HTTP. Mise en place de pound, mod_jk, haproxy et mod_cluster. Réplication d'EJB. Le service de déploiement. Gestion des caches.
Architecture	Intégration des fichiers journaux. Regroupement vers une machine syslog. Mise en place de la sécurité avec ldap. Fichier de configurations.

Administration avancée JBoss

Automatisation

Le langage de scripts. Les objets accessibles. Introduction à OSGi. Requêtes d'interrogation sur le noyau JBoss.

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.

Administration WebSphere

Durée: 5 jours
2525 €

1^{er} au 5 février
13 au 17 juin

26 au 30 septembre
5 au 9 décembre

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 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. Chiffage 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. Chiffage. Mise en place d'un fournisseur externe d'authentification. Utilisation d'un JACC.

Administration WebSphere avancée

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.

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

Exploitation JOnAS

Sécurité avec JAAS et SecurityManager

Intégration de la sécurité dans le serveur d'applications.

Travaux pratiques :

Mise en place d'une politique de sécurité lors du déploiement de l'application. Sécurité: JAAS, rôles, groupes de permissions.

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,

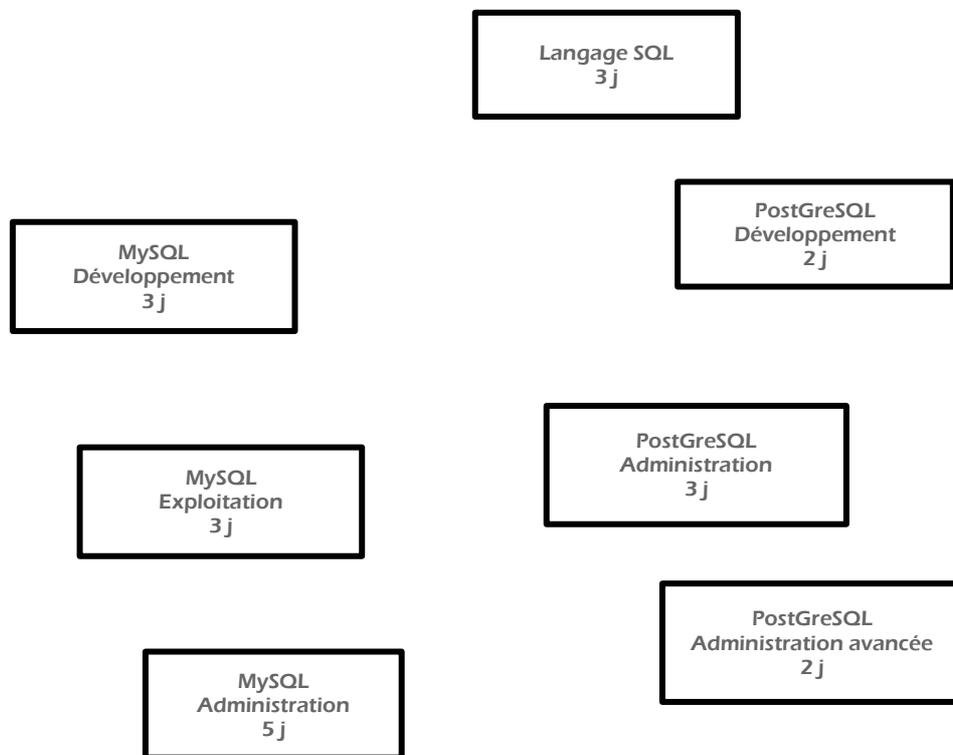
utilisation de Log4J en remplacement de monolog,

ventilation des journaux monolog vers une machine log4j.

Présentation et installation de JASMINE.

Utilisation de JASMINE en outil de suivi.

Filières Bases de Données SQL



Langage SQL

Durée: 3 jours
1505 €

8 au 10 mars
31 mai au 2 juin

13 au 15 septembre
8 au 10 novembre

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
1505 €

14 au 16 mars
6 au 8 juin

19 au 21 septembre
14 au 16 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 : Exploitation

Durée: 3 jours
1505 €

21 au 23 mars
13 au 15 juin

26 au 29 septembre
21 au 23 novembre

Public:

Toute personne souhaitant configurer, installer et exploiter 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
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

MySQL : Administration

Durée: 5 jours
2505 €

21 au 25 mars
13 au 17 juin

26 au 30 septembre
21 au 25 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
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
Optimisation	Les requêtes en cache. Les différents points à optimiser : les requêtes, les tables, la configuration du serveur, et de l'environnement.

MySQL : Administration

Partitionnement

Principe.

Les différents types de partitionnement (RANGE, LIST, HASH, ...)

Les apports.

Haute Disponibilité

Mise en oeuvre de la réplication

PostgreSQL : Développement

Durée: 2 jours
1005 €

11 au 12 février

21 au 22 avril

23 au 24 juin

8 au 9 septembre

28 au 29 novembre

Public:

Développeurs souhaitant mettre en place une application avec PostgreSQL

Objectifs:

Apprendre à créer des applications avec une base de données PostgreSQL

Connaissances préalables nécessaires:

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

Programme:

Présentation de PostgreSQL	PostgreSQL : serveur de base de données relationnelle objet Historique, licence, mode de développement Les composants du serveur PostgreSQL: serveur, client, connecteurs jdbc, tcl, pl, python Les apports de la version 9.0
Installation et configuration	Installation de PostgreSQL Installation avec un système de paquetage
Interagir avec le serveur	Le concept du client serveur Le client PostgreSQL Utiliser les clients graphiques
Spécificités SQL de PostgreSQL	Spécificités DDL (Définition des objets) Spécificités DML (Manipulation des données) Optimisation des requêtes
Programmation coté serveur	Importation et exportation de données L'intérêt de la programmation côté serveur et ses limites Mise en 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

PostgreSQL : Administration

Durée: 3 jours
1470 €

4 au 6 février
22 au 24 avril
17 au 19 juin

30 septembre au 2 octobre
16 au 18 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 \ pgadmin : 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

PostgreSQL : administration avancée

Durée: 2 jours
1030 €

18 au 19 février
28 au 29 avril
30 juin au 1er juillet

15 au 16 septembre
8 au 9 décembre

Public:

Administrateurs souhaitant approfondir leurs connaissances de l'administration de PostgreSQL

Objectifs:

Savoir configurer les sauvegardes et l'archivage, répondre aux contraintes de haute disponibilité, mettre en oeuvre la réplication,

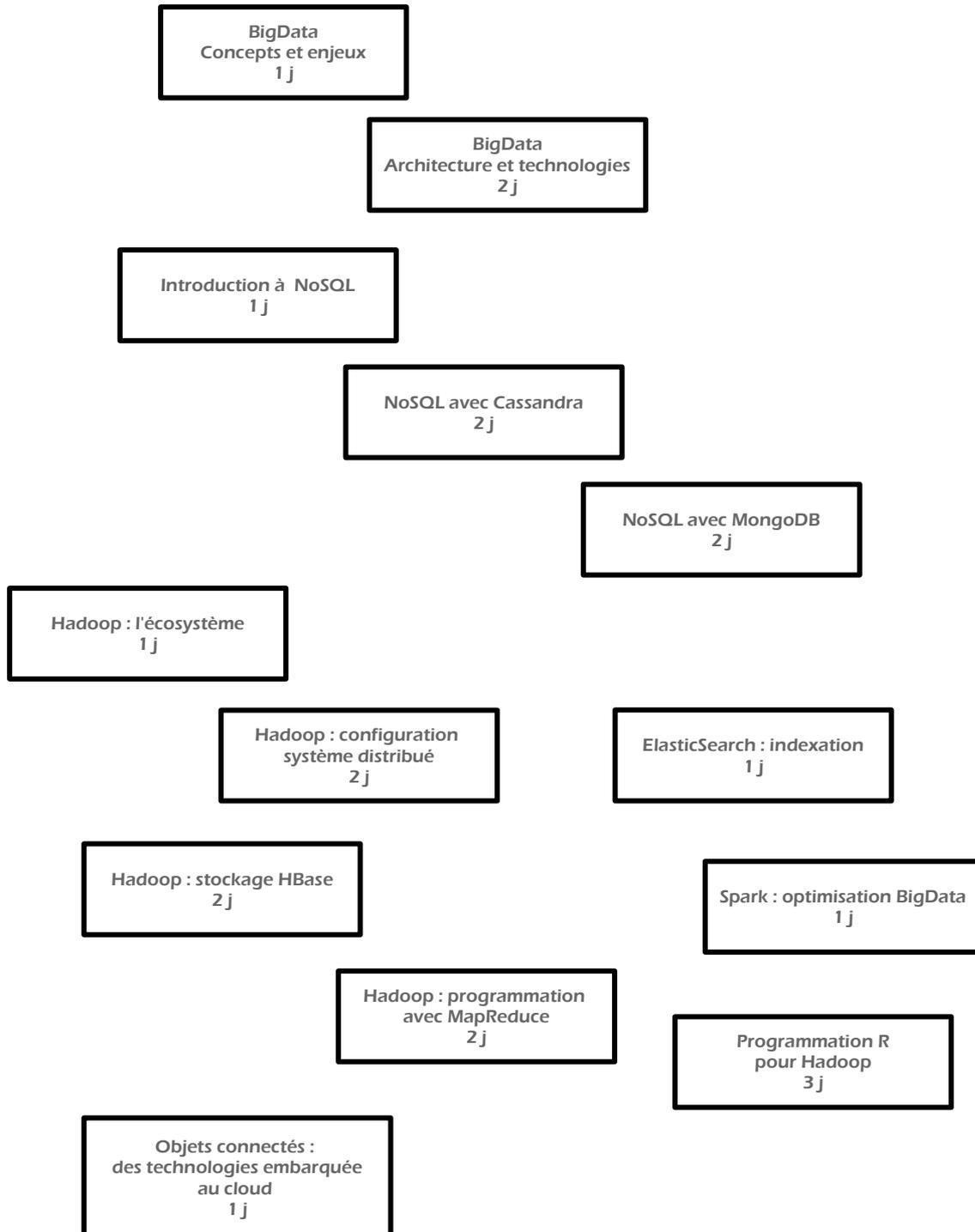
Connaissances préalables nécessaires:

Il est demandé aux participants de connaître l'administration PostgreSQL de base.

Programme:

Sauvegardes et archivage	<p>Les différentes méthodes et outils : sauvegarde SQL, système de fichiers, archivage continu. pgdump : principe, exemple de sauvegarde et restauration des données avec psql, pgdumpall : sauvegarde de toutes les bases d'une instance Archivage continu avec WAL Principe, configuration de l'archivage WAL Sauvegardes avec pg_basebackup Configuration de la récupération d'un archivage continu</p>
Haute disponibilité	<p>Différentes méthodes Principe des serveurs warm et hot standby. Utilisation des flux WAL. Mise en oeuvre du transfert de journaux et de la réplication en continu (streaming replication)</p>
Optimisation	<p>Outils de supervision de l'activité de la base de données Configuration des statistiques: paramètres : track_activities, track_count, track_functions, track_io_timing Contrôle des verrous : pg_locks Activation de traces dynamiques avec DTrace</p>

Filières BigData



BigData : concepts et enjeux

Durée: 1 jour
620 €

12 février
11 mai

2 septembre
30 novembre

Public:

Chefs de projets, architectes, et toute personne souhaitant connaître les impacts du BigData sur l'entreprise au niveau de l'organisation et des architectures.

Objectifs:

Comprendre les concepts et les apports du BigData, les impacts sur l'organisation de l'entreprise.

Connaissances préalables nécessaires:

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

Programme:

Introduction

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, prédictif, ...
Exemples : lutte contre la criminalité, fraude, santé, ressources énergétiques
Développement des techniques sur différents aspects : stockage, indexation/recherche, calcul.
Définition ETL : Extract Transform Load.
Les acteurs.
Le positionnement des technologies de cloud, BigData et noSQL.
Éléments d'architecture.

Stockage

Caractéristiques NoSQL
Les différents modes et formats de stockage.
Importance du théorème CAP.
Stockage réparti : réplication, sharding, gossip protocol, hachage,
Systèmes de fichiers distribués : GFS, HDFS,
Les bases de données : HBase, BigTable, ..

Calcul et restitution.

Apport des outils de calculs statistiques
Langages adaptés aux statistiques, liens avec les outils BigData.

BigData : concepts et enjeux

Evolutions

Liens vers les nouveaux métiers : Hadoop scientists, Data scientists, CDO.

Analyse des données au service de l'entreprise

Rôle de la DSI dans la démarche BigData.

Ouverture sur l'OpenData : principe, la démarche publique, les licences.

Exemple : portail data.gouv.fr

Les offres Saas BigData comme Google BigQuery.

Les limites. Les nouveautés annoncées

BigData Architecture et technologies

Durée: 2 jours
1105 €

7 au 8 mars
30 au 31 mai

19 au 20 septembre
9 au 10 novembre

Public:

Chefs de projets, architectes, et toute personne souhaitant connaître les outils et solutions pour savoir concevoir et mettre en oeuvre 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	Rappel des besoins et des caractéristiques techniques des projets BigData Le positionnement des technologies de cloud, BigData et noSQL. Eléments d'architecture.
Stockage	Caractéristiques NoSQL : adaptabilité, extensibilité, structure de données proches des utilisateurs, développeurs Les types de bases de données : clé/valeur, document, colonne, graphe. Données structurées et non structurées, documents, images, fichiers XML, JSON, CSV, ... Les différents modes et formats de stockage. Importance du théorème CAP. Stockage réparti : réplication, sharding, gossip protocol, hachage, Systèmes de fichiers distribués : GFS, HDFS, Les bases de données : HBase, BigTable, .. Quelques exemples de produits et leurs caractéristiques : Cassandra, MongoDB, CouchDB, DynamoDB, Riak, Hadoop. Qualité des données, gouvernance de données.
Indexation et recherche	Moteurs de recherche. Principe de fonctionnement. Méthodes d'indexation. Mise en oeuvre avec elasticsearch. Exemple de Lucene/solr. Recherche dans les bases de volumes importants. Exemples de produits et comparaison : Dremel, Drill, ElasticSearch, MapReduce,

BigData Architecture et technologies

Calcul et restitution,
intégration

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, sas, RStudio.

Ponts entre les outils statistiques et les bases BigData

Outils de calcul sur des volumes importants :

storm en temps réel, hadoop en mode batch.

Zoom sur Hadoop :

complémentarité de HDFS et MapReduce.

Restitution et analyse : logstash, kibana, elk, pentaho

Présentation de pig pour la conception de tâches MapReduce sur une grappe Hadoop.

Introduction à NoSQL

Durée: 1 jour
630 €

9 mars
1er juin

21 septembre
18 novembre

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.

Introduction à NoSQL

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
1610 €

29 février au 2 mars
23 au 25 mai

31 août au 2 septembre
5 au 7 octobre
30 novembre au 2 décembre

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 COL : 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 MongoDB

Durée: 3 jours
1610 €

15 au 17 février
30 mai au 1er juin

5 au 7 septembre
17 au 19 octobre
12 au 14 décembre

Public:

Chefs de projet, gestionnaires de bases de données.

Objectifs:

Comprendre le fonctionnement de MongoDB, savoir l'installer, le configurer, créer des requêtes d'interrogation, et mettre en oeuvre la réplication.

Connaissances préalables nécessaires:

Connaissance des principes classiques des bases de données.

Programme:

Introduction	Présentation MongoDB, historique du projet, les versions Structure des données : notions de documents, de collections Le format BSON (Binary JSON), comparaison avec JSON Fonctionnalités de MongoDB Interfaces disponibles
Installation et configuration	Plate-formes supportées Packages nécessaires, scripts de lancement Travaux pratiques : installation, lancement du service mongod Test de connexion
Interpréteur	Présentation du shell Mongo Initialisation et premières requêtes Choix d'une base de données Opérations CRUD : Create, Read, Update, Delete
Sécurité	Mise en oeuvre de l'authentification dans MongoDB Paramètres de configuration auth et keyFile Gestion des rôles Etude de la collection system.users
Exploitation	Importation, exportation de données. Sauvegardes Réplication : principe des replica sets et mise en oeuvre, Mécanisme de fail-over automatique Partitionnement des données avec le sharding

Hadoop : l'écosystème

Durée: 1 jour
645 €

21 mars
6 juin

5 septembre
21 novembre

Public:

Chefs de projets, développeurs, et toute personne souhaitant comprendre les mécanismes Hadoop et le rôle de chaque composant.

Objectifs:

Faire le point sur les différents éléments de l'écosystème Hadoop et leurs rôles respectifs.

Connaissances préalables nécessaires:

Connaissances générales des systèmes d'information.

Programme:

Introduction	<p>Les fonctionnalités apportées par Hadoop. Les principaux composants. Les distributions et leurs caractéristiques (HortonWorks, Cloudera, MapR, GreenPlum, Apache, ...)</p>
L'architecture	<p>Terminologie : NameNode, DataNode, ResourceManager Rôle et interactions des différents composants Présentation des outils : d'infrastructure : ambari, avro, zookeeper de gestion des données : pig, oozie, falcon, pentaho, sqoop, flume, d'interfaçage avec les applications GIS, de restitution et requêtage : webhdfs, hive, hawq, impalla, drill, stinger, tajo mahout, lucene, elasticSearch, Kibana</p>
Exemples interactifs	<p>Démonstrations sur une architecture Hadoop multi-noeuds. Mise à disposition d'un environnement pour des exemples de calcul Recherches dans des données complexes.</p>

Hadoop : configuration système distribué

Durée: 2 jours
1105 €

22 au 23 mars
7 au 8 juin

6 au 7 septembre
22 au 23 novembre

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. Les différentes versions.

Architecture et principe de fonctionnement.

Terminologie : NameNode, DataNode, ResourceManager, NodeManager.

Rôle des différents composants.

Le projet et les modules : Hadoop Common, HDFS, YARN, MapReduce

Ooziz, Pig, Hive, HBase, ...

Les outils Hadoop

Infrastructure/Mise en oeuvre :

Avro, Ambara, Zookeeper, Pig, Oozie, Falcon, Pentaho

Vue d'ensemble

Gestion des données.

Exemple de sqoop.

Restitution : webhdfs, hive, Hawq, Mahout, ElasticSearch ..

Outils complémentaires:

Spark, Shark, Storm, BigTop, Zebra

de développement : Cascading, Scalding, Flink, Pachyderm

d'analyse : RHadoop, Hama, Chukwa, kafka

Hadoop : configuration système distribué

Installation et configuration	<p>Trois modes d'installation : local, pseudo-distribué, distribué Première installation.Mise en oeuvre avec un seul noeud Hadoop.</p> <p>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</p> <p>Création des users pour les daemons hdfs et yarn,droits d'accès sur les exécutable et répertoires.</p> <p>Lancement des services.</p> <p>Démarrage des composants : hdfs, hadoop-daemon, yarn-daemon, etc ..</p> <p>Gestion de la grappe, différentes méthodes : ligne de commandes, API Rest, serveur http intégré, APIS natives</p> <p>Exemples en ligne de commandes avec hdfs, yarn, mapred</p> <p>Présentation des fonctions offertes par le serveur http</p> <p>Organisation et configuration d'une grappe</p>
Administration Hadoop	<p>Outils complémentaires à yarn et hdfs : jConsole, jconsole yarn suivi de charges journaux</p> <p>Gestion des noeuds, Accès JMX, mise en oeuvre d'un client JMX</p> <p>Administration HDFS stockage des fichiers, fsck, dfsadmin, gestion centralisée de caches avec Cacheadmin</p>
Sécurité	<p>Activation de la sécurité avec Kerberos dans core-site.xml, et dans hdfs-site.xml pour les NameNode et DataNode.</p> <p>Sécurisation de yarn avec la mise en oeuvre d'un proxy et d'un Linux Container Executor.</p>
Exploitation	<p>Supervision des éléments par le NodeManager. Visualisation des alertes en cas d'indisponibilité d'un noeud. Configuration des logs avec log4j.</p>

Hadoop : stockage avec HBase

Durée: 2 jours
1105 €

4 au 5 avril
20 au 21 juin

10 au 11 octobre
12 au 13 décembre

Public:

Chefs de projet, administrateurs et toute personne souhaitant stocker des données avec Hbase.

Objectifs:

Comprendre le fonctionnement de HBase, savoir mettre en place une configuration distribuée.

Connaissances préalables nécessaires:

Connaissance des principes de base Hadoop et des bases de données.

Programme:

Introduction	<p>Rappels rapides sur l'écosystème Hadoop. Fonctionnalités, architecture Présentation HBase. Historique. Lien avec HDFS. Format des données. Définitions : table, région, ligne, famille de colonnes, cellules, espace de nommage, ... Fonctionnalités : failover automatique, sharding, interface avec des jobs MapReduce.</p>
Architecture	<p>HBase master node, Region Master liens avec les clients HBase Rôle de Zookeeper.</p>
Installation	<p>Choix des packages. Installation et configuration dans le fichier conf/hbase-site.xml Démarrage en mode standalone start-hbase. Test de connexion avec hbase shell Installation en mode distribué. Interrogations depuis le serveur http intégré.</p>
HBase utilisation : shell	<p>Présentation des différentes interfaces disponibles. Travaux pratiques avec hbase shell. Commandes de base, syntaxe, variables, manipulation des données : create, list, put, scan, get désactiver une table ou l'effacer : disable (enable), drop, ... Programmation de scripts. Gestion des tables : principe des filtres. Mise en oeuvre de filtres de recherche, paramètres des tables. Les espaces de nommage.</p>

Hadoop : stockage avec HBase

Cluster HBase	Fonctionnement en mode distribué Première étape : fonctionnement indépendant des démons (HMaster, HRegionServer, Zookeeper) Mise en oeuvre avec HDFS dans un environnement distribué. Tables réparties : mise en oeuvre des splits.
Programmation	Introduction, les APIs (REST, Avro, Java, Ruby, ...) Utilisation d'un client Java . Gestion des tables. Lien avec MapReduce. Accès JMX. Exemple d'un client JMX

Hadoop : développement avec MapReduce

Durée: 2 jours
1105 €

24 au 25 mars
9 au 10 juin

8 au 9 septembre
24 au 25 novembre

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.

ElasticSearch : indexation

Durée: 1 jour
640 €

4 mars
10 mai

23 septembre
8 novembre

Public:

Chefs de projet, développeurs, architectes souhaitant mettre en oeuvre Elasticsearch.

Objectifs:

Comprendre le fonctionnement et les apports d'Elasticsearch dans le traitement de données, et savoir le configurer pour optimiser les requêtes

Connaissances préalables nécessaires:

Connaissances générales des systèmes d'informations

Programme:

Introduction	Présentation, fonctionnalités, licence Définitions et techniques d'indexation Positionnement Elasticsearch et les produits complémentaires : Shield, Watcher, Marvel, Kibana, Logstash, Beats Principe : base technique Lucene et apports d'ElasticSearch
Installation et configuration	Prérequis techniques. Utilisation de l'interface d'administration Marvel Etude du fichier : elasticsearch.yml
Outils d'interrogation	Java API avec "Node client" et "Transport client" Autres clients : Perl, Python, Ruby, etc... Interface http, travaux pratiques.
Traitement des données	Structure des données. stockage, indexation Mise à jour, suppression Gestion des accès concurrents. Format des données. Conversion au format JSON des données à traiter. Interrogations avec Search Lite et avec Query DSL (domain-specific language) Notion de 'filtre' pour affiner des requêtes.
Clustering	Définitions : cluster, noeud, sharding Calculs distribués avec Elasticsearch Stockage distribué

Programmation R pour Hadoop

Durée: 3 jours
1625 €

6 au 8 avril
22 au 24 juin

25 au 27 octobre
14 au 16 décembre

Public:

Chefs de projet, data scientists, développeurs.

Objectifs:

Savoir utiliser des programmes R dans un environnement Hadoop.

Connaissances préalables nécessaires:

Connaissance des bases Hadoop, et notions de calculs statistiques

Programme:

Présentation R	Le projet R Programming Calculs statistiques et génération de graphiques Points forts de R Programming Besoins du BigData Positionnement R programming par rapport à Hadoop
Mise en oeuvre de R	Travaux pratiques : installation et tests sur une plate-forme CentOS Utilisation de R en mode commande. Commandes de base. Syntaxe. Manipulations de nombres, vecteurs, tableaux, matrices, listes, etc ..
Intégration Hadoop	Différents moyens d'intégration : RHive : fonctions R de calculs statistiques s'appuyant sur HiveQL RHadoop : packages rmr2, rhdfs, rhbase
Travaux pratiques avec Hadoop	Installation d'un cluster, rmr2: traduction programmes R en mapreduce, rhdfs: API d'accès R à des données stockées sur HDFS rhbase: API d'accès à des données stockées sur HBase
Evolutions	Les acteurs : IBM avec BigInsights Big, Revolution R avec ScaleR

Spark Optimisation BigData

Durée: 1 jour
640 €

3 mars
9 mai

22 septembre
7 novembre

Public:

Chefs de projet, data scientists, développeurs.

Objectifs:

Savoir mettre en oeuvre Spark pour optimiser des calculs.

Connaissances préalables nécessaires:

Connaissance de Java ou Python, des bases Hadoop, et notions de calculs statistiques

Programme:

Introduction	Présentation Spark, origine du projet, apports, principe de fonctionnement Langages supportés.
Premiers pas	Utilisation du shell Spark avec Scala ou Python Gestion du cache
Règles de développement	Mise en pratique en Java et Python Notion de contexte Spark Différentes méthodes de création des RDD: depuis un fichier texte, un stockage externe. Manipulations sur les RDD (Resilient Distributed Dataset) Fonctions, gestion de la persistance.
Streaming	Objectifs , principe de fonctionnement Notion de StreamingContexte, DStreams, démonstrations
Cluster	Différents cluster managers : Spark en autonome, Mesos, Yarn, Amazon EC2 Architecture : SparkContext, Cluster Manager, Executor sur chaque noeud. Définitions : Driver program, Cluster manager, deploy mode, Executor, Task, Job Mise en oeuvre avec Spark et Amazon EC2 Soumission de jobs, supervision depuis l'interface web
Intégration hadoop	Travaux pratiques avec YARN Création et exploitation d'un cluster Spark/YARN.

Objets connectés:des OS embarqués au cloud

Durée: 1 jour
640 €

le 18 mars
le 27 juin

le 16 septembre
le 2 novembre

Public:

Décideurs,architectes,chefs de projet et toute personne souhaitant aborder les technologies des objets connectés.

Objectifs:

Comprendre quelles sont les briques technologiques mises en oeuvre dans les objets connectés :depuis les systèmes embarqués jusqu'au stockage des données en passant par les technologies réseaux utilisées.Ce cours est illustré de nombreuses démonstrations et travaux pratiques.

Connaissances préalables nécessaires:

Connaissance générale des systèmes d'informations.

Programme:

Introduction	Définitions,applications,services : domotique, santé, loisirs L'internet des objets Les acteurs et produits du marché
Bases embarquées	Plate-formes matérielles (Intel,Samsung) et logicielles Les systèmes classiques android, IOS, systèmes embarqués : UI, Brillo, LiteOS Kit de développement Galileo Les modules Arduino
Communications	Protocoles: bluetooth, wifi, 3G/4G, etc ... Avec un autre objet : M2M Avec le réseau internet:vers un serveur, vers le cloud Les plate-formes IoT, définition de standards : OpenInterconnect Consortium
Traitement des données	Types de données collectées Données locales, limites, Transfert et stockage sur une autre machine, dans le cloud ou sur internet Analyse des données et fourniture de services associés
Sécurité	Techniques de hacking des objets connectés Protection des données, législation

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 2016

Pythagore F.D. : Apprendre à Apprendre

1

Filières Systèmes

Filières Systèmes Linux

Linux/unix introduction.....	UX100	6
Le Shell.....	UX002	8
Atelier : Shell avance.....	UX004	9
Administration UNIX multi-plateformes.....	UX011	10
Administration AIX.....	UX018	12
Administration AIX avancée.....	UX019	14
Administration Solaris 11.....	UX040	16
Administration avancée Solaris.....	UX041	18
Introduction au logiciel libre.....	UX107	20
Administration Linux.....	UX111	22
Les services réseaux Linux.....	UX112	25
Haute disponibilité Linux.....	UX115	27
Linux système sécurisé.....	UX117	29
Linux sécurité des accès.....	UX118	32
Linux : optimisation performances métrologie.....	UX119	35
Administration avancée Linux.....	UX140	36
Administration Ubuntu Server.....	UX142	39
Virtualisation Linux.....	SY011	42
Virtualisation avec Xen.....	SY004	43
Virtualisation avec KVM.....	SY007	45
Virtualisation avec lxc.....	SY008	47
Docker : mise en oeuvre.....	UX180	48

Filières Systèmes embarqués et mobilité

Systèmes Linux embarqués.....	UX125	53
Développement d'applications Android.....	UX128	55

Filières Réseaux et TCP/IP

Introduction aux réseaux.....	RS001	58
TCP/IP : protocoles et mise en oeuvre.....	RS003	60
IP: Migration vers IPv6.....	RS014	62
Messagerie.....	RS006	64
Mise en oeuvre du protocole SNMP.....	RS022	65
Administration réseaux IP.....	RS007	67
Sécurité TCP/IP.....	RS018	68
Voix sur IP.....	RS023	69
Annuaire LDAP.....	RS122	71

Filière Production et supervision

Supervision nagios : utilisation.....	RS129	74
Administration Nagios.....	RS130	76
Supervision avec shinken.....	RS135	79
Reporting avec Cacti.....	RS134	80
Supervision avec Centreon.....	RS140	81
Zabbix administration.....	RS150	82
Gestion de configuration avec Chef.....	UX170	84
Puppet : administration centralisée.....	UX161	86
Puppet : expertise.....	UX162	88
Gestion de Parc avec OCS et GLPI.....	UX124	89

Filières Internet et Web

Serveur WEB : apache.....	RS104	92
Administration serveur Nginx.....	RS105	94
Serveur Proxy Squid.....	RS114	96
Sécurisation des services internet.....	RS115	97
Développement Web HTML et CSS.....	RS102	100
Web Dynamique avec JavaScript.....	RS106	102
Développement Web avec GWT.....	LW001	104

Filières Virtualisation et Cloud

Virtualisation Linux.....	SY011 106
Virtualisation avec Xen.....	SY004 107
Virtualisation avec KVM.....	SY007 109
Virtualisation avec lxc.....	SY008 111
Docker : mise en oeuvre.....	UX180 112
Réseaux virtuels: administration.....	SY012 114
Cloud : technologies et enjeux.....	SY100 115
Architecture cloud d'entreprise.....	SY101 116
CloudStack : mise en oeuvre.....	SY112 118
OpenStack : configuration et administration.....	SY111 120
AWS : architecture SI, réseaux virtuels.....	SY200 122
AWS : stockage,S3,dynamodb.....	SY201 123
AWS : les API EC2 et S3.....	SY202 124

Filières Développement

Programmation en langage C.....	LC001 126
Perfectionnement en langage C.....	LC002 128
Programmation système en C sur Unix/Linux.....	LC010 130
Programmation noyau et drivers en C sur Linux.....	LC011 133
Debugging en C sur Unix.....	UX015 134
Développement d'applications Web avec PHP.....	LH001 135
Programmation Perl.....	LP001 137
Le langage Python.....	LY001 139
Développement avancé avec Python.....	LY002 141

Filières Langages Java et JEE

Concepts Objets et programmation Java.....	LJ001 143
Sécurité Java et JEE.....	LJ005 145
JEE : Développement d'applications web.....	AS004 147
JEE : les EJB.....	AS005 148
Développement Web Services.....	AS006 150
Serveurs d'application JEE.....	AS011 152
Conception JEE avec les Design Patterns.....	AS122 154
Administration Tomcat.....	AS133 155
Administration Jboss.....	AS134 158
Administration avancée Jboss.....	AS144 160
Administration WebSphere.....	AS131 162
Administration WebSphere avancée.....	AS135 164
Exploitation JonAS.....	AS136 166

Filières Bases de Données SQL

Langage SQL.....	BD001 169
MySQL : Développement.....	BD010 170
MySQL : Exploitation.....	BD012 171
MySQL : Administration.....	BD011 172
PostgreSQL : Développement.....	BD020 174
PostgreSQL : Administration.....	BD021 175
PostgreSQL :administration avancée.....	BD022 176

Filières BigData

BigData : concepts et enjeux.....	CB000 178
BigData Architecture et technologies.....	CB001 180
Introduction à NoSQL.....	CB002 182
Base de données NoSQL avec Cassandra.....	CB010 184
Base de données NoSQL avec MongoDB.....	CB017 185
Hadoop : l'écosystème.....	CB015 186
Hadoop : configuration système distribué.....	CB013 187
Hadoop : stockage avec Hbase.....	CB016 189
Hadoop : développement avec MapReduce.....	CB014 191
ElasticSearch : indexation.....	CB020 192
Programmation R pour Hadoop.....	CB018 193
Spark Optimisation BigData.....	CB019 194
Objets connectés:des OS embarqués au cloud.....	CB100 195