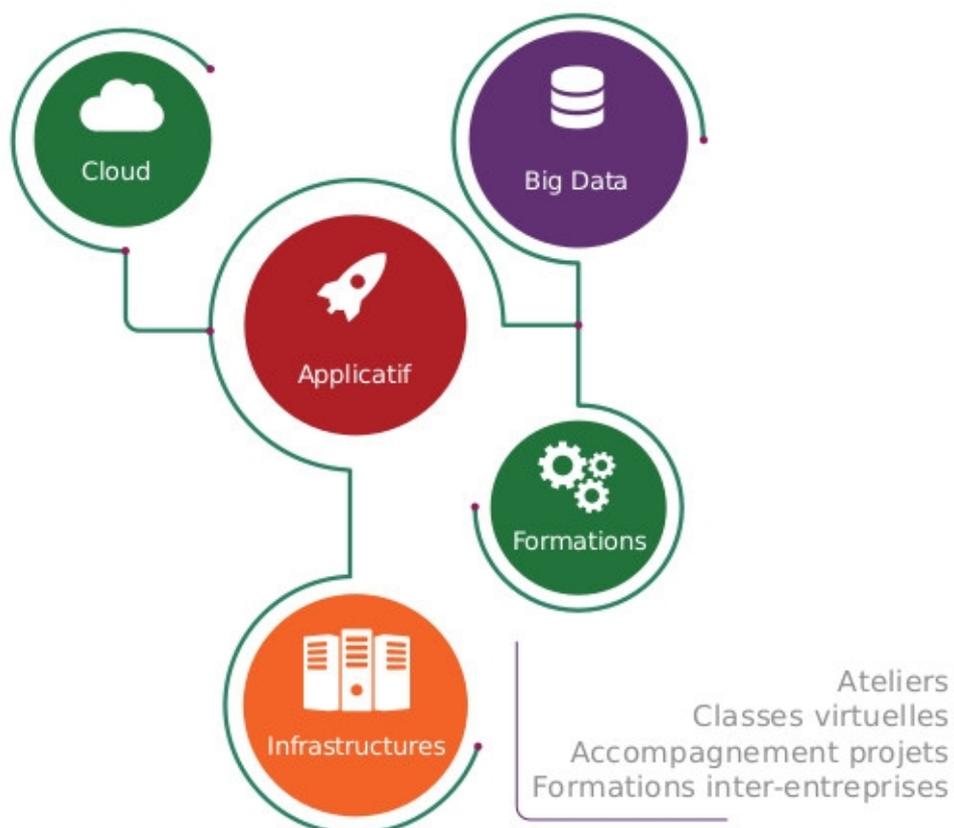


## Catalogue formations 2018



## Pythagore F.D. : Apprendre à Apprendre

### Nouveautés 2018 :

Nous proposons de nouvelles filières de formations BigData, des ateliers avec nos experts sur les IoT, la sécurité Hadoop, l'optimisation des conteneurs avec Kubernetes, les flux de données avec Storm, les solutions analytics de l'écosystème Hadoop.

### Nos méthodes pédagogiques :

Apprendre, concevoir, intégrer ... le meilleur moyen de comprendre les nouveaux concepts et les technologies est la mise en pratique.

Le matériel pédagogique mis à disposition pour chaque formation permet de mettre en pratique tous les concepts abordés : clusters de calcul pour MapReduce, baies de stockage distribué pour cassandra, accès Amazon pour la gestion de ressources dans le cloud, etc ...

Nos formations sont disponibles en stages inter-entreprises, en intra dans notre centre de formation ou sur site, et en classes virtuelles.

### ***Apprendre, concevoir, intégrer ...***

### Nos domaines d'expertise :

- les bases de données et le BigData avec NoSQL, Cassandra, MongoDB, Hadoop, Spark, Storm ...
- la virtualisation et l'orchestration avec xen, kvm, lxc, Docker, et le cloud : cloudstack et openstack, openNebula, cobbler, etc ...
- TCP/IP (IPv6, snmp, Architecture, Sécurité, Administration de réseaux IP, VoIP, ...)
- Unix et Linux, et les applicatifs Apache, Openldap, Squid, Nagios, Zabbix, OCS/GLPI, puppet , chef..
- Développement (langage C, Java, Jee, technologies Jee, JBoss, WebServices, PHP, Perl, Python , ...)

### Notre centre de formation :

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

01 55 33 52 10  
[www.pythagore-fd.fr](http://www.pythagore-fd.fr)

**Pythagore F.D.**

Apprendre à apprendre





## Filières BigData

BigData concepts et enjeux p6  
BigData Architecture et technologies p8

Introduction à NoSQL p10  
Stockage distribué avec Ceph p12  
BigData avec Cassandra p13  
NoSQL avec mongoDB p15  
Neo4J Graphes et analyse p16

Indexation avec Elasticsearch p17  
Elasticsearch programmation p18  
Elasticsearch : infrastructure et administration p21

Hadoop Ecosystème p23  
Administration Hadoop HortonWorks p25  
Administration Hadoop Cloudera p28  
Hadoop Développement p31  
Hadoop : stockage avec HBase p33  
Hadoop : Infrastructure sécurisée p35  
Hadoop : Analytics p36

Spark Mise en oeuvre et programmation p38  
Flux de données avec Storm p40  
Programmation Scala p41

Objets connectés : de l'OS embarqué au cloud p4  
Blockchain : principes et technologies p5



## BigData : concepts et enjeux

**Durée:** 1 jour  
650 €

12 février  
28 mai

3 septembre  
12 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.  
 Définitions : ETL, Extract Transform Load, CAP, 3V, 4V, ...  
 Les acteurs.  
 Le positionnement des technologies de cloud, BigData et noSQL.  
 Eléments d'architecture.
- Stockage** : Caractéristiques NoSQL  
 Les différents modes et formats de stockage. Besoin de distribution. Elasticité.  
 Stockage réparti : réplication, sharding, gossip protocol, hachage,  
 Systèmes de fichiers distribués : GFS, HDFS, Ceph  
 Les bases de données : Cassandra, HBase, MongoDB, CouchBase, Riak, BigTable, ..

## **BigData : concepts et enjeux**

Calcul et restitution.: Apport des outils de calculs statistiques  
Langages adaptés aux statistiques, liens avec les outils BigData.

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  
1160 €

14 au 15 février  
30 au 31 mai

5 et 6 septembre  
14 au 15 novembre

## Public:

Chefs de projets, architectes, développeurs, data-scientists, et toute personne souhaitant connaître les outils et solutions pour concevoir et mettre en oeuvre une architecture BigData.

## Objectifs:

Comprendre les concepts essentiels du BigData, et les technologies implémentées. Savoir analyser les difficultés propres à un projet BigData, les freins, les apports, tant sur les aspects techniques que sur les points liés à la gestion du projet.

## Connaissances préalables nécessaires:

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

## Programme:

- Introduction** : L'essentiel du BigData : calcul distribué, données non structurées.  
Besoins fonctionnels et caractéristiques techniques des projets.  
La valorisation des données.  
Le positionnement respectif des technologies de cloud, BigData et noSQL, et les liens, implications.  
Quelques éléments d'architecture.  
L'écosystème du BigData : les acteurs, les produits, état de l'art.  
Cycle de vie des projets BigData.  
Emergence de nouveaux métiers : Datascientists, Data labs, ...
- 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.  
Stockage réparti : réplication, sharding, gossip protocol, hachage,  
Systèmes de fichiers distribués : GFS, HDFS,  
Quelques exemples de produits et leurs caractéristiques : Cassandra, MongoDB, CouchDB, DynamoDB, Riak, Hadoop, HBase, BigTable, ...  
Qualité des données, gouvernance de données.

# BigData Architecture et technologies

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,

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  
650 €

16 février  
1er juin

7 septembre  
16 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, protocole gossip, 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



## Stockage distribué avec Ceph

**Durée:** 2 jours  
1160 €

15 au 16 mars

14 au 15 juin

20 au 21 septembre

29 au 30 novembre

### Public:

Administrateurs systèmes Linux souhaitant utiliser un système de stockage distribué.

### Objectifs:

Comprendre le fonctionnement de Ceph, savoir le mettre en oeuvre et le configurer.

### Connaissances préalables nécessaires:

Connaissance de l'administration des systèmes Linux.

### Programme:

- Introduction** : Système de stockage distribué, Historique et intégration dans le noyau Linux. Fonctionnalités et points forts.  
Stockage objet, stockage de blocs, système de fichiers CephFS.  
Interfaçage avec des plate-formes de cloud
- Mise en oeuvre** : Architecture : Ceph Monitor, services OSD, Ceph Metadata Server  
Configuration d'un cluster de stockage Ceph multi-noeuds  
Gestion des noeuds, des serveurs de stockage, des Ceph Monitors
- Clients Ceph** : Différents clients Ceph selon la fonctionnalité utilisée.  
Stockage de blocs : installation Ceph et configuration d'un bloc et montage depuis un poste client  
Système de fichiers Ceph : création d'un système de fichier, clé d'authentification, montage du file system  
Stockage objet : installation et création d'une instance Ceph Object Gateway  
APIs de stockage objet Ceph : compatibilité AWS S3, et OpenStack Swift
- Sécurité et fiabilité** : Architecture haute disponibilité. Installation et configuration de ceph-mgr (Ceph Manager Daemon). Tests de fiabilité.

# BigData avec Cassandra

**Durée:** 3 jours  
1710 €

12 au 14 mars  
11 au 13 juin

17 au 19 septembre  
26 au 28 novembre

## Public:

Chefs de projet, gestionnaires de bases de données.

## Objectifs:

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

## Connaissances préalables nécessaires:

Connaissances générales sur les bases de données.

## Programme:

- Introduction : Historique, fonctionnalités de Cassandra, licence  
Format des données, "key-value", traitement de volumes importants, haute disponibilité, système réparti de base de données, ...
  
- Installation et configuration : Prérequis. Plate-formes supportées. Etude du fichier de configuration : conf/cassandra.yaml  
Répertoire de travail, de stockage des données, gestion de la mémoire.  
Démarrage d'un noeud et test de l'interface cliente cqlsh.
  
- CQL : Commandes de base : connexion au système de base de données, création de colonnes, insertion, modification recherche,  
Le CQL : Cassandra Query Language. Exécution de scripts.  
Comment écrire des requêtes? Approches.
  
- Gestion de la grappe : 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.
  
- Supervision : OpsCenter : installation, lancement. Utilisation de base.  
Supervision avec nodetool cfstats, ou export JMX vers des outils de supervision comme Nagios.



## BigData avec Cassandra

- Exploitation : Sauvegardes. Import/export au format JSON.
- Support Hadoop : Principe de MapReduce. Implémentation Hadoop. Mise en oeuvre depuis Cassandra.
- Support Spark : Description rapide de l'architecture spark. Mise en oeuvre depuis Cassandra.  
Execution de travaux Spark s'appuyant sur une grappe Cassandra.

# Base de données NoSQL avec MongoDB

**Durée:** 3 jours  
1710 €

7 au 9 mars  
4 au 6 juin

8 au 10 octobre  
17 au 19 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. Tests 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  
660 €

19 février

23 avril

15 juin

10 septembre

5 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. Comprendre l'architecture des applicatifs hadoop et savoir quels sont les apports et les cas d'usage des solutions hadoop.

### Connaissances préalables nécessaires:

Connaissances générales des systèmes d'information.

### Programme:

- Introduction : Rappels sur NoSQL. Le théorème CAP.  
 Historique du projet hadoop  
 Les fonctionnalités : stockage, outils 'extraction, de conversion, ETL, analyse, ...  
 Exemples de cas d'utilisation sur des grands projets.  
 Les principaux composants :  
 HDFS pour le stockage et YARN pour les calculs.  
 Les distributions et leurs caractéristiques (HortonWorks, Cloudera, MapR, GreenPlum, Apache, ...)
- L'architecture : Terminologie : NameNode, DataNode, ResourceManager  
 Rôle et interactions des différents composants  
 Présentation des outils d'infrastructure : ambari, avro, zookeeper;  
 de gestion des données : pig, oozie, tez, falcon, pentaho, sqoop, flume;  
 d'interfaçage avec les applications GIS;  
 de restitution et requête : webhdfs, hive, hawq, impala, drill, stinger, tajo, mahout, lucene, elasticSearch, Kibana  
 Les architectures connexes : spark, cassandra

## Hadoop : l'écosystème

Exemples interactifs : Démonstrations sur une architecture Hadoop multi-noeuds.  
Mise à disposition d'un environnement pour des exemples de calcul  
Travaux pratiques :  
Recherches dans des données complexes non structurées.

Applications : Cas d'usages de hadoop.  
Les infrastructures hyperconvergées avec les appliances hadoop  
Calculs distribués sur des clusters hadoop



# Hadoop Hortonworks : administration avec Ambari

**Durée:** 3 jours  
1830 €

20 au 22 février  
24 au 26 avril

11 au 13 septembre  
6 au 8 novembre

## Public:

Chefs de projet, administrateurs et toute personne souhaitant mettre en oeuvre un système distribué avec Hadoop. Les travaux pratiques sont réalisés selon le choix des participants sur une distribution Apache ou Cloudera ou Hortonworks.

## 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.  
Distributions : Apache, Cloudera, Hortonworks, EMR, MapR, DSE.  
Spécificités de chaque distribution.  
Architecture et principe de fonctionnement.  
Terminologie : NameNode, DataNode, ResourceManager, NodeManager.  
Rôle des différents composants.  
Le projet et les modules : Hadoop Common, HDFS, YARN, Spark, MapReduce  
Oozie, Pig, Hive, HBase, ...
- Les outils Hadoop** : Infrastructure/Mise en oeuvre :  
Avro, Ambari, Zookeeper, Pig, Tez, Oozie, Falcon, Pentaho  
Vue d'ensemble  
Gestion des données.  
Exemple de sqoop.  
Restitution : webhdfs, hive, Hawq, Mahout, ElasticSearch ..  
Outils complémentaires:  
Spark, SparkQL, SparkMLib, Storm, BigTop, Zebra  
de développement : Cascading, Scalding, Flink/Pachyderm  
d'analyse : RHadoop, Hama, Chukwa, kafka

## **Hadoop Hortonworks : administration avec Ambari**

- Installation et configuration** : Trois modes d'installation : local, pseudo-distribué, distribué  
Première installation.  
Mise en oeuvre avec un seul noeud Hadoop.  
Configuration de l'environnement, étude des fichiers de configuration :  
core-site.xml, hdfs-site.xml, mapred-site.xml, yarn-site.xml et capacity-scheduler.xml  
Création des users pour les daemons hdfs et yarn, droits d'accès sur les exécutable et répertoires.  
Lancement des services.  
Démarrage des composants : hdfs, hadoop-daemon, yarn-daemon, etc ..  
Gestion de la grappe, différentes méthodes :  
ligne de commandes, API Rest, serveur http intégré, APIS natives  
Exemples en ligne de commandes avec hdfs, yarn, mapred  
Présentation des fonctions offertes par le serveur http  
Travaux pratiques :  
Organisation et configuration d'une grappe hadoop
- Administration Hadoop** : Outils complémentaires à yarn et hdfs : jConsole, jconsole yarn  
Exemples sur le suivi de charges, l'analyse des journaux.  
Principe de gestion des noeuds, accès JMX.  
Travaux pratiques :  
mise en oeuvre d'un client JMX  
Administration HDFS :  
présentation des outils de stockage des fichiers, fsck, dfsadmin  
Mise en oeuvre sur des exemples simples de récupération de fichiers  
Gestion centralisée de caches avec Cacheadmin
- Sécurité** : Mécanismes de sécurité et mise en oeuvre pratique :  
Activation de la sécurité avec Kerberos dans core-site.xml, et dans hdfs-site.xml pour les NameNode et DataNode. Sécurisation de yarn avec la mise en oeuvre d'un proxy et d'un Linux Container Executor.
- Exploitation** : Installation d'une grappe Hadoop avec Ambari. Tableau de bord.  
Lancement des services.  
Principe de la supervision des éléments par le NodeManager.  
Monitoring graphique avec Ambari.  
Présentation de Ganglia, Kibana  
Travaux pratiques :  
Visualisation des alertes en cas d'indisponibilité d'un noeud.  
Configuration des logs avec log4j.



## Hadoop Cloudera : administration

**Durée:** 3 jours  
1830 €

26 au 28 février  
18 au 20 juin

1er au 3 octobre  
10 au 12 décembre

### Public:

Chefs de projet, administrateurs et toute personne souhaitant mettre en oeuvre un système distribué avec Hadoop. Les travaux pratiques sont réalisés selon le choix des participants sur une distribution Apache ou Cloudera ou Hortonworks.

### 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.  
Distributions : Apache, Cloudera, Hortonworks, EMR, MapR, DSE.  
Spécificités de chaque distribution.  
Architecture et principe de fonctionnement.  
Terminologie : NameNode, DataNode, ResourceManager, NodeManager.  
Rôle des différents composants.  
Le projet et les modules : Hadoop Common, HDFS, YARN, Spark, MapReduce  
Oozie, Pig, Hive, HBase, ...
- Les outils Hadoop : Infrastructure/Mise en oeuvre :  
Avro, Ambari, Zookeeper, Pig, Tez, Oozie, Falcon, Pentaho  
Vue d'ensemble  
Gestion des données.  
Exemple de sqoop.  
Restitution : webhdfs, hive, Hawq, Mahout, ElasticSearch ..  
Outils complémentaires:  
Spark, SparkQL, SparkMLib, Storm, BigTop, Zebra  
de développement : Cascading, Scalding, Flink/Pachyderm  
d'analyse : RHadoop, Hama, Chukwa, kafka

## Hadoop Cloudera : administration

- Installation et configuration** : Trois modes d'installation : local, pseudo-distribué, distribué  
 Première installation.  
 Mise en oeuvre avec un seul noeud Hadoop.  
 Présentation de Cloudera Manager.  
 Configuration de l'environnement, étude des fichiers de configuration :  
 core-site.xml, hdfs-site.xml, mapred-site.xml, yarn-site.xml et capacity-scheduler.xml  
 Création des users pour les daemons hdfs et yarn, droits d'accès sur les exécutable et répertoires.  
 Lancement des services.  
 Démarrage des composants : hdfs, hadoop-daemon, yarn-daemon, etc ..  
 Gestion de la grappe, différentes méthodes :  
 ligne de commandes, API Rest, serveur http intégré, APIS natives  
 Exemples en ligne de commandes avec hdfs, yarn, mapred  
 Présentation des fonctions offertes par le serveur http  
 Travaux pratiques :  
 Organisation et configuration d'une grappe hadoop avec Cloudera Manager  
 Traitement de données.  
 Requêtage SQL avec Impala.
- Administration Hadoop** : Outils complémentaires à yarn et hdfs : jConsole, jconsole yarn  
 Exemples sur le suivi de charges, l'analyse des journaux.  
 Principe de gestion des noeuds, accès JMX.  
 Travaux pratiques :  
 mise en oeuvre d'un client JMX  
 Administration HDFS :  
 présentation des outils de stockage des fichiers, fsck, dfsadmin  
 Mise en oeuvre sur des exemples simples de récupération de fichiers  
 Gestion centralisée de caches avec Cacheadmin
- Sécurité** : Mécanismes de sécurité et mise en oeuvre pratique :  
 Activation de la sécurité avec Kerberos dans core-site.xml, et dans hdfs-site.xml pour les NameNode et DataNode. Sécurisation de yarn avec la mise en oeuvre d'un proxy et d'un Linux Container Executor.



## **Hadoop Cloudera : administration**

Exploitation : Installation d'une grappe Hadoop. Lancement des services.  
Principe de la supervision des éléments par le NodeManager.  
Présentation de Ganglia, Kibana  
Travaux pratiques :  
Visualisation des alertes en cas d'indisponibilité d'un noeud.  
Configuration des logs avec log4j.



## Hadoop : développement

**Durée:** 3 jours  
1830 €

5 au 7 mars  
25 au 27 juin

24 au 26 septembre  
3 au 5 décembre

### Public:

Chefs de projets, développeurs, data-scientists, 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, Spark, MapReduce  
Utilisation de yarn pour piloter les jobs 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.  
Travaux pratiques :  
Rédaction d'un premier programme et exécution avec Hadoop.
- Programmation** : Configuration des jobs, notion de configuration.  
Les interfaces principales : mapper, reducer,  
La chaîne de production : entrées, input splits, mapper, combiner, shuffle/sort, reducer, sortie.  
partitioner, outputcollector, codecs, compresseurs..  
Format des entrées et sorties d'un job MapReduce : InputFormat et OutputFormat.  
Travaux pratiques : type personnalisés : création d'un writable spécifique. Utilisation. Contraintes.



## Hadoop : développement

- Outils complémentaires** : Mise en oeuvre du cache distribué.  
Paramétrage d'un job : ToolRunner, transmission de propriétés.  
Accès à des systèmes externes : S3, hdfs, har, ...  
Travaux pratiques : répartition du job sur la ferme au travers de yarn.
- Streaming** : Définition du streaming map/reduce.  
Création d'un job map/reduce en python.  
Répartition sur la ferme. Avantage et inconvénients.  
Liaisons avec des systèmes externes.  
Introduction au pont HadoopR  
Travaux pratiques : suivi d'un job en streaming.
- Pig** : Présentation des pattern et best practices Map/reduce.  
Introduction à Pig.  
Caractéristiques du langage : latin.  
Travaux pratiques : installation/lancement de pig.  
Ecriture de scripts simples pig. Les fonctions de base.  
Ajouts de fonctions personnalisées. Les UDF. Mise en oeuvre.
- Hive** : Simplification du requêtage. Etude de la syntaxe de base.  
Travaux pratiques : Création de tables. Ecriture de requêtes.  
Comparaison pig/hive.
- Sécurité en environnement hadoop** : Mécanisme de gestion de l'authentification.  
Travaux pratiques : configuration des ACLs.

## Hadoop : stockage avec HBase

**Durée:** 2 jours  
1160 €

1<sup>er</sup> et 2 mars  
21 et 22 juin

4 et 5 octobre  
13 au 14 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** : Rappels rapides sur l'écosystème Hadoop.  
Les fonctionnalités du framework Hadoop  
Le projet et les modules : Hadoop Common, HDFS, YARN, Spark, MapReduce  
Présentation HBase. Historique. Lien avec HDFS.  
Format des données dans HBase  
Définitions : table, région, ligne, famille de colonnes, cellules, espace de nommage, ...  
Fonctionnalités : failover automatique, sharding, interface avec des jobs MapReduce.
- Architecture** : HBase master node, Region Master, liens avec les clients HBase.  
Présentation du rôle de Zookeeper.
- Installation** : 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.  
Travaux pratiques :  
Interrogations depuis le serveur http intégré.



## Hadoop : stockage avec HBase

- HBase utilisation :** : Présentation des différentes interfaces disponibles.  
shell  
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.  
Présentation des espaces de nommage.
- Cluster HBase** : Fonctionnement en mode distribué  
Première étape : fonctionnement indépendant des démons (HMaster, HRegionServer, Zookeeper)  
Passage au mode distribué :  
mise en oeuvre avec HDFS dans un environnement distribué.  
Travaux pratiques :  
sur un exemple de tables réparties : mise en oeuvre des splits.
- Programmation** : Introduction, les APIs (REST, Avro, Thrift, Java, Ruby, ...)  
Utilisation d'un client Java.  
Gestion des tables. Lien avec MapReduce.  
Principe des accès JMX.  
Travaux pratiques :  
création d'un client JMX

# Hadoop : infrastructure sécurisée

**Durée:** 1 jour  
660 €

23 février  
27 avril

14 septembre  
9 novembre

## Public:

Chefs de projet, administrateurs et toute personne souhaitant sécuriser une infrastructure hadoop.

## Objectifs:

Comprendre les mécanismes de sécurité hadoop, et savoir les mettre en oeuvre.

## Connaissances préalables nécessaires:

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

## Programme:

- Introduction : Rappels rapides sur l'écosystème Hadoop.  
Le projet et les modules : Hadoop Common, HDFS, YARN, Spark, MapReduce  
Les risques et points à sécuriser dans un système distribué
  
- Architecture sécurité hadoop : Sécurisation réseau, système d'exploitation, les rôles hadoop et stratégies
  
- Kerberos : Principe de fonctionnement. Travaux pratiques: kerberisation d'une grappe hadoop
  
- Sécurité des accès : Authentification, autorisations, accounting.Travaux pratiques: gestion des autorisations dans HDFS, YARN, HBase, ..Mise en oeuvre des ACLs dans Zookeeper
  
- Apache Sentry : Présentation du projet, architecture : sentry server, sentry plugin. Gestion de l'authentification et des droits d'accès aux données.Travaux pratiques: intégration avec Hadoop
  
- Sécurité des données : Cryptage des données stockées, et en transit. Mécanisme de sécurité des données en entrée et en consultation par des accès clients: interface hadoop en ligne de commande, sqoop, oozie, HBase, webHDFS, httpFS



## Hadoop : analytics

**Durée:** 2 jours  
1160 €

5 au 6 février  
24 au 25 mai

20 au 21 septembre  
6 au 7 décembre

### Public:

Chefs de projet, développeurs, data scientists, architectes souhaitant mettre en oeuvre des solutions analytics avec hadoop.

### Objectifs:

Savoir mettre en oeuvre les frameworks analytics dans un environnement hadoop.

### Connaissances préalables nécessaires:

Connaissances des principes du BigData, d'un langage de programmation comme Java ou Scala ou Python.

### Programme:

- Introduction** : Définitions : Analytics.  
Arbres de décision, de régression, régression automatique  
Classifieurs, scoring.  
Apprentissage supervisé, apprentissage automatique  
Etapes de préparation des données.  
Présentation du data munging.
- Hadoop et les outils:** Rôle des différents composants :  
d'analyse socle hadoop, yarn, hdfs  
Frameworks analytics : Mahout, Flink, Spark ML
- Mahout** : Principe de fonctionnement.  
Sources de données, format de stockage des données,  
Génération de recommandations, traitement, filtrage  
Exemples de base : génération de recommandations, traitement,  
filtrage  
Présentation des algorithmes les plus courants.  
Compatibilité avec Hadoop Yarn, Spark, H2O, Flink
- Flink** : Origine du projet, fonctionnalités  
Traitement distribué de flux de données, en temps réel ou batch  
APIs disponibles  
Mise en oeuvre avec des programmes Java/Scala  
Analyse de graphe avec l'API Gelly

## Hadoop : analytics

- Spark MLib : Fonctionnalités : Machine Learning avec Spark, algorithmes standards, gestion de la persistance, statistiques. Support de RDD. Mise en oeuvre avec les DataFrames.
- GraphX : Fourniture d'algorithmes, d'opérateurs simples pour des calcul statistiques sur les graphes  
Travaux pratiques : exemples d'opérations sur les graphes.



## Spark Mise en oeuvre et programmation

**Durée:** 3 jours  
1750 €

26 au 28 mars  
27 au 30 juin

16 au 18 octobre  
17 au 19 décembre

### 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.
- Cluster : Différents cluster managers : Spark en autonome, avec Mesos, avec Yarn, avec 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.

## Spark Mise en oeuvre et programmation

Support Cassandra : Description rapide de l'architecture Cassandra. Mise en oeuvre depuis Spark.  
Exécution de travaux Spark s'appuyant sur une grappe Cassandra.

Spark SQL : Objectifs : traitement de données structurées,  
Optimisation des requêtes.  
Mise en oeuvre de Spark SQL.  
Comptabilité Hive  
Travaux pratiques:  
en ligne de commande avec Spark SQL,  
avec un pilote JDBC.  
L'API Dataset :  
disponible avec Scala ou Java.  
Collections de données distribuées.  
Exemples.

Streaming : Objectifs , principe de fonctionnement : stream processing.  
Source de données : HDFS, Flume, Kafka, ...  
Notion de StreamingContext, DStreams, démonstrations  
Travaux pratiques : traitement de flux DStreams en Java.

MLib : Fonctionnalités : Machine Learning avec Spark,  
algorithmes standards,  
gestion de la persistance,  
statistiques.  
Support de RDD.  
Mise en oeuvre avec les DataFrames.

GraphX : Fourniture d'algorithmes, d'opérateurs simples  
pour des calcul statistiques sur les graphes  
Travaux pratiques :  
exemples d'opérations sur les graphes.



## Flux de données avec Storm

**Durée:** 2 jours  
1160 €

15 au 16 mars  
28 au 29 juin

25 au 26 octobre  
13 au 14 décembre

### Public:

Chefs de projet, data scientists, développeurs.

### Objectifs:

Savoir mettre en oeuvre Storm pour le traitement de flux de données.

### Connaissances préalables nécessaires:

Connaissance d'un langage de programmation comme Java ou Python.

### Programme:

- Introduction : Présentation de Storm:fonctionnalités, architecture, langages supportés  
Définitions:spout, bolt, topology
- Architecture : Etude des composants d'un cluster Storm :master node 'nimbus' et worker nodes  
Positionnement par rapport à un cluster Hadoop.Le modèle de données. Différents types de flux.
- Premiers pas : Configuration d'un environnement de développement.  
Installation d'un cluster Storm. Travaux pratiques sur le projet storm-starter
- Flux de données : Définition du nombre de flux dans un noeud, création de topologies regroupants des flux entre différents noeuds,  
communication entre flux en JSON, lecture de flux d'origines diverses (JMS, Kafka, ...)
- Haute disponibilité : Tolérance aux pannes: principe de fiabilisation des master node, workers node, nimbus  
Garantie de traitement des flux: principe,paramètres TOPOLOGY\_MESSAGE\_TIMEOUT\_SECS, TOPOLOGY\_ACKERS  
Traitements temps réel avec Trident. Scalabilité: parallélisme dans un cluster storm, ajouts de noeuds, commande 'storm rebalance'

# Programmation Scala

**Durée:** 3 jours  
1750 €

31 janvier au 2 février  
16 au 18 mai

29 au 31 août  
21 au 23 novembre

## Public:

Chefs de projet, data scientists, développeurs.

## Objectifs:

Comprendre les apports du langage Scala, de la programmation fonctionnelle. Maîtriser la programmation Scala, savoir s'interfacer avec des programmes Java.

## Connaissances préalables nécessaires:

Connaissance d'un langage de programmation et de la programmation objet.

## Programme:

- Introduction : Présentation de Scala. Les points forts du langage : extensibilité, programmation objet, programmation fonctionnelle, utilisation de la JVM.
  
- Premiers pas : Différents modes d'utilisation de Scala : compilé, en script, avec un interpréteur  
Outils de développement Scala : compilateur scalac, sbt (Scala's Build Tool), IntelliJ avec le plugin Scala  
Travaux pratiques: réalisation de programmes simples (calcul et affichage)
  
- Syntaxe : Les variables, les fonctions, les classes, les traits  
Le cas particulier des "singleton objects" et "companion objects"  
Les opérateurs. Les annotations.
  
- Programmation fonctionnelle : Principe et différences par rapport à la programmation impérative  
Particularités sur les tuples, listes, tables associatives
  
- Interfaçage avec Java : Fonctionnement de scala, byte code.  
Différences entre Java et Scala. Appel de classes Scala depuis du code Java  
Utilisation de bibliothèques Java dans un programme Scala



## Objets connectés:des OS embarqués au cloud

CB100

**Durée:** 1 jour  
660 €

13 février  
29 mai

4 septembre  
13 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: mqtt, 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.  
Mise en évidence avec AWS/IoT en mqtt sur websocket.

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 par vpn.  
Authentification et autorisations. Protection des données, législation

## Blockchain : principes et technologies

**Durée:** 1 jour  
660 €

12 février  
28 mai

10 septembre  
12 novembre

### Public:

Décideurs, architectes, chefs de projet et toute personne souhaitant comprendre le principe de la blockchain, les applications, et la mise en oeuvre.

### Objectifs:

Comprendre les principes, les apports, les mécanismes mis en oeuvre dans le cadre de la blockchain.

### Connaissances préalables nécessaires:

Connaissance générale des systèmes d'informations.

### Programme:

- Introduction : Principe, historique, notions de transactions, de blocs, de stockage distribué, de noeuds du réseau, de mineurs, exemples de blockchain, cas d'usage concrets et applications: crypto-monnaies, smart contracts, traçabilité, ...
- Cas du bitcoin : Fonctionnement, Description des transactions, comptes, wallet  
Principe du minage : assemblage des transactions en blocs, présentation de la structure des blocs et du calcul de l'empreinte  
Exemple de mise en oeuvre technique : outils matériels et logiciels pour configurer un noeud  
Démonstrations sur une plate-forme Linux
- Blockchain Ethereum : Plate-forme de smart-contracts, fonctionnement, outils, démonstrations de minage sur Linux  
Choix du mode de validation : preuve de travail ou preuve de participation
- Limites des blockchains : Problème de la consommation énergétique.  
Risque de prise de contrôle par un groupe de mineurs, Défauts de sécurité



## ElasticSearch : indexation

**Durée:** 1 jour  
660 €

5 mars  
18 juin

24 septembre  
10 décembre

### Public:

Chefs de projet, développeurs, architectes.

### Objectifs:

Comprendre le fonctionnement et les apports d'Elasticsearch dans le traitement de données.

### 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 :  
Watcher, Marvel, Kibana, X-Pack,  
Logstash, Beats  
Principe : base technique Lucene et apports d'ElasticSearch
- Installation de base : Prérequis techniques.  
Utilisation de l'interface d'administration Marvel
- Outils d'interrogation : Java API avec "Node client" et "Transport client"  
Autres clients : Perl, Python, Ruby, etc...  
Interface http, travaux pratiques, démonstration
- Traitement des données : Structure des données. stockage, indexation  
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.

# ElasticSearch : mise en oeuvre et programmation

**Durée:** 2 jours  
1220 €

6 au 7 mars  
19 au 20 juin

25 au 26 septembre  
11 au 12 décembre

## Public:

Architectes techniques, ingénieurs système, administrateurs..

## Objectifs:

Comprendre le fonctionnement et les apports d'Elasticsearch dans le traitement de données, et savoir le mettre en oeuvre, analyser les données, programmer des requêtes et créer des rapports et tableaux de bord avec kibana.

## Connaissances préalables nécessaires:

Connaissances générales des systèmes d'information, et des systèmes d'exploitation (Linux ou Windows). Les travaux pratiques sont réalisés sur Linux. Connaissance d'un langage de programmation structuré

## Programme:

- Introduction : Présentation ElasticSearch, fonctionnalités, licence  
Les différentes versions : fonctionnalités et particularités des versions de 2.0 à 5.0.  
Nouveautés de la version 6.0.  
Positionnement d'Elasticsearch et des produits complémentaires : Watcher, Marvel, Kibana, Logstash, Beats, X-Pack  
Principe : base technique Lucene et apports d'ElasticSearch  
Fonctionnement distribué
  
- Installation et configuration : Prérequis techniques.  
Utilisation de l'interface Marvel.  
Premiers pas dans la console Sense.
  
- Format et stockage des données : Format des données. Conversion au format JSON des données à traiter.  
Structure des données. Stockage, indexation.  
Terminologie Elasticsearch : notions de document, type, index  
Métadonnées : `_index`, `_type`, `_ID`  
Choix de l'identifiant par l'application avec l'API index, ou génération automatique d'un identifiant..  
Indexation inversée.



## ElasticSearch : mise en oeuvre et programmation

- Outils d'interrogation** : Java API avec "Node client" et "Transport client"  
API RESTful en HTTP  
Exemples de requêtes simples et plus complexes : recherche de «phrases», extraction de plusieurs documents, etc ..  
Notion de pertinence du résultat : «score»  
Requêtes avec Search Lite et avec Query DSL (domain-specific language)  
Utilisation de 'filtre' pour affiner des requêtes.  
Autres clients : Perl, Python, Ruby, etc...  
Aggrégation de résultats.
- Mises à jour** : Fonctionnement d'Elasticsearch pour les ajouts, modifications, suppression.  
Notion de version affectée par Elasticsearch.  
L'API bulk pour les traitements groupés.  
Réalisation de scripts avec groovy
- Gestion des accès concurrents** : Utilisation du numéro de version. Gestion par l'application : différentes méthodes selon les contraintes fonctionnelles.  
Utilisation d'un numéro de version externe.
- Kibana présentation** : Fonctionnalités : recherche, visualisation, création de tableaux de bord et graphiques à partir des données fournies par Elasticsearch
- Kibana, mise en oeuvre** : Installation, configuration du mapping avec Elasticsearch.  
Paramétrage dans le fichier kibana.yml  
Mapping automatique ou manuel. Configuration des indexes à explorer.  
Visualisation et sauvegarde de graphiques, étude des différents types de graphiques disponibles,  
création de tableaux de bord et rapports à partir des graphiques.

# ElasticSearch : infrastructure et administration

**Durée:** 2 jours  
1220 €

8 au 9 mars  
21 au 22 juin

27 au 28 septembre  
13 au 14 décembre

**Public:**  
Architectes techniques, ingénieurs système, administrateurs..

**Objectifs:**  
Comprendre le fonctionnement d'Elasticsearch, savoir l'installer et le configurer, gérer la sécurité avec X-Pack, et installer / configurer kibana pour le mapping sur les données Elasticsearch.

**Connaissances préalables nécessaires:**  
Connaissances générales des systèmes d'information, et des systèmes d'exploitation (Linux ou Windows). Les travaux pratiques sont réalisés sur Linux.

- Programme:**
- Introduction : Présentation Elasticsearch, fonctionnalités, licence  
Positionnement d'Elasticsearch et des produits complémentaires :  
Watcher, Kibana, Logstash, Beats, X-Pack  
Principe : base technique Lucene et apports d'ElasticSearch  
Fonctionnement distribué
  
  - Installation et configuration : Prérequis techniques.  
Installation depuis les RPM.  
Utilisation de l'interface X-Pack monitoring.  
Premiers pas dans la console Sense.  
Etude du fichier : elasticsearch.yml
  
  - Kibana : Présentation : objectifs, collecte de données, logs, ... par les APIs  
d'administration et de supervision ;  
Stockage dans elasticsearch et mise à disposition dans une interface  
web de graphiques  
Démonstrations.
  
  - Clustering : Définitions : cluster, noeud, sharding  
Nature distribuée d'elasticsearch  
Présentation des fonctionnalités : stockage distribué, calculs distribués  
avec Elasticsearch, tolérance aux pannes.



## ElasticSearch : infrastructure et administration

- Fonctionnement : Notion de noeud maître,  
stockage des documents : , shard primaire et réplicat,  
routage interne des requêtes.
- Gestion du cluster : Outils d'interrogation : `/_cluster/health`  
Création d'un index : définition des espaces de stockage (shard),  
allocation à un noeud  
Configuration de nouveaux noeuds : tolérance aux pannes  
matérielles et répartition du stockage
- Cas d'une panne : Fonctionnement en cas de perte d'un noeud :  
élection d'un nouveau noeud maître si nécessaire, déclaration de  
nouveaux shards primaires
- Mise en oeuvre X-  
Pack Security : Présentation des apports de X-Pack security: authentification, gestion  
des accès aux données (rôles), filtrage par adresse IP ;  
cryptage des données, contrôle des données;  
audit d'activité.
- Exploitation : Gestion des logs : `ES_HOME/logs`  
Paramétrage de différents niveaux de logs : INFO, DEBUG, TRACE  
Suivi des performances.  
Sauvegardes avec l'API snapshot.
- Evolutions : Les différentes versions : fonctionnalités et particularités des versions  
de 2.0 à 5.0.  
Nouveautés de la version 6.0.



## Filières bases de données SQL

Le Langage SQL p43

PostgreSQL Administration p47  
PostgreSQL Administration avancée p49

MySQL Administration p45

# Langage SQL

**Durée:** 3 jours  
1610 €

5 au 7 février  
23 au 25 avril  
25 au 27 juin

1er au 3 octobre  
10 au 12 décembre

## Public:

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

## Objectifs:

Maîtriser les fonctionnalités standards du langage SQL. Connaître et maîtriser les requêtes d'interrogation SQL.

## Connaissances préalables nécessaires:

Connaissance des principes des bases de données.

## Programme:

Introduction au langage SQL : Le modèle relationnel, Les composantes de SQL, les tables, la norme SQL. Le schéma général d'une base de données

Interrogations des données : La requêtes SELECT. Syntaxe générale. Sélection de lignes. L'agrégation. Le tri. La clause WHERE, les tris avec ORDER BY, les regroupements avec GROUP BY Les différents types de prédicats. Les expressions. Les fonctions. Les tables temporaires. La notion de jointure : syntaxe, Inner join, Outer join Les requêtes imbriquées : le Subselect simple, le Subselect corrélé Les opérateurs ANY, SOME, ALL, EXISTS

Le dictionnaire des données : La définition des objets : Data Definition Language Les types de données, la notion d'index, La création de tables CREATE TABLE, CREATE INDEX, l'intégrité référentielle Les VUES : création et utilisation

Mise à jour des données : Ajout, mise à jour ou suppression d'enregistrements avec INSERT, UPDATE, DELETE Modification ou suppression de tables avec ALTER et DROP

Les fonctions : Présentation des fonctions les plus courantes : numériques, de test, de gestion date/heure, de manipulation des chaînes de caractères.



## Langage SQL

- La confidentialité des données** : Gestion des droits d'accès, attribution et suppression de droits avec GRANT et REVOKE, utilisation des rôles pour sécuriser les accès
- Les contraintes d'intégrité** : Intégrité contrôlée par le SGBDR : valeurs par défaut, contrôle de la valeur nulle, de l'unicité d'une colonne : DEFAULT, NOT NUL, UNIQUE, CHECK, principe de la clé primaire et contrôle par le SGBDR, notion de FOREIGN KEY
- La transaction et les accès concurrents** : Principe des accès concurrents, solution des verrous, définition d'une transaction  
Mise en oeuvre des verrous, gestion des verrous en place sur une table  
Gestion des modifications : validation, retour à l'état antérieur, synchronisation avec COMMIT, ROLLBACK, SAVEPOINT
- L'optimisation** : Techniques d'optimisation des requêtes avec postgresSQL  
Phases d'exécution d'une requête. Analyse du plan d'exécution d'une requête EXPLAIN  
Bonnes pratiques et erreurs à éviter pour garantir de bonnes performances.

## MySQL : Exploitation

**Durée:** 3 jours  
1610 €

12 au 14 février  
14 au 16 mai

17 au 19 septembre  
14 au 16 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



## MySQL : Exploitation

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



## PostgreSQL : Administration

**Durée:** 3 jours  
1610 €

19 au 21 février  
14 au 16 mai

10 au 12 septembre  
26 au 28 novembre

### 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.



## PostgreSQL : Administration

Exploitation : Sauvegardes/Restaurations : 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.



## PostgreSQL :administration avancée

**Durée:** 2 jours  
1060 €

22 au 23 février  
17 au 18 mai

13 au 14 septembre  
29 et 30 novembre

### 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 : 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

Haute disponibilité : 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)

Optimisation : 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



## Filières Systèmes Unix/Linux

Linux/Unix Introduction	p51
Le Shell	p53
Shell avancé	p54
Administration Linux	p56
Administration avancée Linux	p61
Services réseaux sous Linux	p64

Préparation LPI101	p76
Préparation LPI102	p78
Préparation LPI201	p80
Préparation LPI201	p82

Haute disponibilité sous Linux	p66
Linux système sécurisé	p68
Linux sécurité des accès	p71
Linux optimisation et performances	p74

Virtualisation Linux	p84
Virtualisation avec Xen	p87
Virtualisation avec kvm	p89
Virtualisation avec lxc	p92

## Linux/unix introduction

**Durée:** 3 jours  
1590 €

22 au 24 janvier  
26 au 28 mars  
28 au 30 mai

27 au 29 août  
19 au 21 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".



## Linux/unix introduction

- 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
- 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  
1080 €

25 au 26 janvier  
29 au 30 mars  
31 mai au 1er juin

30 au 31 août  
22 au 23 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  
1630 €

26 au 28 février  
25 au 27 avril

25 au 27 juin  
17 au 19 septembre

### 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 : 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), getopt : décoder les options d'un script  
Programmation parallèle, programmation événementielle.  
Gestion des alertes. Le debugging et l'optimisation.

Programmation m4 : utilisation de la protection, encapsulation

Les outils : 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 : 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 Linux

**Durée:** 5 jours  
2370 €

5 au 9 février

9 au 13 avril

11 au 15 juin

10 au 14 septembre

26 au 30 novembre

### 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.



## Administration Linux

- Environnement graphique** : Présentation, gestionnaire de fenêtres. Différentes solutions : gnome, KDE, Windowmaker, xfce.
- Outils d'administration** : Webmin : Présentation, installation, configuration. Démonstration.
- 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...

## Administration Linux

- 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.
- Impressions** : 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.  
Installation d'une imprimante, modification d'un pilote : lpr, cups, printtool, system-config-printer.
- Programmation de tâches** : Le besoin, l'automatisation des tâches système.  
Exécution différée avec at. Programmation de tâches avec cron.  
Etude du fichier crontab. Les produits du marché : openPBS, fcron
- Exploitation** : Journaux : /var/log/messages
- 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.  
Présentation des utilitaires ssh, clients windows (Putty, WinSCP)
- NFS** : Fonctionnalités : partage de fichiers en réseau, avec gestion de la sécurité.  
Description du fonctionnement client/serveur.  
Etude du fichier /etc/exports.  
Travaux pratiques : configuration d'un serveur NFS sur chaque poste, et configuration des clients NFS pour tester les accès.



## Administration Linux

Intégration système : Samba : Principe.  
d'information

Intégration de SMB au niveau des couches réseaux.

Fonctionnalités : partage de répertoires, d'imprimantes, création de comptes....

Travaux pratiques : Installation et configuration de samba pour le partage de fichiers.



## Administration avancée Linux

**Durée:** 5 jours  
2410 €

19 au 23 février

16 au 20 avril

18 au 22 juin

24 au 28 septembre

3 au 7 décembre

### 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.



## Administration avancée Linux

Systèmes de fichiers journalisés: Exemples de systèmes de fichiers journalisés.  
journalisés

Les types de journalisation.

XFS : fonctionnement, mise en oeuvre, administration  
compatibilité NFS

Ext3, ext4 : caractéristiques et mise en oeuvre.

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 madm 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.



## 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  
Tcpcmdump  
Outils de diagnostic
- Les services : Serveurs de configuration  
dns, dhcp, bootp (présentation de kickstart  
Serveurs de fichiers : nfs, ftp, tftp, http  
Serveurs d'accès : routage, firewall, proxy
- DHCP : Définition, principe.  
Configuration poste client, serveur.  
Notion de bail.  
configuration avancée.  
Redondance DHCP.
- DNS : Définition, fonctionnement.  
Travaux pratiques avec bind : configuration client et serveur.

## Les services réseaux Linux

- FTP** : File transfer protocol  
Travaux pratiques : mise en oeuvre de ftp: configuration, droits sur les répertoires, gestion des utilisateurs, surveillance, fichiers de logs
- NTP** : Définition.Fonctionnement.  
Déclaration d'un point de synchronisation.  
Configuration d'un serveur d'horloge. Configuration de clients.  
Architecture. Contrôles d'accès.  
Implémentations de NTP.
- Messagerie** : Les protocoles, POP3 et IMAP4  
Structure des messages.  
Présentation de Postfix, installation et configuration.  
Fichiers master.cf, main.cf
- 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  
1660 €

12 au 14 février  
22 au 24 mai  
27 au 29 août

5 au 7 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** : Le besoin : pourquoi la haute disponibilité, mesure de la disponibilité.  
Quelques définitions : tolérance aux pannes, fail-over, RAID, Mirroring, redondance, MTBF, etc ...  
Les acteurs du marché, positionnement de Linux.  
Présentation de l'architecture LVS.  
Les solutions de haute disponibilité.
- Clustering** : Les différentes fonctions de clustering :  
répartition des accès disques, répartition de la charge CPU,  
basculement automatique ou programmé sur un autre processeur,  
exécution simultanée sur plusieurs processeurs.
- Adresses réseaux** : Principe du basculement d'adresses.  
Solution avec Fake.  
Agrégation d'interfaces réseau.  
Travaux pratiques :  
mise en place de l'agrégation avec deux cartes réseaux ethernet.  
Configuration dynamique et configuration statique.  
Test et vérification dans les fichiers journaux.

## Haute disponibilité Linux

Linux Virtual Server : Architecture : pacemaker, ldirector, heartbeat, fake, coda  
Utilisation de mon pour la détection des services défaillants.  
Travaux pratiques :  
Installation, configuration de heartbeat et ldirectord  
Configuration de Pacemaker pour la gestion du cluster.

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.

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

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  
1660 €

19 au 21 février  
2 au 4 mai

17 au 19 septembre  
12 au 14 novembre

### 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  
Chiffrement des disques durs :  
mise en oeuvre de LUKS et dm-crypt  
Protection de la mémoire :  
principe et outils de sécurisation.

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  
1660 €

30 janvier au 1er février  
28 au 30 mai

24 au 26 septembre  
19 au 21 novembre

**Public:**

Toute personne souhaitant sécuriser les accès à un système Linux

**Objectifs:**

Savoir configurer les mécanismes de sécurité réseau de Linux.

**Connaissances préalables nécessaires:**

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

**Programme:**

- Introduction : Le besoin, définition du D.I.C.  
Les attaques possibles.  
Evaluation des risques.  
Méthodes de protection.
  
- Les ports de niveaux 5 : Rappels sur la notion de port.  
Les ports UDP et les ports liés au réseau.  
Exemples de trames.
  
- Outils de captures réseau : Les analyseurs de trames : tcpdump, wireshark.  
Travaux pratiques :  
mise en 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.

## Linux sécurité des accès

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  
1170 €

8 et 9 mars  
7 et 8 juin

3 et 4 septembre  
15 et 16 novembre

### 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.

## Linux : optimisation performances métrologie

- 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 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 : etx2/3/4, reseatfs, 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.

# Administration Ubuntu Server

- 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'opnSSH pour des transferts de fichiers. Génération de clés. Création et utilisation de tunnels.
  
- LVM** : Principe. Organisation du Logical Volume Manager. Gestion du LVM : les volumes physiques et logiques, les snapshots, les partitions. Travaux pratiques : Céation de volumes physiques, logiques, ajout d'un disque, export/import d'un disque. Combinaison de groupes de volumes.
  
- RAID** : Définitions : les différents niveaux de RAID. Description du raid logiciel sur Linux. Travaux pratiques : Configuration et utilisation des différents raids logiciels.
  
- Utilisateurs** : 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.
  
- Processus** : Les processus. Les threads. Gestion des priorités. Utilisation des pseudo-processus /proc: stat, cpuinfo, ...



## Administration Ubuntu Server

- Sauvegardes** : 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.
- Impressions** : 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.
- 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

## Preparation certification LPI101

**Durée:** 3 jours  
1680 €

26 au 28 février  
2 au 4 mai

5 au 7 septembre  
5 au 7 novembre

### Public:

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

### Objectifs:

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

### Connaissances préalables nécessaires:

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

### Programme:

- Architecture système : Déterminer et configurer les paramètres du matériel.  
Démarrage du système.  
Changer de niveau d'exécution et arrêter ou redémarrer le système.
- Installation de Linux et gestion de paquetages : Concevoir un schéma de partitionnement du disque dur  
Installation d'un gestionnaire de démarrage  
Gestion des bibliothèques partagées  
Utilisation du gestionnaire de paquetages Debian et des gestionnaires de paquetages RPM et YUM
- Commandes GNU et Unix : Travail en ligne de commande  
Traitement de flux de type texte par des filtres  
Effectuer une gestion de base sur les fichiers  
Utilisation des flux, des tubes (pipes) et des redirections  
Création, surveillance et destruction de processus  
Modification des priorités des processus  
Recherche dans des fichiers texte avec les expressions rationnelles  
Edition de fichiers texte avec vi



## Preparation certification LPI101

Disques : Système de fichiers Linux, Arborescence de fichiers standard.  
Création de partitions et systèmes de fichiers  
Maintenir l'intégrité des systèmes de fichiers  
Contrôle du montage et du démontage des systèmes de fichiers  
Gestion des quotas de disque  
Gérer les permissions et les propriétaires des fichiers  
Créer et changer les liens symboliques et physiques sur les fichiers  
Recherche de fichiers et placement des fichiers aux endroits adéquats

## Preparation certification LPI102

**Durée:** 3 jours  
1680 €

5 au 7 mars  
4 au 6 juin

17 au 19 septembre  
19 au 21 novembre

### Public:

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

### Objectifs:

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

### Connaissances préalables nécessaires:

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

### Programme:

Shells, scripts et gestion de données : Personnalisation et utilisation de l'environnement du shell  
Personnalisation ou écriture des scripts simples  
Gestion de données avec le langage SQL

Interface utilisateur et environnement graphique : Installation et configuration de X11. Configurer un gestionnaire d'affichage (Display Manager)  
Accessibilité.

Tâches d'administration : Gestion des comptes utilisateurs et des groupes ainsi que des fichiers systèmes concernés.  
Automatisation des tâches d'administration par la planification.  
Paramètres régionaux et langues.

Services système essentiels : Gestion de l'horloge système. Journaux systèmes.  
Configuration de base de l'agent de transfert de courrier (MTA).  
Gestion de l'impression et des imprimantes.



## Preparation certification LPI102

- Notions élémentaires sur les réseaux : Notions élémentaires sur les protocoles Internet. Configuration réseau de base.  
Résolution de problèmes réseau. Configuration DNS sur les postes de travail.
- Sécurité : Tâches d'administration de sécurité. Configuration de la sécurité du système.  
Sécurisation des données avec le chiffrement.

## Préparation certification LPI201

**Durée:** 3 jours  
1680 €

21 au 23 mars  
18 au 20 juin

24 au 26 septembre  
3 au 5 décembre

### Public:

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

### Objectifs:

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

### Connaissances préalables nécessaires:

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

### Programme:

Noyau Linux : Composants du noyau.  
Compilation, mise à jour d'un noyau.  
Personnalisation, construction et installation d'un noyau et des modules.  
Gestion/interrogation des modules.

Démarrage du système : Personnalisation du démarrage système et séquence de boot.  
Restauration du système.

Systèmes de fichiers et périphériques : Intervention sur le système de fichiers Linux.  
Maintenance du système de fichiers.  
Création et configuration des options du système de fichiers.  
Gestion de périphérique udev.

Administration avancée de périphérique de sauvegarde : Configuration RAID.  
Ajustement des accès au périphérique de sauvegarde.  
Gestionnaire de volumes logiques.



## Préparation certification LPI201

- Configuration réseau : Configuration élémentaire, et avancée d'un réseau et dépannage.  
Identification et correction des problèmes réseau.  
Notification des utilisateurs concernant les problèmes relatifs au système.
- Maintenance système : Construction et installation de programmes à partir du code source.  
Opérations de sauvegarde.
- Serveur de nom de domaine : Configuration élémentaire d'un DNS.  
Création et maintenance de zones DNS.  
Sécurisation d'un serveur DNS.

## Preparation certification LPI202

**Durée:** 3 jours  
1680 €

28 au 30 mars  
25 au 27 juin

1er au 3 octobre  
10 au 12 décembre

### Public:

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

### Objectifs:

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

### Connaissances préalables nécessaires:

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

### Programme:

Services web : Mise en place et maintenance d'un serveur web.  
Mise en place d'un serveur mandataire.

Partage de fichiers. : Configuration du serveur Samba, d'un serveur NFS.

Gestion des clients réseau : Configuration DHCP. Authentification PAM. Usage de client LDAP.

Services e-mail : Utilisation des serveurs e-mail.  
Gestion de la livraison locale et distante des e-mail.

Sécurité système : Configuration d'un routeur. Sécurisation des serveurs FTP. Shell sécurisé (SSH). TCP Wrapper. Tâches de sécurité.

Dépannage : Identification des étapes de démarrage et dépannage des chargeurs de démarrage.  
Dépannage général. Dépannage des ressources système, et des configurations liées à l'environnement.



# Administration AIX

**Durée:** 5 jours

Prix et dates: nous consulter

## 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.

## Administration AIX

- Périphériques : le répertoire /dev
- Impressions : Fonctionnement. Installation. Imprimante réseau.  
Les commandes d'impression, la gestion des travaux, des files d'attente, d'un serveur d'impression : Ajout d'imprimantes, configuration.  
Suivi du spooler, /etc/qconfig, lpq/lpr
- Pagination : ajout d'un espace de pagination. Modification dynamique d'un espace de pagination. Activation d'un espace de pagination
- Réseau : Les différents paramètres à prendre en compte. Configuration de l'interface. Configuration du nom de l'adresse du système
- 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.

## Administration AIX avancée

- 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.
- Accès : 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.  
Autorisation de connexion. Masque applicatif. Localisation du compte. Accès à su. Shell sécurisé/base TCB
- Performances : La mesure des performances : les temps de réponse. Les commandes de base : sar, vmstat, iostat. Le démon sadc.  
Présentation des ressources et impact sur les performances : la mémoire, l'espace disque, les accès disques, la cpu.  
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.  
Utilisation du Performance Diagnostic Tool. Structure du PDT, installation et activation. Configuration avec pdt\_config.
- CPU : 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.
- Mémoire : Utilisation de la commande vmstat. Les options -l et -s. Mesure de la mémoire utilisée: svmon.  
Analyse de la charge mémoire VVM avec schedtune et vmtune.
- Disques : Commandes de base: df, du. Mesure avec acctmerge. La commande dodisk et le fichier dacct.
- Le diagnostic : Fichiers de logs : syslog, analyse par errpt
- Traces : Utilisation de traces systèmes. Création d'un fichier trace. Exploitation avec trcrpt, filters.

# Administration Solaris 11

**Durée:** 5 jours

Prix et dates: nous consulter

## 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.<br>Introduction à Solaris.<br>Principaux apports de Solaris 11.  |
| Installation    | : Installation de Solaris à partir d'un dvd ou du réseau.<br>Ajout de logiciel : gestion des packages et des dépôts.<br>Autoconfiguration du système : principe, périphériques                        |
| Démarrage/arrêt | : Etapes de lancement, procédure de boot<br>Niveaux de boot, fichier /etc/inittab<br>Montages au boot : /etc/vfstab<br>Arrêt du système : halt, reboot, shutdown<br>Configuration matérielle          |
| SMF             | : Service Management Facility : Présentation, utilisation.<br>Gestion des services.   |
| Utilisateurs    | : Fichier /etc/passwd<br>Principe des shadow passwords<br>Création de comptes utilisateurs, valeurs par défaut<br>Fichiers profils , choix d'un shell : .cshrc, .profile,<br>gestion des roles (RBAC) |

## Administration Solaris 11

- Fichiers : Les inodes. Gestion des partitions.  
Les différents systèmes de fichiers, le système UFS, le système ZFS.  
Création d'un système ZFS  
Le montage du système de fichiers.  
Montage automatique, mount/umount.  
Les pseudo-systèmes de fichiers  
les ACLs sous Solaris
- Sauvegardes : Définitions, outils standards : tar, cpio, dd  
Commandes ufsdump/ufsrestore snapshots ZFS
- Ordonnancement : At, cron, fichier crontab
- 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.

# Administration avancée Solaris

- Configuration réseau avancée : Configuration ip :  
nwadm, ipadm, dladm  
Aggrégation de liens avec IPMP. groupes, le démon in.mpathd, impstat.  
Architecture, mécanismes de basculement.  
Aggrégation 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.
- 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



# 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.

## Systèmes Linux embarqués

- Noyau** : Compilation d'un noyau.  
Travaux pratiques : compilation d'un noyau pour carte 6410 (ARMv11).  
Optimisation de l'empreinte mémoire. Gestion de l'initrd.
- Système racine** : Organisation du système de fichiers: linux, rootfs, nvram, rootfs\_data.  
Systèmes de fichiers embarqués. Présentation de jffs2, yaffs2. Autres types de systèmes de fichiers.  
Services embarqués : dropbear, dnsmasq, busybox.  
Compilation d'une busybox. Présentation, ajouts de fonctionnalités, limitations.
- Développement** : Ajout de modules au noyau.  
Travaux pratiques : Portage de netsnmp.  
Compilation d'une MIB spécifique au matériel et intégration dans la distribution embarquée.  
Mise en 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.



## Filières virtualisation, cloud et orchestration

Virtualisation Linux	p84	
Virtualisation avec Xen	p87	
Virtualisation avec kvm		p89
Virtualisation avec lxc	p92	

Docker, mise en oeuvre	p95	
Docker, administration avancée	p98	
Kubernetes, optimisation conteneurs	p101	
Réseaux virtuels avec OpenVswitch	p100	

Cloud, technologies et enjeux		p103
Architecture cloud d'entreprise	p105	
OpenStack, administration	p107	
Cloud d'entreprise avec Nebula	p110	

AWS, architecture	p121	
AWS, stockage S3, dynamoDB	p123	
AWS, les APIs EC2 et S3	p125	

Cobbler, gestion de configuration	p112	
Ansible, industrialiser les déploiements	p113	
Puppet, administration centralisée	p114	
Puppet, expertise	p117	
Gestion de configuration avec chef	p119	

# Virtualisation Linux

**Durée:** 2 jours

Prix et dates: nous consulter

**Public:**

Chefs de projet, administrateurs souhaitant mettre en oeuvre une solution de virtualisation Linux.

**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 de virtualisation possibles sur Linux : 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.
  
- VirtualBox : Principe et caractéristiques du produit.  
Les différentes éditions.  
Travaux pratiques :  
installation VirtualBox.  
Création de machines virtuelles.  
Différents paramètres de configuration.  
Configuration des machines virtuelles en XML.



## Virtualisation Linux

- lxc** : Présentation des Linux Containers  
 Objectifs du projet  
 Isolation et contrôle des ressources.  
 Principe des 'cgroup' et création de containers  
 Travaux pratiques :  
 activation des cgroup, installation lxc.  
 Utilisation de lxc-checkconfig.  
 Configuration de containers  
 Exemple de Busybox
- QEMU et kvm** : Principe de QEMU et architecture  
 Travaux pratiques : installation et lancement d'une image  
 Etude des options de lancement de qemu.  
 Consoles des machines virtuelles :  
 graphiques (console VNC, Spice, ..) ou consoles en mode texte.  
 Kernel Based Virtual Machine : positionnement par rapport aux autres systèmes de virtualisation, et par rapport à QEMU  
 Travaux pratiques :  
 installation avec un noyau contenant les modules kvm  
 Gestion des images :  
 création d'images, différents supports possibles, options de lancement  
 Travaux pratiques :  
 commandes info, check, resize, convert  
 Gestion du matériel:  
 architectures supportées, processeurs, mémoire,  
 périphériques de stockage, audio, video, usb, bluetooth, etc ...  
 Configuration du réseau:  
 différents modes possibles (user, tap, bridge,...)  
 Travaux pratiques :  
 configuration réseau sur les images créées aux chapitres précédents  
 Snapshots et migrations :  
 principe de fonctionnement  
 Mise en oeuvre et options de la commande 'migrate'
- Administration avec libvirt** : Présentation de l'API libvirt et des fonctionnalités apportées,  
 Virtual Machine Manager  
 Travaux pratiques :  
 installation de libvirt et lancement de virt-manager

## 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.

## Virtualisation avec Xen

Domaines utilisateurs	: Gestion des systèmes invités : le service xend La commande d'administration xm (xm create, xm liste...) l'accès à xend par l'interface web. Travaux pratiques : ajout de systèmes invités avec xm create. Utilisation d'un fichier de description de machines virtuelles. Configuration de domU. Arrêt et démarrage de systèmes invités avec xm shutdown et xm reboot.
Supports de stockage	: Déclaration des espaces de stockage accessibles au domU : périphériques blocs, partitions physiques, ou volumes logiques.
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  
1160 €

5 au 6 mars  
3 au 4 mai

30 au 31 août  
8 au 9 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** : 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 de virtualisation sur Linux.  
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.

**Présentation QEMU:** Principe de QEMU et architecture  
Deux modes de fonctionnement : code compilé pour un processeur, émulation d'une machine physique.  
Travaux pratiques : installation et lancement d'une image  
Etude des options de lancement de qemu.  
Consoles des machines virtuelles :  
graphiques (console VNC, Spice, ..) ou consoles en mode texte.

## Virtualisation avec KVM

Installation de kvm : 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.

Travaux pratiques : : installation avec un noyau contenant les modules kvm  
 Gestion des images :  
 création d'images, différents supports possibles, options de lancement  
 Travaux pratiques :  
 commandes info, check, resize, convert  
 Gestion du matériel:  
 architectures supportées, processeurs, mémoire,  
 périphériques de stockage, audio, video, usb, bluetooth, etc ...  
 Configuration du réseau:  
 différents modes possibles (user, tap, bridge,...)  
 Travaux pratiques :  
 configuration réseau sur les images créées aux chapitres précédents

Migration d'images : Le besoin.  
 Sauvegarde/chargement de machines virtuelles :  
 à l'arrêt ou en fonctionnement  
 Limites par rapport aux processeurs  
 Snapshots et migrations :  
 principe de fonctionnement  
 Mise en oeuvre et options de 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.

## Virtualisation avec lxc

**Durée:** 2 jours

Prix et dates: nous consulter

### **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** : 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 de virtualisation possibles sur Linux  
conteneurs d'application, noyaux secondaires, machines virtuelles, hyperviseur, virtualisation matérielle...
- Présentation de lxc** : Linux containers, historique, principe de fonctionnement.  
L'isolation de ressources, création d'un environnement utilisateur.  
Positionnement par rapport aux autres solutions de virtualisation.
- Cgroup** : Fonctionnement de Control Group.  
Travaux pratiques :  
Vérification de la configuration du noyau.  
Activation des Cgroups.



## Virtualisation avec lxc

- Les outils LXC** : Site de référence pour le téléchargement.  
 Installation de LXC par rpm, urpmi, yum ou apt-get install.  
 Présentation des différents outils pour :  
 vérifier la configuration du noyau, créer, détruire, gérer les conteneurs, et les tâches associées  
 lxc-checkconfig, lxc-console, lxc-create, lxc-start, lxc-stop etc...  
 Travaux pratiques :  
 vérification de la configuration avec lxc-checkconfig
- Gestion des conteneurs** : Configuration, création, démarrage.  
 Travaux pratiques :  
 étude des exemples de configuration dans /share/doc/lxc/examples  
 Utilisation des templates pour créer des containers standards.  
 Exemple de busybox  
 Choix des systèmes de fichiers.
- Configuration du réseau** : Les différentes méthodes : interface physique, pont/commutateur virtuel, vlan.  
 Exercices pratiques :  
 Mise en oeuvre. Configuration d'un point par brctl.
- Exploitation** : Description des ressources à administrer:  
 répertoires du produit lxc  
 fichiers de configuration des containers  
 systèmes de fichiers des containers  
 Interventions possibles : visualisation des connexions réseau d'un container,  
 des processus s'exécutant dans un lxc, ..  
 Travaux pratiques :  
 exemples du fichier fstab permettant de configurer les partitions du container  
 Commandes lxc d'arrêt/relance, de vérification des containers et de supervision  
 Exercices avec lxc-console, lxc-monitor, lxc-nostat, lxc-execute, ...
- Administration avec libvirt** : Présentation de l'API libvirt et des fonctionnalités apportées, Virtual Machine Manager  
 Travaux pratiques :  
 installation de libvirt et lancement de virt-manager  
 lxc avec libvirt : gestion des containers,  
 création d'images, configuration du réseau, du stockage virtuel, de la mémoire.

## Docker : mise en oeuvre

**Durée:** 2 jours  
1110 € HT

12 au 13 mars  
22 au 23 mai

18 au 19 septembre  
26 au 27 novembre

### 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 de l'écosystème pour déployer des images tout en intégrant les contraintes de production.

### 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, déploiements rapides, coexistence de plusieurs versions d'une application sur un même serveur.  
 Les différentes éditions et leurs fonctionnalités : Docker Enterprise Edition, Docker Community Edition, Docker Cloud
- Principe et architecture : 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 conteneurs Dockers.  
 Plate-formes supportées.  
 L'écosystème Docker : Docker Machine, Docker Compose, Kitematic, Docker Swarm, Docker Registry



## Docker : mise en oeuvre

- Installation et configuration** : Prérequis techniques.  
 Travaux pratiques : installation sur Linux  
 Mise en oeuvre des scripts fournis par Docker pour l'installation.  
 Création d'un groupe Docker.  
 Mise en oeuvre en ligne de commande.  
 Démarrage d'un container simple.  
 Configuration de Docker et des containers.  
 Travaux pratiques :  
 démarrage automatique des containers, contrôle avec systemd,  
 limitation des ressources.
- Gestion des images et des conteneurs** : Création de nouvelles images.  
 Principe des DockerFile.  
 Travaux pratiques :  
 Utilisation de DockerFile pour créer des images personnalisées:  
 principales instructions (RUN, FROM, ENV, EXPOSE, etc ...)  
 Recommandations et bonne pratiques d'écriture de DockerFile.  
 Gestion des conteneurs : création, affichage, sauvegarde de l'état  
 Exemple de déploiement d'une application web avec des containers.  
 Présentation du Docker Hub.  
 Publication d'images vers un registry.
- Volumes de données** : Initialisation des volumes de données lors de la création d'un container.  
 Travaux pratiques:  
 ajout de volumes de données, contrôle avec la commande Docker inspect;  
 Sauvegarde, migration, restauration de volumes  
 Création de conteneurs de volumes de données
- Administration** : Présentation des outils Swarm, Compose, Docker Machine  
 Fonctionnalités de swarm : cluster Docker, principe du mode swarm,  
 load balancing.  
 Démonstrations de load balancing.  
 Applications de multi-containers avec Compose  
 Méthode d'administration des containers en production.  
 Orchestration avec Docker Machine  
 Configuration réseau et sécurité dans Docker  
 Présentation des plugins Docker

# Réseaux virtuels avec OpenvSwitch

**Durée:** 2 jours  
1 180 € HT

22 au 23 mars  
7 au 8 juin

11 au 12 octobre  
20 au 21 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  
650 € HT

22 janvier

3 avril

2 juillet

15 octobre

### 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 : technologies et enjeux

Cloud d'entreprise : Comment s'organise une infrastructure d'entreprise en cloud ?

Comment migrer les services reseau : dns, dhcp, smtp, ...

Organisation des machines virtuelles en sous-réseaux, switches et cables 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  
1710 €

23 au 25 janvier  
4 au 6 avril

3 au 5 juillet  
16 au 18 octobre

### 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 webservice ; 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 : 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".
- Mise en oeuvre : 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.
- configuration d'un cloud privé : 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.
- La sécurité : Evaluation des risques, présentation des outils et solutions  
 Aspects juridiques, protection des données,  
 traçabilité, chiffrement, génération de clés...

## OpenStack : configuration et administration

**Durée:** 3 jours  
1780 €

19 au 21 mars  
4 au 6 juin

8 au 9 octobre  
17 au 19 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. Savoir gérer des machines virtuelles avec OpenStack, configurer le stockage virtuel et les réseaux virtuels.

### 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, réseaux virtuels,  
compatibilité Amazon EC2, EBS, S3, ...  
Possibilité de créer des clouds privés ou des clouds hybrides avec AWS.
- Caractéristiques techniques** : Prérequis : plate-formes Linux,  
Hyperviseurs et systèmes de stockage supportés.  
Architecture d'OpenStack : notion de services et de projets  
Présentation des principaux projets, et de leurs rôles :  
Nova, Cinder, Glance, Swift, Neutron, Keystone, Horizon  
Principe d'interrogation des services : accès par des clients webservice.  
Etude de l'architecture réseau d'OpenStack :  
cloud controller, noeuds avec stockage et exécution d'instances virtuelles, serveur de monitoring  
Outils et interfaces de gestion :  
le Dashboard, l'interface de gestion, pilotage, suivi.

## OpenStack : configuration et administration

- Installation et configuration** : Prérequis matériel et logiciel.  
 Etude des phases de l'installation et des composants à installer :  
 authentification avec Keystone, serveur d'images Glance,  
 stockage d'objets avec Swift,  
 serveur de calcul Nova,  
 services auxiliaires : dashboard, FlatDHCP, MySQL ou PostgreSQL  
 Présentation de packstack.  
 Définition des tenants.  
 Préparation de l'installation.  
 Travaux pratiques :  
 installation à partir de scripts,  
 identification des différents types de machines virtuelles disponibles,  
 vérification de l'état des services,  
 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.
- Utilisation de la console Web** : Présentation des fonctionnalités offertes par la console.  
 Notions de projets.  
 Travaux pratiques :  
 création de nouveaux projets, d'utilisateurs,  
 génération des clés pour la connexion aux instances,  
 création d'images et lancement des instances,  
 test de différents moyens d'accès : par une console VNC, par ssh
- Extension** : Mise en pratique :  
 Ajout d'un noeud de calcul. Configuration du nouveau noeud.  
 Visualisation de la capacité supplémentaire.  
 Migration d'un noeud de calcul.  
 Passage à l'échelle.  
 Les mises à jour.  
 Analyse des problèmes.
- Gestion des volumes** : Présentation de Cinder. Architecture, locale, NAS, SAN.  
 Travaux pratiques:  
 Mise en oeuvre : démarrage du service, installation de volumes iScsi.  
 Manipulation de disques virtuels : création, attachement, formatage,  
 suppression.  
 Analyse de stratégies pour le stockage.



## OpenStack : configuration et administration

- Gestion du réseau** : Principe : virtualisation des éléments d'un réseau physique : ponts, commutateurs, adressage, routage, répartition de charge, pare-feux, ..  
 Définitions et terminologie : adresses mobiles, adresses fixes, Virtual network, physical network, flat network, provider network.  
 Les modes réseaux : Flat, FlatDHCP, VLAN.  
 Présentation de Neutron : fonctionnalités, architecture.  
 Travaux pratiques :  
 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 des adresses.  
 Scripts SDN (Software Defined Network).  
 Définition d'une architecture globale.
- Les utilitaires EC2 Tools** : Principe de fonctionnement des EC2 tools.  
 Travaux pratiques :  
 téléchargement des identifiants de connexion,  
 gestion et installation d'images,  
 lancement de nouvelles instances, ..  
 Utilisation de la compatibilité Amazon EC2
- Exploitation d'une infrastructure OpenStack en production** : Présentation des fonctionnalités disponibles en ligne de commande avec Nova.  
 Travaux pratiques :  
 gestion de services, des journaux, ajouts de noeuds, configuration, migration de machines virtuelles d'un noeud de calcul vers un autre

## CloudStack : mise en oeuvre

**Durée:** 2 jours

Prix et dates: nous consulter

### **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...



## CloudStack : mise en oeuvre

- Installation et configuration : Prérequis matériel et logiciel pour le CloudStack Management Server et pour l'hyperviseur.  
Travaux pratiques : installation du Management Server, de la base MySQL et du partage NFS.  
Choix et installation d'un hyperviseur.  
Configuration d'une infrastructure cloud par la console web : création de zones, pods, clusters, hôtes, et stockage primaire et secondaire.  
Initialisation et création des premières instances du cloud.
- Administration du cloud : Gestion des utilisateurs.  
Configuration du réseau.  
Exploitation des machines virtuelles : création, migration, suppression,  
Utilisation de projets pour regrouper des utilisateurs partageant les mêmes ressources.  
Mise en oeuvre des templates.

# Cloud d'entreprise avec OpenNebula

**Durée:** 3 jours  
1780 €

du 26 au 28 février  
du 2 au 4 mai

du 29 au 31 août  
du 22 au 24 octobre

**Public:**

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

**Objectifs:**

Savoir installer OpenNebula, le configurer et l'utilisation pour le déploiement de machines virtuelles. Tous les concepts abordés dans cette formation sont illustrés dans de nombreux travaux pratiques.

**Connaissances préalables nécessaires:**

Connaissance de l'administration des systèmes Linux et réseaux IP.

**Programme:**

Introduction : Fonctionnalités : gestion de machines virtuelles, d'images, de réseaux virtuels et de stockage  
Historique projet OpenNebula, écosystème, support OpenNebula Systems.

Caractéristiques techniques : Hyperviseurs supportés : xen, kvm, VMware. Notion d'instance OpenNebula et de VDC 'Virtual Data Centers'. Fédérations.  
Compatibilité EC2. Cloud-bursting.  
Présentation des différentes APIs disponibles : Ruby, Java, XMLRPC  
Architecture OpenNebula : management daemon et scheduler  
Haute disponibilité et outils supervision.

Installation et configuration : Prérequis matériel et logiciel.  
Installation depuis les packages et démarrage : partie frontend, et noeuds.  
Gestion des utilisateurs et accès de base.  
Interface d'administration : les différentes méthodes : ligne de commande ou interface Sunstone.  
Sécurité : gestion des utilisateurs, groupes et droits d'accès aux ressources.  
Lien vers ldap et Active Directory.



## Cloud d'entreprise avec OpenNebula

- Stockage** : Différents types de stockage : filesystem, LVM, Ceph et vmdk.  
Manipulation de disques virtuels : création, attachement, formatage, suppression.  
Stratégie pour le stockage. Gestion des images virtuelles.
- Gestion des machines virtuelles** : Hyperviseurs. Gestion de modèles. Gestion des machines : création, cycle de vie, migration.  
Modifications des caractéristiques. Elasticité.  
Groupement de machines virtuelles. Notion de service. Mise en oeuvre de OneFlow.  
Suivi des consommations.
- Gestion du réseau** : Notion de réseau de service. Réseaux virtuels, routeurs virtuels. Mise en oeuvre.  
Différents types de drivers associés à chaque hôte  
Intégration avec Etables, Vlan, Vxlan, OpenvSwitch

# Cobbler : gestion de configurations

**Durée:** 2 jours  
1215 €

18 au 19 janvier  
22 au 23 mai

27 au 28 août  
8 au 9 novembre

**Public:**

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

**Objectifs:**

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

**Connaissances préalables nécessaires:**

Il est demandé aux participants de connaître l'administration des systèmes Linux et un langage de développement de scripts.

**Programme:**

- Introduction : Présentation de Cobbler, fonctionnalités.  
Gestion automatisée en réseau des installations de systèmes Linux : serveurs, machines virtuelles, containers.  
License, version et utilisations classiques de cobbler dans le monde Linux.
- PXE et kickstart : Principe de fonctionnement de PXE et étude de l'installation automatisée de Linux CentOS avec kickstart.
- Installation cobbler : Installation avec yum et configuration dans le fichier /etc/cobbler/settings.  
Démarrage du service cobbler. Premiers déploiements simples : import d'une distribution et création d'un système à déployer. Présentation de l'interface client de cobbler.
- Les primitives : Modélisation d'une distribution avec Cobbler. Les objets, les règles standards, l'héritage.
- Les commandes : Les commandes d'interrogation des objets : list, report, remove, copy, find, etc ...  
Administration : Check, sync, import, buil. Recherche en ligne de commande : dsitro, profile, system, repo , 'find'  
Présentation de l'interface web.





# Ansible : industrialiser les déploiements

**Durée:** 2 jours  
1215 €

1er et 2 février  
31 mai et 1er juin

20 et 21 septembre  
13 et 14 décembre

**Public:**

Administrateurs, exploitants souhaitant mettre en oeuvre Ansible pour le déploiement des services et applications.

**Objectifs:**

Comprendre le fonctionnement d'Ansible, savoir rédiger des scripts de déploiement

**Connaissances préalables nécessaires:**

Il est demandé aux participants de connaître l'administration des systèmes Linux et un langage de développement de scripts.

**Programme:**

Introduction : Gestion automatisée de l'infrastructure systèmes et réseaux (serveurs, machines virtuelles,containers, équipements réseaux)  
Notion de playbooks.  
Langage de configuration, déploiement, orchestration  
Commandes Ad-Hoc

Installation et : Travaux pratiques sur CentOS.  
inventaire avec Configuration de la connexion avec les serveurs distants : ssh  
Ansible Premiers pas avec ansible :  
inventaire des serveurs accessibles : hôtes, groupes, etc ...  
Configuration d'un inventaire automatique.

playbooks et : Exemples de playbooks  
modules Les modules fournis avec Ansible, écriture de nouveaux modules

commandes Ad : commandes shell, gestion du parallélisme,  
Hoc transfert de fichiers,  
gestion des utilisateurs et groupes,  
déploiement à partir des sources,  
administration des services.



## Puppet : administration centralisée

**Durée:** 3 jours  
1610 €

26 au 28 février  
11 au 13 juin

1er au 3 octobre  
10 au 12 décembre

### Public:

Administrateurs, exploitants souhaitant utiliser Puppet pour la gestion centralisée des configurations logicielles de multiples serveurs ou machines virtuelles.

### Objectifs:

Comprendre le fonctionnement de Puppet, et savoir le mettre en oeuvre pour une administration centralisée des configurations. 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.  
La gestion de configuration de machines virtuelles.  
Historique de puppet et les différentes versions de puppet.  
Présentation de Puppet Enterprise.
- Architecture** : Principe client-serveur, modules de configuration, les agents Puppet, la console, l'outil de gestion de cloud, les manifests puppet.  
Présentation des plate-formes supportées comme serveur et comme client (agent puppet).  
Mode opératoire : définition des configurations, vérification de l'état des clients, simulation des changements proposés par Puppet, application sur les systèmes cibles.

## **Puppet : administration centralisée**

Installation et configuration : Prérequis systèmes.  
Travaux pratiques :  
Installation d'une infrastructure Puppet : serveur, base de données, agents.  
Etude du fichier de configuration puppet.conf  
Configuration des rôles : maître, agent, console...  
Gestion des certificats sur les clients.

Le langage puppet : Introduction : présentation des manifests, modules, templates, des modules disponibles sur la forge puppet.  
Les manifests : description du langage déclaratif de puppet.  
Syntaxe, dépendance entre instructions.  
Etudes des variables, présentation des facts.  
Déclarations conditionnelles : if, case, selectors  
Travaux pratiques :  
rédaction d'un exemple de manifest de création d'un fichier et modification des droits,  
application de ce manifest sur un poste client cible.  
Etude détaillée des 'ressources' puppet :  
file, package, service, user, exec, notify, ..  
Exemple de la ressource 'file' :  
différents attributs disponibles pour déterminer les caractéristiques d'un fichier,  
s'assurer de sa présence, ...  
Exemple de la gestion de dépendance :  
production de manifest pour l'installation de sshd  
Les classes et modules.  
Installation de modules.  
Travaux pratiques :  
création d'un module pour propager la configuration de bash sur les postes clients.  
Définition des templates.  
Exemple de création de templates en utilisant le langage ERB.  
Etude des paramètres de classes.  
Définition de nouveaux types de ressources.  
Mise en oeuvre sur de nombreux travaux pratiques.  
Utilisation et génération de documentations au format puppet.  
Organisation du site .pp.  
Présentation des 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  
1090 €

1er au 2 mars  
14 au 15 juin

4 au 5 octobre  
13 au 14 décembre

### Public:

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

### Objectifs:

Comprendre le fonctionnement le fonctionnement avancé de puppet, 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 .  
Travaux pratiques :  
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.



## Puppet : expertise

- Les modules** : Présentation de "Puppet Forge" et recherche de modules.  
Etude de modules disponibles sur 'puppet forge' :  
puppetlabs/stdlib, utilisé dans la plupart des autres modules,  
puppetlabs/ntp : pour synchroniser les postes ou machines virtuelles,  
puppetlabs/apt, puppetlabs/firewall, ...  
Le langage de description de configuration.  
Travaux pratiques:  
Création de modules, mise en oeuvre, et dépôt sur la forge.  
Exemple : lien avec Nagios par le module thias/nagios
- Cloud Provisionner Puppet** : Présentation : gestion des instances de machines virtuelles dans le cloud  
Travaux pratiques :  
installation de Cloud Provisionner et démarrage d'instances  
Exemple d'utilisation :  
cas du bootstrap pour installer puppet dès la création de la machine virtuelle.
- Traitement des rapports** : Description du fonctionnement et des différentes étapes :  
activation dans le fichier puppetconf,  
stockage par le puppet Master.  
Travaux pratiques :  
installation de la Dashboard et configuration de l'environnement,  
création d'un nouvel utilisateur,  
préparation de la base,  
activation des jobs des delayed jobs,  
gestion des rapports émis par les agents puppet
- MCollective** : Présentation des fonctionnalités  
Principe de fonctionnement Middleware  
Installation et premier test : validation des clients collectés.

## Gestion de configuration avec Chef

**Durée:** 3 jours  
1610 € HT

7 au 9 février  
2 au 4 mai

12 au 14 septembre  
28 au 30 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,  
Etapas 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.



## Gestion de configuration avec Chef

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.

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..

# AWS : architecture SI, réseaux virtuels

**Durée:** 1 jour  
620 €

29 janvier  
2 mai

27 août  
19 novembre

**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 : Rappels sur les définitions du cloud. 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, objets connectés, ...
  - Principe de la compatibilité EC2 et S3. APIS. Interopérabilité, automatisation.
  
- 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) Création de VM. Les AMIs disponibles. MarketPlace. Les gabarits disponibles. Les droits d'accès, gestion des clés. Paramètres des machines : Elastic Block Storage (EBS), adresses IP élastiques,



## **AWS : architecture SI, réseaux virtuels**

Sécurité : Authentification et autorisation dans le cloud.  
Présentation AWS Identity et Access Management.  
Les rôles prédéfinis. Création d'un rôle.  
Ajout d'utilisateurs, de groupes. Affectation aux rôles.  
Mise en oeuvre sur les ressources AWS.

## AWS : stockage,S3,dynamodb..

**Durée:** 2 jours  
1190 €

5 et 6 février  
14 au 15 mai

28 au 29 août  
20 au 21 novembre

### 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. Emplacement géographiques des buckets.  
définition des droits d'accès,  
de règles de cycles de vie  
(destruction, sauvegarde et archivage avec Glacier, ...)  
Chargement de données depuis un poste local. Interopérabilité avec OpenStack/Swift, Hadoop, Cassandra, ceph.



## AWS : stockage,S3,dynamodb..

Bases de données : RDS :

mise en oeuvre d'une base de données au travers de RDS. Accès distant. Sécurisation de l'accès. Intégration dans des serveurs d'applications locaux.

Mise en oeuvre avec jboss.

Redshift :

principe des clusters Redshift,

définition de snapshots,

gestion de la sécurité, groupes,

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

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  
1190 €

22 et 23 février  
16 au 17 mai

30 au 31 août  
22 au 23 novembre

**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 compatibilité EC2 et S3. Le positionnement des grands acteurs du cloud.

Exemples de scripts : Automatisation des procédures de gestion de serveurs de calcul ou d'espaces de stockage. 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 plateformes 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. Utilisation des outils d'administration d'AWS en ligne de commandes.  
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 Réseaux et TCP/IP

Introduction aux réseaux	p127
TCP/IP : protocoles et mise en oeuvre	p130

Migration vers IPv6	p132
Messagerie	p135
Mise en oeuvre SNMP	p136
Annuaire ldap	p138



## Introduction aux réseaux

**Durée:** 4 jours  
2140 €

29 janvier au 1er février  
3 au 6 avril

2 au 5 juillet  
15 au 18 octobre

### 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

## Introduction aux réseaux

- Réseaux sans fils : HiperLAN. IEEE 802.11  
Wlan
- Normes Wifi : Présentation. Points forts, points faibles.  
Architecture des réseaux Wifi : 802.11, exemple d'ESS, le monde ad hoc, OLSR  
Le matériel, interopérabilité
- Utilisation du Wifi : Points d'accès. Modes de fonctionnement, mode répéteur, Mode pont  
Alignement d'antennes, supervision de réseaux
- 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, fonctionnalités. La trame IPV6, adressage.
- Sécurisation : VPN et tunnels: Objectif, fonctionnement  
DMZ et Pare-feux : Définition, serveur Proxy, fonctionnement pare-feux et tunneling  
Filtrage: les iptables, politique par défaut, état des connexions, traduction d'adresses, traduction de ports, connexion à internet
- Voix sur IP : Commutation de paquets. Avantages de la voix sur IP  
Les protocoles : H323, SIP. Introduction RTP : définition et applications, RTP et Nat  
Utilisation du registrar SIP avec Asterisk. Création des comptes téléphones, du dialplan, verification et tests  
Enregistrements SRV : serveurs DNS et Asterisk  
Transport de données  
Bande passante et qualité de service (QoS)
- Evolutions : L'adressage IP, la sécurité, les réseaux de stockage.



## TCP/IP : protocoles et mise en oeuvre

**Durée:** 4 jours  
1870 €

12 au 15 février  
22 au 25 mai

8 au 11 octobre  
3 au 6 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)

## TCP/IP : protocoles et mise en oeuvre

- 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).  
 Définitions : résolution de noms  
 Principe : noms de domaines, notion de zones et de responsabilité d'une zone  
 Architecture : client/serveur  
 Présentation des notions de serveur primaire, secondaire, cache dns  
 Arborescence des noms de domaines.  
 Etude du traitement d'une requête de résolution de nom DNS.  
 Mise en oeuvre avec bind. Configuration d'un client dns.  
 Outils d'interrogation : nslookup, host, dig.  
 Configuration d'un serveur DNS sous Linux.  
 Etude du fichier named.conf  
 Analyse des flux et des requêtes client-serveur avec wireshark  
 Principe d'un serveur DNS secondaire.  
 SNMP (Simple Network Management Protocol) :  
 fonctionnalités, apports SNMP V2.
- IPv6** : Adressage actuel, attribution des adresses.  
 Le travail de l'IETF (BradnerMankin). Plan d'adressage sur 128bits.  
 Agrégateurs : découpage TLA/NLA/SLA/IID. Intégration des Regional Registries  
 Fonctionnement : Surcharge d'entêtes. Structures des trames. Les nouveaux mécanismes: fragmentation: MTU universelle, DHCPv6, dynamic DNS, renumérotation simplifiée d'un plan d'adressage
- Sécurité** : Ipvsec (IP Security Protocol)  
 TP de mise en oeuvre

## IP: Migration vers IPv6

**Durée:** 2 jours

Prix et dates: nous consulter

### 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, ...

## IP: Migration vers IPv6

- 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.
- Gestion des adresses** : 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.
- DNSv6** : Mise en oeuvre d'un DNS v4/v6.  
Les différentes implémentations. DNS dynamiques.
- Migration v4/v6** : Les différentes approches : double pile, encapsulation statique, encapsulation dynamique.  
Impacts de la suppression du NAT.  
Utilisation du cycle de vie des adresses.  
Les différents tunnels. Mise en 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.

## IP: Migration vers IPv6

- Cas concret : Exemple de migration d'une infrastructure complète.  
La sécurité : IPsec, les pare feux et les filtres. ip6tables, routeurs et répartiteurs de charge, haproxy.  
Organisation de la migration.  
Impacts de la mise en oeuvre de plusieurs dhcpv6 et de multiples agents RA.  
Problème de boucles de tunnels.
- Routage dynamique IPv6 : 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.

# Messagerie

**Durée:** 2 jours

Prix et dates: nous consulter

**Public:**  
Administrateurs réseaux.

**Objectifs:**  
Savoir installer, configurer et administrer une messagerie sous Unix.

**Connaissances préalables nécessaires:**  
Il est demandé aux participants de connaître les notions de base sur les réseaux TCP/IP.

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



## Sécurité TCP/IP

**Durée:** 5 jours

Prix et dates: nous consulter

### Public:

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

### Objectifs:

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

### Connaissances préalables nécessaires:

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

### Programme:

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

## Sécurité TCP/IP

Surveillance : Le protocole SNMP; la surveillance d'applications  
TP : écriture d'un analyseur de topologie  
Contrôle des flux. Analyseurs de trames : ethereal, tcpdump,  
analyse de failles : nessus



## Mise en oeuvre du protocole SNMP

**Durée:** 2 jours  
1100 €

15 et 16 mars  
7 au 8 juin

11 au 12 octobre  
13 au 14 décembre

### 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 oeuvre du protocole SNMP

- Mise en pratique : Commandes d'interrogation des agents SNMP : snmpget, snmpwalk, Notions de communauté et d'Oid (Object Identifier).  
Configuration d'un agent snmp sous Linux.  
Exécution de l'agent comme un service.  
Interrogations simples : description des cartes réseaux du poste client, affichage de la table de routage, ...
- Outils d'interrogation : Graphiques : PTKMib, Mib Browser, MIB Smithy, Automatisation des requêtes avec net-snmp et scli (en mode commande).
- Les alertes : Création d'un serveur d'alertes avec snmptradd.  
Définition des conditions d'alertes pour chaque objet.
- Sécurité : Authentification  
Protection du contenu
- L'usage de SNMP sur le marché : Les produits d'analyse, les MIBs développées par les constructeurs.
- Développement : Développement d'une MIB. Présentation des produits de développement.  
Description de la structure en ASN-1.  
Travaux pratiques :  
conversion en C et compilation dans l'agent SNMP,  
ajout d'OID surveillant la température du processeur,  
ajout d'OID surveillant le nombre de threads d'un serveur JEE.



## Voix sur IP

**Durée:** 4 jours

Prix et dates: nous consulter

### **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

## Voix sur IP

- Qualité** : Qualité de la voix. Compression. Détection des défauts.  
 Les différents codecs pour l'encodage de la voix.  
 Bande passante utilisée. Calibrage. Utilisation de RSVP (Resource Reservation Protocol), MPLS (Multiprotocol Label Switching) ou TOS.  
 Mesure du trafic. Utilisation des iptables.  
 Gestion de la qualité de service et répartition voix/données.
- Mise en pratique** : Configuration de téléphones SIP depuis un navigateur :  
 choix du codec, renseignement de l'adresse SIP, indication d'un serveur DNS, ...  
 Installation et configuration d'un registrar Asterisk.  
 Création des comptes des téléphones et du dialplan (plan de numérotation).
- Les enregistrements SRV** : Communication entre des réseaux différents.  
 Utilisation des serveurs DNS.  
 Travaux pratiques : Création de deux domaines DNS, mise à jour des enregistrements SRV pour la localisation des services sip/udp.  
 Configuration des softphones, et tests.  
 Mise en évidence des requêtes DNS par les traces réseau.
- Plan de numérotation** : Travaux pratiques avec le dialplan Asterisk :  
 /etc/asterisk/extensions.conf.  
 Notions de contexte, extensions, priorités, et applications.  
 Configuration d'appels stateful et stateless.
- La sécurité** : Critères de sécurité : Disponibilité, Intégrité, Confidentialité.  
 Mise en évidence des risques pour chaque critère.  
 Contraintes légales concernant les appels d'urgence.  
 Travaux pratiques : à partir d'un analyseur réseau (wireshark), capture de trames VoIP sur le réseau,  
 relecture des échanges téléphoniques, saturation d'un serveur, déni de service.  
 Solutions techniques : traitement des appels d'urgence par Asterisk, chiffrement du protocole SIP, chiffrement des flux RTP avec RTPS, mise en place de tunnels, solutions haute disponibilité pour les serveurs.

## Voix sur IP

Déploiement de masse : Le provisioning : automatisation de la configuration des terminaux téléphoniques.  
Avantages : gestion d'un nombre important de téléphones (logiciels ou matériels).  
Principe : distribution d'adresses IP dynamiques par un serveur BOOTP/DHCP, mise à disposition des fichiers de configuration pour chaque téléphone.  
Travaux pratiques : configuration du provisioning pour des terminaux SIP (Linksys de Cisco).

# Annuaire LDAP

**Durée:** 2 jours  
1090 €

15 au 16 février  
2 au 3 mai

27 au 28 septembre  
5 et 6 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.



## Annuaire LDAP

- Schéma ldap : Définitions : attributs, objets  
Format du schéma.  
Mécanisme d'héritage des attributs.  
Organisation d'un schéma.  
Travaux pratiques :  
intégration d'un schéma extérieur.
- 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 p141  
Administration Nagios p143

Supervision avec Shinken p146  
Administration Zabbix p148

Gestion de parc avec OCS et glpi p150  
Gestion de versions avec git p153

# Supervision nagios : utilisation

**Durée:** 3 jours  
1610 €

5 au 7 février  
9 au 11 avril

3 au 5 septembre  
10 au 12 décembre

**Public:**

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

**Objectifs:**

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

**Connaissances préalables nécessaires:**

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

**Programme:**

Supervision : : Les objectifs de la supervision, les techniques disponibles.  
définitions : Objets supervisés.

Les services et : Rappels sur les principes HTTP, SMTP, NNTP, POP3, PING.  
ressources : Définition des ressources à surveiller.

Présentation de : Les fonctionnalités  
Nagios : 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



## Supervision nagios : utilisation

Configuration Nagios	: Objets à définir : hôtes, groupes, services, dépendances, notifications, escalades Description des serveurs à surveiller, des contacts, création de groupes de serveurs, de groupes de contacts Notion de hiérarchie avec les hôtes parents, les dépendances de services, hôtes et groupes.
Déploiement	: Sur les hôtes, principes de NRPE, NSCA. Travaux pratiques : écriture de scripts de déploiement. (NRPE)
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  
2550 €

5 au 9 février  
9 au 13 avril

3 au 7 septembre  
10 au 14 décembre

### Public:

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

### Objectifs:

Connaître les fonctionnalités de Nagios. Savoir installer, configurer et administrer le produit. Savoir développer des nouveaux plugins, et mettre en oeuvre la supervision dans un environnement hétérogène.

### Connaissances préalables nécessaires:

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

### Programme:

- Supervision : : Les objectifs de la supervision, les techniques disponibles.  
définitions : Rappels sur les principes HTTP, SMTP, NNTP, POP3, PING  
Définition des ressources à surveiller.
  
- Présentation : : Les fonctionnalités  
Nagios : 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



## Administration 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.
- Configuration : 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
- Optimisation de l'ordonnanceur : Méthode d'ordonnancement.  
 Délai entre chaque test.  
 Entrelacement des services.  
 Tests concurrents.  
 Fréquence de récupération.
- Contrôle et débuge : Analyse des fichiers de logs.  
 Commandes de contrôle.  
 Mode d'exécution des plugins.  
 Options détaillées.
- Les plugins : Principe de fonctionnement.  
 Mise en oeuvre des plugins standards.  
 Travaux pratiques :  
 Personnalisation de Nagios par développement de nouveaux plugins.
- Gestionnaire d'évènements : Mécanisme de traitement d'erreur.  
 Normalisation.  
 Algorithmie de l'ordonnanceur.  
 Macros d'évènements.  
 Démarche d'implémentation.  
 Exemple : relance d'un serveur web.

## Administration Nagios

- Lien SNMP : Présentation du protocole SNMP.  
Hétérogénéité des superviseurs et du parc supervisé.  
Tests actifs et passifs.
- Supervision distribuée : Principe des agents.  
Sur les hôtes, principes de NRPE, NSCA.  
Travaux pratiques :  
écriture de scripts de déploiement. (NRPE)  
installation de nsca et configuration.
- Superviseurs redondants : 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.
- 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  
1630 €

26 au 28 février  
2 au 4 mai

17 au 19 septembre  
19 au 21 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 unix/linux.

### Programme:

- |                               |  |
|-------------------------------|--|
| Supervision :<br>définitions  | : Les objectifs de la supervision, les techniques disponibles.<br>Objets supervisés.   |
| Les services et<br>ressources | : Rappels sur les principes HTTP, SMTP, NNTP, POP3, PING.<br>Définition des ressources à surveiller.   |
| Présentation de<br>Shinken    | : Historique, licence, fonctionnalités :<br>Supervision, exploitation, surveillance des services réseaux,<br>surveillance des ressources (charge CPU, espace disque),<br>émission d'alertes,<br>actions automatiques programmables, gestion de règles métier.<br>L'architecture : Arbiter, Scheduler, Poller, Reactionner, Broker.<br>Principe d'architecture distribuée.<br>Comparaison avec Nagios.<br>Apports de Shinken. |

## Supervision avec shinken

- Installation et configuration** : Prérequis techniques, le référentiel MongoDB  
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.  
Configuration : ajout de widgets, packs, etc  
Gestion de la configuration :  
utilisation des templates  
Notions de groupes d'hôtes et de services  
Autodécouverte avec nmap.
- Les modules** : Principe, installation de modules  
Présentation du site shinken.io  
Travaux pratiques : mise en oeuvre de modules simples.  
Installation de l'interface Thruk.  
Développement de modules en shell et en python.
- Performances** : La haute disponibilité avec Shinken.  
Lissage automatique de la charge par l'architecture de Shinken.



# Zabbix administration

RS150

**Durée:** 3 jours  
1630 €

26 au 28 février  
2 au 4 mai

17 au 19 septembre  
19 au 21 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.



## Zabbix administration

- 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 évènements.  
Visualisation.  
Création de templates  
Les utilisateurs : configuration, groupes d'utilisateurs,  
droits d'accès
- Applications types** : Supervision de services web,  
de machines virtuelles,  
auto-découverte des éléments réseau  
Travaux pratiques :  
utilisation de l'interface web
- Supervision distribuée** : Principe des proxy Zabbix.  
Mise en oeuvre d'une architecture distribuée.



## Gestion de Parc avec OCS et GLPI

**Durée:** 3 jours  
1660 €

5 au 7 mars  
14 au 16 mai

5 au 7 septembre  
12 au 14 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** : 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.
- Les plugins** : Principe des plugins.  
Mise en oeuvre pratique des plugins tracker, reports, racks, data injection, item uninstallation.
- Import OCS NG** : Objectif et principe de fonctionnement.  
Configuration du mode OCSNG. Options d'importation. Mode d'import OCS.  
Import OCS par le plugin massocsimport.
- Gestion du cycle de vie** : 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.
- Télédéploiement de:** Principe et architecture. Notion de priorité et action à exécuter paquets
- Exploitation OCS glpi** : Sauvegardes, journalisation. Lien avec un annuaire ldap.
- Fusion Inventory** : 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.



## Gestion de versions avec GIT

**Durée:** 2 jours  
1140 €

15 au 16 mars  
14 au 15 juin

20 au 21 septembre  
22 au 23 novembre

### Public:

Tout développeur, chef de projet, architecte, souhaitant utiliser git comme gestionnaire de versions

### Objectifs:

Comprendre les principes d'un gestionnaire de version distribué, les apports de git, savoir le mettre en oeuvre pour gérer les codes sources d'un projet, les versions, corrections de bugs, etc ..

### Connaissances préalables nécessaires:

Connaissance des processus de développement et d'un langage de programmation.

### Programme:

Présentation de Git : La notion de gestionnaire de versions distribué  
Les principes techniques de Git  
Aperçu des workflows possibles

Prise en main : Installation et configuration de git  
Création d'un premier référentiel  
Utilisation de la ligne de commande pour les opérations de base

Comprendre les principes de Git : Référentiels  
Clonage de référentiels  
Index  
Répertoire de travail

Travailler en équipe: Connexion à un référentiel  
Ajout, modification, suppression de fichiers et répertoires  
Gestion des commits  
Synchronisation avec un référentiel distant  
Comparaison  
Utilisation des tags  
Créer et appliquer des patches

## Gestion de versions avec GIT

- Gestion des branches : Création de branches  
Navigation entre branches  
Fusion de branches  
Résolution des conflits  
Branche temporaire
- Compléments : Interagir avec des référentiels partagés via GitHub  
Gerrit, un système de revue de code à connaître  
Gitweb, l'interface web officielle



## Filières Internet et Web

Serveur web Apache p156  
Administration serveur Nginx p158

Serveur proxy Squid p160  
Sécurisation services internet p162

Développement web HTML et CSS p165  
Web dynamique avec Javascript p167  
HTML5 , CSS3 et Javascript p169  
Javascript avec JQuery p170

Angular JS Développement p172  
Angular V2 p174  
Angular JS avancé p176  
Angular V2 avancé p177

Node JS : mise en oeuvre p179

# Serveur WEB : apache

**Durée:** 3 jours  
1610 €

12 au 14 février  
3 au 5 avril

2 au 4 juillet  
8 au 10 octobre

**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



## Serveur WEB : apache

- WebDynamique : Principe des scripts CGI, fonctionnement, mise en place. Apports de la solution fastCGI.
- 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  
1610 €

12 au 14 février  
3 au 5 avril

2 au 4 juillet  
8 au 10 octobre

## Public:

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

## Objectifs:

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

## Connaissances préalables nécessaires:

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

## Programme:

- Introduction : Présentation, historique, licence.  
Fonctionnalités : serveur http, proxy inverse, proxy de messagerie, diffusion de contenus vidéo, ...  
Plates-formes supportées. Architecture : principe de serveur asynchrone, modularité.
- Mise en oeuvre : Choix des modules, d'une version/distribution.  
Travaux pratiques : installation, démarrage, configuration de base
- Protocole HTTP : Fonctionnement, format des requêtes.  
Méthodes. Syntaxe d'une URL.
- Configuration du serveur nginx : Etude du fichier /etc/nginx/nginx.conf : http-block, server-block, location-block.
- Configuration d'un site principal : Nom interne du serveur, rappel sur les DNS, page d'accueil, types de fichiers, les alias, définition de chemins relatifs,  
Travaux pratiques : mise en oeuvre avec création d'un site et configuration du serveur Nginx.
- Virtuals Hosts, locations : Principe, configuration, hôtes virtuels basés sur l'adresse IP, sur le nom.  
Configuration des URLs.



## Administration serveur Nginx

- Scripts CGI : Exécution d'applications en PHP, en Python.
- Serveur proxy : Configuration Nginx en proxy inverse et en proxy messagerie.
- Exploitation : Administration du site.  
Les fichiers journaux : création de fichiers de logs séparés pour chaque hôtel virtuel.  
Mécanisme de rotation des fichiers journaux.  
Analyse des informations stockées dans les logs.  
Travaux pratiques : mise en charge du site et visualisation du comportement.
- 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



## Serveur Proxy Squid

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

Prix et dates: nous consulter

**Public:**

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

**Objectifs:**

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

**Connaissances préalables nécessaires:**

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

**Programme:**

Introduction : La sécurité : Définitions  
Le DIC : Disponibilité, Intégrité, Confidentialité  
Evaluation des risques.  
Elaboration d'une politique de sécurité.  
Définitions : DMZ, proxy, pare-feux, routeurs filtrants  
Fonctionnement, mise en oeuvre

Les ports de niveau 5 : Rappels sur la notion de ports.  
Principe d'empilement des trames.  
Les ports UDP.  
Exemples de trames.

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

DMZ et pare-feux : Définition, et architecture  
Fonctionnement des pare-feux.



## Sécurisation des services internet

- Firewall iptables** : Description du filtrage avec iptables.  
Syntaxe.  
Traduction d'adresses. Traduction de ports.  
Notion de politique de sécurité par défaut.  
Sécurisation de l'ouverture d'un port.  
Travaux pratiques :  
Mise en oeuvre d'une fonction d'activation d'ouverture avec les iptables.
- SSH et GPG** : SSH : Secure SHell  
Présentation du protocole de connexion SSH  
Utilisation de la cryptographie. Authentification par clé.  
Travaux pratiques :  
configuration,  
sécurisation de l'ouverture du port.  
GPG : GNU Privacy Guard  
Transmission de messages chiffrés et signés.
- VPN et tunnels** : Objectifs.  
Fonctionnement. Mise en oeuvre : tunnels ssh, stunnel.  
Travaux pratiques avec openVPN.  
Principe d'IPsec.  
Implémentation avec FreeS/WAN.
- Proxy** : Introduction : la fonction d'un serveur proxy.  
Les multi-serveurs proxys.  
Mise en oeuvre avec squid.  
Architecture Squid.  
Exemples d'utilisation.  
Travaux pratiques :  
configuration réseau d'un serveur Squid.  
Les Access Control List.  
Optimisation de la bande passante.  
Programmes d'authentification.  
Exemple d'authentification NCSA, LDAP.
- Services reseaux** : Sécurisation DNS :  
architecture redondante, DNS fermé, Hidden master, Stealth DNS.  
Mise en oeuvre de la sécurisation d'un DNS.  
Authentification.  
Les signatures TSIG. Les ACL.

## Sécurisation des services internet

- Messagerie : Sécurisation.  
Notion de relais ouvert.  
Outils de test du serveur.  
Travaux pratiques :  
mise en oeuvre avec postfix.  
Apport de Cyrus.
- Serveur http : Restriction d'accès.  
Authentification http.  
Sécurité avec SSL et HTTPS : principe, configuration par défaut.  
Génération de clés SSL
- Acces reseaux : Authentification par un serveur radius.  
Techniques de filtrage à base d'iptables.  
Mise en oeuvre d'un système d'authentification par utilisateur et non  
par adresse : NuFW.
- Détection de failles : Outils : snort, openvas.  
Travaux pratiques :  
mise en oeuvre et réalisation d'un audit de sécurité de  
l'environnement de test.



# Développement Web HTML et CSS

**Durée:** 2 jours  
1090 €

5 au 6 février  
14 au 15 mai

24 au 25 septembre  
12 au 13 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:

Maitriser les fonctionnalités de base d'HTML5 pour concevoir des pages webmestres

## Connaissances préalables nécessaires:

Connaissances générales sur Internet.

## Programme:

- Introduction** : Contexte historique du web, présentation du langage HTML, bon usage des balises  
Les nouveautés du HTML5
- Contenu et présentation** : Texte simple, titres, paragraphes  
Mise en forme, caractères spéciaux  
Listes, séparateurs, autres balises de texte  
Les iframe
- Les tableaux** : Utilisation, structure  
Les cellules, fusion des cellules  
Titre et légende, en-tête  
Groupes de colonnes/lignes
- Les liens** : Balise HTML  
Liens externes, liens internes  
Liens de mail, liens vers des fichiers  
Attributs, couleurs
- Les images** : Les couleurs en HTML  
Formats d'images du Web  
Insertion d'une image, attributs, arrière plan

## Développement Web HTML et CSS

- Les formulaires : Structure HTML  
Types de champs:  
texte de saisie, listes, cases à cocher  
Champs cachés, mot de passe  
Envoi de formulaire GET, POST  
Transfert de fichier  
Validation
- Feuilles de styles CSS : Objectifs, syntaxe  
Style interne/externe  
Types de sélecteurs (Balise, ID, Classe)  
Les pseudo-classes  
Modèle de boîtes, positionnement, décoration  
Séparation contenu présentation
- En-tête HTML : Le DOCTYPE, le rôle des balises, balises meta importantes  
Conseils pour le référencement
- Multimédia : Présentation, fichiers audio, fichiers vidéo  
Animation Flash
- JavaScript : Le code côté client/serveur  
Présentation, scripts internes et externes  
Contenu du document, événements  
Fonctions utiles, traiter un formulaire



## Web Dynamique avec JavaScript

**Durée:** 3 jours  
1650 €

22 au 24 janvier  
14 au 16 mars  
28 au 30 mai

12 au 14 septembre  
26 au 28 novembre

### Public:

Développeurs web, architectes web, chefs de projets, webmestres

### Objectifs:

Comprendre et maîtriser le langage JavaScript

### Connaissances préalables nécessaires:

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

### Programme:

**Présentation** : Historique et évolution du langage. Evolution de l'utilisation du langage  
Organisation du code. Outils de développement.  
Principes de base HTML et CSS. Les règles, sélecteurs, propriétés de styles, etc ..  
Interaction avec Javascript.

**Syntaxe Javascript** : Les variables, les types (Number, String, Boolean, ...). Tableaux, boucles et tests.  
Les opérateurs arithmétiques et logiques.  
Travaux pratiques : réalisation d'exemples simples  
Gestion des erreurs et des exceptions, exemples de mise en oeuvre des instructions 'try', 'catch', 'throw', 'finally'  
Utilisation de la console. Méthodes et outils de debugging.  
Présentation des fonctions globales et des classes natives.  
Définition des fonctions. Gestion des arguments

**Utilisation du DOM** : Présentation du Document Object Model (DOM).  
Fonctions de sélection, fonctions de création d'objet DOM  
Modifier les éléments du DOM.  
Travaux pratiques : exemple de validation d'un formulaire

## Web Dynamique avec JavaScript

- Gestion des évènements : Principe et définitions. Présentation des évènements courants.  
Flux événementiel du DOM. Lier un évènement à un objet du DOM.  
Intéragir avec les éléments du DOM.  
Travaux pratiques : exemple d'un gestionnaire d'évènement générique  
L'objet 'event'. Les types d'évènements à gérer. Bonnes pratiques.
- AJAX : : Présentation et exemple d'utilisation  
Asynchronous JavaScript And XML
- Déroulement d'une requête AJAX : Protocole utilisé, limites.  
Détails de l'objet XMLHttpRequest  
Travaux pratiques : Initialisation d'une requête AJAX et utilisation  
Contourner les limitations
- Programmation Orientée Objet : Définitions de la POO.  
Utilisation de la POO en Javascript.  
Plusieurs façons de créer un objet en Javascript.  
Améliorer la création d'un objet avec « prototype »  
Emuler un singleton en javascript  
Travaux pratiques : exemple de création d'objets 'inline'
- Fonctions avancées en Javascript : Utilisation du mot clé « this »  
Les closures, définition, cas d'utilisation.  
Méthodes apply et call
- Présentation des nouvelles fonctionnalités liées à HTML 5 : Nouvelles balises (vidéo, son, ...)  
L'API File, les websockets, les workers, le webstorage



## HTML5 CSS3 et javascript

**Durée:** 3 jours  
1550 €

7 au 9 février  
16 au 18 mai

26 au 28 septembre  
14 au 18 novembre

### Public:

Développeurs web, architectes web, chefs de projets, webmestres

### Objectifs:

Maîtriser les fonctionnalités avancées d'HTML5 JavaScript et CSS3 pour concevoir des applications webmestres

### Connaissances préalables nécessaires:

Bonne connaissance du langage javascript. Connaissance des fondamentaux internet et HTML

### Programme:

HTML5	: Nouveautés, doctype Éléments syntaxiques et nouvelles balises Les nouveaux attributs, evolution des formulaires Découverte d'Ajax Level 2, les microformats, les attributs data
API Javascript	: Sélecteurs CSS, usage des Timers Les Workers et l'API Message
Web Storage	: Session, local. Database et IndexedDB
Drag and Drop	: Content Editable et commandes. Offline web application. Géolocalisation, web Socket, device API
CSS3	: Présentation, les sélecteurs, pseudo classes Media Queries et responsive design
Mise en forme	: Polices exotiques, ombrages, transparence, dégradés Propriétés display
Transitions	: Transformations, animations
API mobile	: JQuery Mobile

# Javascript avec JQuery

**Durée:** 3 jours  
1650 €

31 janvier au 2 février  
2 au 4 mai

3 au 5 septembre  
5 au 7 décembre

**Public:**

Développeurs web, architectes web, chefs de projets, webmestres

**Objectifs:**

Développer en JavaScript avec jQuery

**Connaissances préalables nécessaires:**

Très bonne pratique du langage Javascript. Connaissances des fondamentaux internet et HTML

**Programme:**

Présentation de JQuery : Installer JQuery  
Conseils sur les performances

Première utilisation : La fonction `$()` ou `jQuery()`, les sélecteurs jQuery  
Sélecteur CSS, sélecteur d'attribut, sélecteur personnalisés  
La méthode `.filter()`, autres méthodes de parcours du DOM

Manipuler les éléments du DOM : Accéder directement à un élément du DOM  
Modifier les balises dynamiquement

Les évènements en jQuery : Description des évènements  
Évènements au chargement de la page, évènements associés au DOM  
Attacher un évènement sur un élément du DOM  
Méthode `.bind()`, `.live()`, `.delegate()`, `.on()`  
Les raccourcis d'évènements, les callbacks  
callback avec et sans arguments  
Gérer les évènements multiples, propagation des évènements  
Retirer un évènement sur un élément du DOM  
Les évènements et les espaces de nom

AJAX avec JQuery : Premiers pas, les paramètres de la méthode `.ajax()`  
`callback.done()`, `callback.fail()`



## Javascript avec JQuery

Les webStorage et : Mise en pratique  
éléments éditables

Les effets visuels : Effets visuel intégrés à jQuery  
Animations personnalisées



# Angular JS développement

**Durée:** 3 jours  
1650 €

19 au 21 mars  
23 au 25 mai

17 au 19 octobre

**Public:**

Développeurs web, architectes web, chefs de projets, webmestres

**Objectifs:**

Comprendre les principes de AngularJS, savoir développer une application web et utiliser les bonnes pratiques de développement.

**Connaissances préalables nécessaires:**

Très bonne pratique du langage Javascript. Connaissance des fondamentaux internet et HTML

**Programme:**

- Introduction : Historique et principes généraux  
Pattern MV\*, data-binding bi-directionnel  
Avantages et inconvénients  
Travaux pratiques: installation et découverte  
Développement application « Hello world! »
  
- Modèles, vues et contrôleurs : Le modèle  
Les vues : templates, expression, directives, filtres  
Les contrôleurs : fonctionnement, le \$scope
  
- Les modules : Création et configuration, partage de services  
Injection de dépendances
  
- Le routage : « Single page application »  
Configuration, méthode .config du module  
Paramètres d'url, événements de routage
  
- Les formulaires : Lien entre ng-model et contrôleur  
Validation, états des formulaires, gestion d'erreur  
Types de champs, classes CSS, custom validation



## Angular JS développement

Les directives et les filtres	: Convention de nommage, écriture normalisée Types de composants, directives de templates Options de configuration, transclude, scope, cycle de vie Fonctions compile et link, créer ses propres filtres
Cycles de vie et scopes	: Présentation du fonctionnement d'AngularJS L'arbre des scopes, les méthodes de \$rootScope Les événements : diffusion et interception
Les services AngularJS	: Les services natifs, les values et contants Factory, Service et Provider
AJAX et REST	: Le service « \$http », le service « \$resource » L'API de promise
Aller plus loin	: Bonnes pratiques Internationalisation, angular-translate L'interface, l'initialisation d'un projet
Tester une application AngularJS	: Tests unitaires : Karma, Jasmine Tests end-to-end (e2e) : Protractor Outils de debug ng-inspector

# Node JS mise en oeuvre

**Durée:** 3 jours  
1650 €

27 février au 1er mars  
26 au 28 avril

19 au 21 juin  
2 au 4 octobre

**Public:**

Développeurs web, architectes web, chefs de projets, webmestres

**Objectifs:**

Développer en javascript côté serveur avec Node.js. Comprendre les principes de Node.js et utiliser les bonnes pratiques de développement.

**Connaissances préalables nécessaires:**

Connaître les fondamentaux internet.Savoir programmer en javascript.

**Programme:**

- Introduction : Historique, principes généraux.  
Fonctionnement interne  
Exemples d'applications  
Avantages et inconvénients  
Installation et découverte  
Application « hello world »
  
- Les modules : Se servir des modules de base, NPM
  
- Serveur web : Ma première application web  
Gérer les requêtes et les réponses HTTP  
Routage des URLs  
Opérations bloquantes et non-bloquantes
  
- Express : Paramétrage, gestion des requêtes HTTP  
Sessions, templating
  
- Bonnes pratiques de développement : Organisation (modules, routes, etc.)  
Code
  
- Autres composants : Socket.IO, connect, async
  
- Tests : assert / expect.js, Mocha / Zombie, PhantomJS / CasperJS



## Node JS mise en oeuvre

Performances : single-thread et event-loop  
Communication inter-processus  
Redis, le module « cluster »



# Développement Web avec GWT

**Durée:** 4 jours

Prix et dates: nous consulter

## **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 Développement

Programmation en langage C p182  
Perfectionnement en langage C p184  
Programmation système en C p186  
Programmation noyau et drivers en C p189

Développement web avec PHP p190  
Programmation Perl p192

Le langage Python p194  
Développement python avancé p196  
Python web avec Django p198

Introduction à XML p200  
Développement applications android p201



## Programmation en langage C

**Durée:** 5 jours

Prix et dates: nous consulter

### 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.

## Programmation en langage C

- Fonctions** : Fonctions de base : affichage et lecture des données.  
 Entrées/sorties formatées : options d'affichage des caractères, entiers, ...  
 Fonctions spécifiques aux chaînes de caractères : strcpy, strcat, strchr,strupr, strlwr, strlen.  
 Fonctions personnalisées : définitions, règles de fonctionnement, récursivité.
- Programmation structurée** : Notion de structure.  
 Les unions, champs binaires, types énumérés.  
 Définition, déclaration, utilisation de structures.  
 Exemples d'utilisation des champs binaires et mise en oeuvre.  
 Déclaration d'énumération avec enum. Etude d'exemples.
- Les opérateurs.** : opérateurs de calcul, simplification d'écriture, opérateurs de décalage (>> et <<), et binaires( | ^ ).  
 Erreurs de conversion implicite.  
 Opérateurs de comparaison (== <= >= ? ...) et opérateurs logiques (! ||)
- 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

Prix et dates: nous consulter

### 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.

## Perfectionnement en langage C

- 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.
- 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

LC010

**Durée:** 3 jours

Prix et dates: nous consulter

## **Public:**

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

## **Objectifs:**

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

Prix et dates: nous consulter

## **Public:**

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

## **Objectifs:**

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

## **Connaissances préalables nécessaires:**

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

## **Programme:**

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



# Développement d'applications Web avec PHP

LH001

**Durée:** 5 jours  
2490 €

16 au 20 avril  
25 au 29 juin

1er au 5 octobre  
17 au 21 décembre

## 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.

## Développement d'applications Web avec PHP

- Connectivité avec les SGBDR : Les principaux SGBDR.  
SQLLite et MySQL.  
Connexion et déconnexion.  
Lire et mettre à jour les données.  
Gérer les erreurs. Requêtes préparées.
- La gestion des sessions. : Fonctionnement des sessions.  
Débuter une session et assigner des variables.  
Suppression des variables de session. Destruction d'une session. Les cookies.
- PHP et le système de fichiers. : La sécurité sur les fichiers.  
Les fonctions de manipulation d'images.  
Création de graphiques (JPGRAPH, ARTICHOW).  
Génération de fichiers Excel.  
Les concepts de sécurité, l'authentification.



## Programmation Perl

**Durée:** 3 jours  
1520 €

12 au 14 mars  
22 au 24 mai

17 au 19 septembre  
3 au 5 décembre

### Public:

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

### Objectifs:

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

### Connaissances préalables nécessaires:

La connaissance d'un langage de programmation sera appréciée.

### Programme:

- Introduction : Présentation de Perl : caractéristiques, positionnement par rapport à d'autres langages (C, Java, Shell, PHP)  
Plates-formes d'utilisation.  
Installation de Perl. 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

# Programmation Perl

- Tableaux : Manipulation de tableaux, hachage, sauvegarde des tableaux
- 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  
2170 €

12 au 15 mars  
4 au 7 juin

24 au 27 septembre  
26 au 29 novembre

## Public:

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

## Objectifs:

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

## Connaissances préalables nécessaires:

La connaissance d'un langage de programmation sera appréciée.

## Programme:

- Introduction** : Présentation Python : caractéristiques, positionnement par rapport à d'autres langages.  
Installation.  
Utilisation de l'interpréteur.  
Premier programme en Python.
- Les bases** : Principaux types de données : nombres, booléens, chaînes de caractères.  
Déclaration de variable, typage dynamique, mots clés réservés.  
Les opérateurs : priorité, associativité, opérateurs d'affectation, logiques, de comparaison.  
Quelques fonctions utiles : print(), input().  
Structures conditionnelles : if et elif.  
Boucle while, mots clés break et continue.
- Chaînes de caractères et listes** : Définition et manipulation de chaînes de caractères.  
Le type séquence.  
Les listes : définition, accès à un élément.  
Les références.  
Les tuples.  
Manipulation de listes : mot clé del, fonctions list et range; parcours d'une liste.

## Le langage Python

- 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 : Rappels sur la programmation objet  
Objet : Les classes en Python  
Constructeurs, attributs privés, méthodes, héritage
- Les fichiers : Méthodes d'accès aux fichiers :  
ouverture (accès en mode lecture, écriture, ajout, ...),  
fermeture, le mot clé with;  
lecture dans un fichier
- Les types de : Les listes de listes.  
données complexes : Une liste de tuples.  
Les dictionnaires.  
Parcours d'un dictionnaire.  
Les méthodes update(), clear(), pop(), del(), values, keys()
- Modules et : Les modules : définition, la fonction help()  
expressions : Importer des fonctions  
régulières : Créer ses propres modules  
Expressions régulières, les caractères spéciaux.  
Groupes et classes de caractères.  
Le module 're'.
- Gestion des : Principe, exemples d'exceptions.  
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 Gadgetfly, interfaçage MySQL avec MySQLdb, accès à PostgreSQL avec les modules PyGreSQL ou Psycopg, ...
- Développement web** : Présentation et comparaison des frameworks et langages de template.  
Les frameworks disponibles : CherryPy, Paste, CPS, Django, TurboGears, Pylons, ...  
Les langages de templates : Myghty, Python Server Pages, Cheetah, Zope,  
Mise en oeuvre de Django.  
Installation, configuration initiale :  
création d'un projet, serveur de développement Django, configuration des accès aux bases de données...  
création et activation de modèles, développement d'une application simple.

# Développement avancé avec Python

Liens avec les langages C et Java : Les besoins : accès à des programmes en C ou à des classes Java depuis Python, bénéficier des avantages de Python depuis des programmes Java, .. Les outils :Jython, Jepp (Java Embedded Python), JPE (Java Python Extension), Boost.Python, ...



# Développement web avec Django

**Durée:** 3 jours

Prix et dates: nous consulter

## Public:

Les développeurs en Python souhaitent créer des sites web avec Django

## Objectifs:

Savoir utiliser le framework Django pour le développement d'applications web

## Connaissances préalables nécessaires:

Connaissance de base de Python et des concepts de base des applications web

## Programme:

- 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
- Présentation de Django** : Langage de templates  
système de mapping d'url  
design pattern MVT,  
principe DRY  
Traitement d'une requête avec le pattern MVT
- Mise en oeuvre** : Installation, configuration initiale  
création d'un projet,  
configuration du projet : fichier settings.py  
serveur de développement Django,  
configuration des accès aux bases de données.  
authentification
- Développement applicatif** : Création et activation de modèles, de vues,  
développement d'une application simple  
Les templates  
Requêtes de type GET et POST  
Utilisation des tokens CSRF pour la sécurité  
Déploiement des projets sur un serveur Apache

# Développement web avec Django

- Outils de développement : Le framework de tests unitaires.  
Quelques commandes utiles : dumpdata/loaddata, dbshell, inspectdb, check...  
Internationalisation  
Fichier data  
Configuration Dev / Prod  
Création d'une commande personnalisée
- API REST : Créer un API REST avec django-rest-framework  
Authentification



## XML Développement

**Durée:** 3 jours  
1650 €

19 au 21 février  
14 au 16 mai

24 au 26 septembre  
26 au 28 novembre

### Public:

Développeurs, concepteurs, chefs de projet.

### Objectifs:

Maîtriser les bases du langage XML, connaître les grands principes du méta-langage.

### Connaissances préalables nécessaires:

Connaissances générales des systèmes d'information.

### Programme:

Introduction à XML : Historique. Principes du langage. Concepts d'XML.  
Principaux domaines d'application

Règles d'écriture d'un document XML : Structures d'un document XML. Problématiques liées à l'encodage  
Les espaces de noms (namespace). XHTML

Les grammaires XML : Les DTD (Document Type Definition). Définition. Les éléments et les attributs  
Les schémas XML (XSD). Définition. Les éléments et les attributs.  
Découpage d'un schéma  
Les analyseurs de document XML ( parseur ) et les outils (XML-Spy?)

Le langage XSL : Les feuilles de style : css, XSL  
Principes du langage. Le langage XPATH. Le langage XSLT. Structure d'un programme XSLT  
Instructions XSLT : Template, Macros, ... Génération HTML depuis XML.  
Le langage XSL-FO.

Manipulation des documents XML avec Java : Présentation des technologies. Les interfaces DOM (Document Object Model), et SAX (Simple API for XML)

# Développement d'applications Android

**Durée:** 4 jours  
2040 €

12 au 15 février  
14 au 17 mai

27 au 30 août  
19 au 22 novembre

## 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 de la programmation en java.

## Programme:

- Introduction** : Présentation du système d'exploitation Android, des plate-formes matérielles, des outils : SDK, android market, etc .. Architecture : la couche noyau Linux, l'environnement de développement, machines virtuelles, base de données SQLite  
Les bibliothèques de base.
- Applicatif** : Les principales applications existantes. Installation/désinstallation d'une application.  
Arborescence des fichiers.
- Développement** : Présentation du SDK pour le développement en java, installation. Utilitaires : émulateurs, simulateur de carte.  
Développement d'une application de base.  
Structure générale des applications. Les quatre types de composants applicatifs : Activity, Services, Broadcast receivers, Content providers.  
Cycle de vie des composants.  
Les versions d'android et du SDK : propriétés, adaptation aux matériels (smartphones, tablettes, télévisions, etc ...)  
Les outils de développement : Android Studio IntelliJ,AVD : Android Virtual Device, ADB : Android Debug Bridge, DDMS : Dalvik Debug Monitor Server  
Développement d'une première application.



## Développement d'applications Android

- Interface utilisateur : Définitions. Présentation des layouts.  
Récupération du contexte applicatif. Les widgets.  
Gestion des menus. Boîtes de dialogue. Thèmes. Notifications (Toast, Status Bar, Dialog).
- Interaction d'applications : Présentation des Intents.  
Le fonctionnement des services. L'interrogation de WebServices RESTfull (client http, json)  
Les bonnes pratiques.
- Persistance des données : Stockage des préférences utilisateur. Le système de fichiers.  
Mise en oeuvre de SQLite.
- Graphique et multimédia : Développement 2D. Les APIs. Utilisation du MediaPlayer.
- API : Accès réseau, accès au système de fichiers.  
Capteurs internes. Gestion des périphériques : carte son, écran, caméra, clavier,..  
Mise en oeuvre de la classe Sensor.

## Filières Langages Java et JEE

Concepts objets et programmation Java SE 8 p204  
Programmation avancée Java SE 8 p207

Sécurité applications Java Jee p209  
Jee : développement web p211  
Jee : les EJB p213  
Développement Webservices p215

Serveurs d'application Jee p217  
Administration WebSphere p219  
Administration JBoss p221  
Administration Tomcat p224

Java EE avec JSF p226  
Persistance Java avec Hibernate p228  
Framework Spring p230



# Concepts Objets et programmation Java SE 8

LJ001

**Durée:** 5 jours  
2490 €

5 au 9 mars  
4 au 8 juin

8 au 12 octobre  
3 au 7 décembre

## Public:

Développeurs, ingénieurs logiciels et architectes d'applications.

## Objectifs:

Apprendre le langage Java et assimiler les concepts objets. Utiliser les outils du JDK et les principales API de la Standard Edition 8. Tous les concepts sont illustrés par des travaux pratiques : soit sur des exemples de base, soit sur des exercices plus complets. Un projet global permettant de mettre en oeuvre l'ensemble des concepts abordés est réalisé tout au long de la formation au fur et à mesure de l'acquisition des concepts.

## Connaissances préalables nécessaires:

Connaissance d'un langage de programmation structuré

## 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. Présentation JSE, du jdk. Les API de la SE 8.  
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.

## Concepts Objets et programmation Java SE 8

- Les classes : Présentation des concepts orientés objet (classe, attribut, constructeur, héritage, ..)  
 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.
- Apport des Design Pattern : Principes des solutions de conception cataloguées.  
 Méthodologie: définition des besoins techniques, des classes "types" du pattern, des collaborations entre classes.
- Les classes internes : Définition.  
 Caractéristiques principales. Déclaration. Exemples.
- Les exceptions et erreurs : Définition. Graphes d'héritage.  
 Mécanisme de gestion des exceptions.  
 Gestion des logs Java.
- Les structures de données : La classe Vector. La classe Stack.  
 L'interface Enumeration.  
 Structures de données ordonnées. Les collections.
- Les génériques : Définition. Exemples.



## Concepts Objets et programmation Java SE 8

Auto Boxing et  
Auto UnBoxing : Objectif. Exemples.

Les annotations : Définition. Annotations standards.  
Exemples.

Accès aux SGBD : 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.  
Présentation de JPA (Java Persistence API).

## Programmation avancée Java SE 8

**Durée:** 5 jours  
2520 €

12 au 16 mars  
11 au 15 juin

15 au 19 octobre  
17 au 21 décembre

### Public:

Développeurs java, ingénieurs logiciels et architectes d'applications.

### Objectifs:

Approfondir la connaissance de Java notamment dans les domaines de la programmation multi-tâches, des I/O, des tests et du logging des technologies jdbc et de la persistance avec JPA.

### Connaissances préalables nécessaires:

Maîtriser les concepts objets et les bases du langage Java, ou avoir suivi le stage « Concepts Objets et bases Java »

### Programme:

- Java SE : Présentation des et rappels sur les classes Java.  
Les interfaces et les expressions lambda.
- Collections : Création de collections : ArrayList, TreeSet, HashMap, etc ...  
L'interface Stream.  
Filtrage de collections avec les expressions Lambda.
- Le multi-threading : Fonctionnement.  
Ordonnancement et priorité.  
Exclusion mutuelle. Synchronisation.  
Thread démon. Communication par flux "pipe".
- L'API de concurrence : Les exécuteurs de tâches. Les queues. Les maps atomiques. La représentation du temps et de ses unités. Les synchroniseurs.  
Les traitements asynchrones anticipés. Les variables atomiques. Les verrous "haute performance".
- Les annotations : Objectif. l'API Reflection. Annotations standards. Les méta-annotations.  
Fabriquer ses annotations. Annotation Processing Tool (APT)

## Programmation avancée Java SE 8

- La gestion des I/O : La gestion des flux standards.  
Lecture/écriture depuis la console  
Utilisation des streams  
API d'accès aux fichiers (NIO.2)
- 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).
- Les tests : Objectif. Le framework JUnit.
- Traçabilité des applications : Objectif. L'API Java Logging.  
Gestion des dates avec l'API Java Date/Time  
Création de timestamps.

## Sécurité Java et JEE

**Durée:** 3 jours  
1650 €

19 au 21 mars  
22 au 24 mai

5 au 7 septembre  
5 au 7 novembre

### Public:

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

### Objectifs:

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

### Connaissances préalables nécessaires:

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

### Programme:

- Besoins** : Les risques  
Politique de Sécurité  
Evaluation des risques en fonction des différents modes d'utilisation de Java (applets, application, servlets)
- Sécurisation de la JVM** : Limites naturelles imposées par Java : gestion mémoire.  
Contrôle du bytecode par la machine virtuelle.  
Mise en oeuvre du SecurityClassLoader
- Protection de l'exécution** : Exécution protégée : SecurityManager, ClassLoader.  
Surcharge des méthodes d'accès s: lecture, écriture, exécution, ouverture de socket, autorisation de connexions...  
TP : Protection des accès sur le disque local d'une application.
- Chiffrement** : Les mécanismes de signature. Création de clés publiques et privées.  
Les clés RSA, DSA.  
Signature d'un document.  
Les algorithmes SHA1withDSA, MD5withRSA.  
Les MessageDigest. Les algorithmes MD2, MD5, SHA-1, SHA-512  
TP : Vérification de l'authenticité d'un document
- Certificats** : Cycle de vie d'un certificat. La fabrique de certificats Java.  
Les certificats de modification X509.



## Sécurité Java et JEE

- 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 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  
1550 €

27 au 29 mars  
29 au 31 mai

2 au 4 octobre  
11 au 13 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 JavaBeans en respectant le modèle MVC. Assurer la persistance en utilisant JDBC.

### Connaissances préalables nécessaires:

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

### Programme:

- L'API JDBC : Objectif de JDBC. Les types de drivers. Les architectures applicatives. Les classes et interfaces en jeu. Connexion. La gestion des transactions et l'isolation transactionnelle. Interrogation et mise à jour. Appel d'une procédure stockée. Les types de données. Les pools de connexion. Les Rowset. La libération des ressources. La gestion des exceptions. Présentation de JPA (Java Persistence API).
- Architecture en couches : Présentation, Métier et Persistance. Couplage fort, couplage faible. Les Design Patterns nécessaires : Singleton, Factory, Façade, Iterator. Notion de composant. Découpage du composant en 3 couches (service, donnée, persistance). Mapping opérationnel pour la persistance. L'implémentation de la persistance avec JDBC.
- La couche Présentation : Servlet, JSP et Taglib. Design Pattern MVC
- Servlet (le contrôleur) : Objectif. Le protocole HTTP. L'API Servlet. Cycle de vie d'une servlet. Gestion de contexte. Gestion de la requête client. Gestion des cookies. Redirection côté client et côté serveur. Configuration et déploiement.



## **JEE : Développement d'applications web**

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 Controleur (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.



## **JEE : les EJB**

- Les EJB entité et la norme JPA : Norme JPA (Java Persistence API) et configuration.  
Gestionnaire de persistance.  
Relation avec les graphes d'objets.  
Les mécanismes d'héritage.
- Le langage EJB-QL (Query Language) : Les types de requêtes.  
Jointure et restrictions.
- Gestion des transactions : Les différents modèles transactionnels.  
Gestion des transactions distribuées.  
Transactions de niveau conteneur ou bean.  
Transactions au niveau du client.
- Gestion de la sécurité : Sécurité Jee avec JAAS.  
Sécurité par programmation.  
Sécurité déclarative.
- 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  
1720 €

21 au 23 février  
28 au 30 mai

24 au 26 octobre  
12 au 14 décembre

## Public:

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

## Objectifs:

Savoir développer des applications utilisant les techniques des Web services.

## Connaissances préalables nécessaires:

Ce cours présente les WebServices en environnement Java.

## Programme:

Introduction : Historique. Définitions.  
Les différents types de webservices : ws-\* et RESTful.  
Les EJB3 et annotations Java pour créer des webservices.

Le protocole SOAP : Présentation : Simple Object Access Protocol pour l'échange de messages XML.  
L'interopérabilité avec SOAP, les avantages. Structure d'un message.Exemples.

WSDL : Définition. Structure d'un document WSDL. Définition d'un service. Gestion de la sécurité.

Les annuaires UDDI: Universal Description, discovery and Integration pour la recherche des services web disponibles.  
Les annuaires publics. Structure des données. Mise en oeuvre de jUDDI. Publication d'un WebService.

Les API Java pour XML : JAXP, JAXB, SAAJ : pour le traitement des données XML  
JAX RPC, lancement de procédures distantes  
JAXM, messages XML; JAXR, identification de services web  
Le WSDP : Java Web Services Developer Pack.

WebServices et Axis: Présentation Axis. Principe de fonctionnement. Mise en oeuvre.  
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.



## Serveurs d'application JEE

- Développement avec les EJB** : Les EJB : spécifications (état actuel et limites).  
EJB entité, EJB session, EJB Message-driven. Apports des EJB3.  
Jointures. Transactions. Présentation de EJB-QL. Relations entre le développeur d'EJB et le DBA.
- Intégration** : Transactions. Utilisation de JTA, JTS. Transactions explicites.  
Transactions gérées par container.  
Sécurité : JAAS, rôles, groupes de permissions.  
Mécanismes de connexion aux bases de données : JDBC, pool de connexions.  
Connexion interapplicatives avec JCA.
- 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.

## Administration WebSphere

**Durée:** 5 jours  
2680 €

22 au 26 janvier  
9 au 13 avril

2 au 6 juillet  
22 au 26 octobre

### 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.



## Administration WebSphere

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

# Administration Tomcat

**Durée:** 3 jours  
1610 €

5 au 7 février  
14 au 16 mai

10 au 12 septembre  
26 au 28 novembre

## Public:

Toute personne souhaitant configurer et administrer une application avec Tomcat.

## Objectifs:

Comprendre le fonctionnement de Tomcat, et savoir le mettre en oeuvre, l'installer, le configurer et l'administrer, optimiser le fonctionnement du serveur. 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** : Travaux pratiques : installation et configuration des droits d'accès dans le fichier tomcat-users.xml  
Fonctions disponibles : liste des ports écoutés par tomcat pour les requêtes http, applications déployées sur chaque hôte virtuel, configuration des connecteurs, de la sécurité, (utilisateurs, groupes, rôles), des sources de données.
- Sécurité** : 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
- Fichiers de configuration** : Etude des paramètres à positionner dans le fichier server.xml, les balises server, engine, host, context, logger, loader.  
Le descripteur de déploiement web.xml : déclaration de la servlet, lien entre la servlet et la requête.  
Configuration des sources de données : présence des pilotes, optimisation des paramètres de connexion.  
Travaux pratiques : mise en place d'un pilote pour l'accès à une base postgresSQL.  
Les connecteurs : la balise connector. Le connecteur HTTP, fonctionnement Tomcat en standalone.  
Sécurisation : configuration du protocole https.
- Performances et tests** : 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.

## 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  
1690 €

19 au 21 février  
22 au 24 mai

17 au 19 septembre  
19 au 21 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.
- 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.

## Administration JBoss

- 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.
- 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.
- 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.



## Java EE avec JSF

**Durée:** 3 jours  
1550 €

6 au 8 juin

21 au 23 novembre

### Public:

Architectes, chefs de projets, concepteurs, développeurs, ingénieurs

### Objectifs:

Etre capable de développer une application web en intégrant une interface utilisateur basée sur la technologie standard JSF (Java Server Faces) intégrée à Java EE

### Connaissances préalables nécessaires:

Il est demandé aux participants de connaître la programmation Java, et le développement Java EE pour le web (JSP/servlet), et les fondamentaux internet et HTML

### Programme:

Java Server Faces, : Caractéristiques essentielles  
les bases Les implémentations : JSF-RI, MyFaces, IBM-JSF...  
Configuration logicielle requise

La technologie JSF : Les éléments d'une application JSF  
Découverte du fonctionnement à travers un exemple simple  
Les étapes du développement  
Création des pages web, définition de la navigation, développement des beans métier, configuration des beans  
Le modèle de composant de l'interface utilisateur, concept de 'render'  
Le modèle événementiel, validation des données  
Le modèle de navigation, gestion des 'Backing beans'  
Vue d'ensemble et intégration des éléments JSF  
Cycle de vie d'une page JSF

Mise en oeuvre des : Configuration de la page  
pages JSF La bibliothèque des balises de base (JSF Core Tags)  
Les balises de composants HTML, les classes associées  
La gestion de l'internationalisation  
Gestion des messages personnalisés  
Les conversions de données, utilisation d'un converteur personnalisé  
Les validateurs de données, un validateur personnalisé  
La gestion des listeners, référence à une méthode de Backing Bean

## Java EE avec JSF

Création de composants personnalisés d'interface utilisateur : Concevoir un composant personnalisé, composition, facelets

Configuration des applications JSF : Le descripteur de déploiement : web.xml  
Le fichier de configuration de ressource : faces-config.xml  
Configuration des règles de navigation

Bibliothèques : OpenFaces, RichFaces, ADF Faces, PrimeFaces



## Persistence Java avec Hibernate

**Durée:** 3 jours  
1550 €

31 janvier au 2 février  
23 au 25 avril

21 au 23 novembre

### Public:

Architectes, chefs de projets, concepteurs, développeurs, ingénieurs

### Objectifs:

Savoir développer une application Java EE en réalisant la persistance des objets avec le framework Hibernate

### Connaissances préalables nécessaires:

Il est demandé aux participants de connaître la programmation Java.

### Programme:

- Concepts de base : Définition de la persistance objet  
Classes, fichier de mapping, configuration, cycle de vie des objets  
Mappage des associations unidirectionnelle/bidirectionnelle et des collections
- L'architecture d'Hibernate : Vue en couche, cycle de vie des instances  
Intégration JNDI, JDBC, JTA, gestion des sessions
- Configuration : APIs permettant la configuration par programmation  
Ouverture des sessions, gestion des connexions JDBC  
Dialectes SQL, cache de second niveau et cache de requêtes,  
Fichier de configuration XML, intégration à un serveur d'application Java EE
- Réalisation de classes persistantes : APIs permettant la configuration par programmation  
Ouverture des sessions, gestion des connexions JDBC  
Dialectes SQL, cache de second niveau et cache de requêtes,  
Fichier de configuration XML, intégration à un serveur d'application Java EE
- Les bases du Object Relational Mapping : Les déclarations de mapping, les types utilisés par Hibernate  
Plusieurs mappings pour une même classes  
Les annotations JAVA 5

# Persistence Java avec Hibernate

Etablir la persistance des collections : Les types de collections persistantes, les clés étrangères, les éléments et les index d'une collection  
Associations \*.\* et 1..\*

Etablir la persistance des associations : Associations unidirectionnelles, bidirectionnelles, mappings complexes

Etablir la persistance des relations d'héritage : Différentes stratégies, utilisation d'un discriminant, gestion du polymorphisme  
Comparaison des méthodes, avantages et inconvénients, dans quels cas les utiliser ?

Hibernate Query Language (HQL) et gestion des objets persistants : Stocker et charger un objet  
Syntaxe du langage HQL, les clauses from, select, where, order by, group by  
Jointures, agrégation, requêtes polymorphiques, les requêtes imbriquées  
Les bonnes pratiques, externalisation de requêtes nommées, chargement tardif des propriétés  
Gestion du cache de second niveau, le cache de requêtes



## Framework Spring : mise en oeuvre

**Durée:** 3 jours  
1550 €

9 au 11 avril

12 au 14 novembre

### Public:

Architectes, chefs de projets, concepteurs, développeurs, ingénieurs

### Objectifs:

Savoir développer une application JavaEE en utilisant Spring,

### Connaissances préalables nécessaires:

Il est demandé aux participants de connaître la programmation Java.

### Programme:

- Introduction : Concepts de conteneur léger, vue d'ensemble et exemples d'utilisation  
Pattern 'Inversion de Contrôle (IoC) par Injection de dépendance'
- Mise en oeuvre : le conteneur léger de Spring : Vue d'ensemble du framework, les Beans, BeanFactory et ApplicationContext  
Modes singleton ou normal, méthodes d'injection de dépendance  
Configuration de beans spécifiques à Spring, cycle de vie  
Définition de bean abstrait et héritage, injections automatiques  
APIs d'accès aux ressources, techniques de validation (PropertyEditors, DataBinder, BeanWrapper)
- Framework MVC Web spécifique à Spring : Orientations MVC2, rappels, configuration du serveur  
Les contrôleurs, les gestionnaires de vues, les mappings  
Spring WebFlow, intégration avec JSF et Struts
- Pattern DAO, Accès aux données avec JDBC: Classes abstraites pour le support DAO  
Utilisation de JDBC via les APIs d'encapsulation Spring  
Persistance des composants, gestion des ressources  
Configuration de SessionFactory dans le contexte d'application Spring  
IoC : HibernateTemplate et HibernateCallback, implémentation DAO avec les APIs Hibernate  
Démarcation de transactions par programmation et déclaration

# Framework Spring : mise en oeuvre

- Programmation Orientée Aspect (AOP) : Les concepts  
La terminologie utilisée : Aspect, Joinpoint, Advice, Pointcut, Introduction, Target object, AOP Proxy, Weaving  
Les différents types « advice » : Around, Before, Throws, After  
Les objectifs et fonctionnalités AOP de Spring. Intégration avec le framework AspectJ
  
- Gestion des transactions : La couche d'abstraction Spring  
Les stratégies de transaction : par programmation ou par aspects  
Mise en oeuvre d'une gestion transactionnelle sans programmation et par les aspects





## 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<sup>e</sup> jour ouvré au 2<sup>e</sup> 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](http://www.Pythagore-fd.fr)



## Table des matières

Pythagore F.D. : Apprendre à Apprendre		
Filières BigData		
BigData : concepts et enjeux	CB000	5
BigData Architecture et technologies	CB001	7
Introduction à NoSQL	CB002	9
Stockage distribué avec Ceph	CB004	11
BigData avec Cassandra	CB010	12
Base de données NoSQL avec MongoDB	CB017	14
Hadoop : l'écosystème	CB030	15
Hadoop Hortonworks : administration avec Ambari	CB031	17
Hadoop Cloudera : administration	CB032	19
Hadoop : développement	CB033	22
Hadoop : stockage avec HBase	CB034	24
Hadoop : infrastructure sécurisée	CB035	26
Hadoop : analytics	CB036	27
Spark Mise en oeuvre et programmation	CB037	29
Flux de données avec Storm	CB038	31
Programmation Scala	CB039	32
Objets connectés:des OS embarqués au cloud	CB100	33
Blockchain : principes et technologies	CB200	34
ElasticSearch : indexation	CB020	35
ElasticSearch : mise en oeuvre et programmation	CB021	36
ElasticSearch : infrastructure et administration	CB022	38
Filières bases de données SQL		
Langage SQL	BD001	42
MySQL : Exploitation	BD012	44
PostgreSQL : Administration	BD021	46
PostgreSQL :administration avancée	BD022	48
Filières Systèmes Unix/Linux		
Linux/unix introduction	UX100	50
Le Shell	UX002	52
Atelier : Shell avance	UX004	53
Administration Linux	UX111	54
Administration avancée Linux	UX140	58
Les services réseaux Linux	UX112	61
Haute disponibilité Linux	UX115	63
Linux système sécurisé	UX117	65
Linux sécurité des accès	UX118	68
Linux : optimisation performances métrologie	UX119	71
Administration Ubuntu Server	UX142	73
Preparation certification LPI101	UC101	76
Preparation certification LPI102	UC102	78
Préparation certification LPI201	UC201	80
Preparation certification LPI202	UC202	82
Administration AIX	UX018	83
Administration AIX avancée	UX019	85
Administration Solaris 11	UX040	87
Administration avancée Solaris	UX041	89
Systèmes Linux embarqués	UX125	91
Filières virtualisation, cloud et orchestration		
Virtualisation Linux	SY011	94
Virtualisation avec Xen	SY004	96
Virtualisation avec KVM	SY007	98
Virtualisation avec lxc	SY008	100

Docker : mise en oeuvre	SY009	102
Réseaux virtuels avec OpenvSwitch	SY012	104
Cloud : technologies et enjeux	SY100	105
Architecture cloud d'entreprise	SY101	107
OpenStack : configuration et administration	SY111	109
CloudStack : mise en oeuvre	SY112	112
Cloud d'entreprise avec OpenNebula	SY113	114
Cobbler : gestion de configurations	SY123	116
Ansible : industrialiser les déploiements	SY124	118
Puppet : administration centralisée	UX161	119
Puppet : expertise	UX162	122
Gestion de configuration avec Chef	UX170	124
AWS : architecture SI, réseaux virtuels	SY200	126
AWS : stockage,S3,dynamodb..	SY201	128
AWS : les API EC2 et S3	SY202	130
Filières Réseaux et TCP/IP		
Introduction aux réseaux	RS001	134
TCP/IP : protocoles et mise en oeuvre	RS003	136
IP: Migration vers IPV6	RS014	138
Messagerie	RS006	141
Sécurité TCP/IP	RS018	142
Mise en oeuvre du protocole SNMP	RS022	144
Voix sur IP	RS023	146
Annuaire LDAP	RS122	149
Filière Production et supervision		
Supervision nagios : utilisation	RS129	153
Administration Nagios	RS130	155
Supervision avec shinken	RS135	158
Zabbix administration	RS150	160
Gestion de Parc avec OCS et GLPI	UX124	162
Gestion de versions avec GIT	UX151	164
Filières Internet et Web		
Serveur WEB : apache	RS104	167
Administration serveur Nginx	RS105	169
Serveur Proxy Squid	RS114	171
Sécurisation des services internet	RS115	173
Développement Web HTML et CSS	RS102	176
Web Dynamique avec JavaScript	RS106	178
HTML5 CSS3 et javascript	RS107	180
Javascript avec JQuery	RS108	182
Angular JS développement	RS110	184
Node JS mise en oeuvre	RS111	186
Développement Web avec GWT	LW001	188
Filières Développement		
Programmation en langage C	LC001	191
Perfectionnement en langage C	LC002	193
Programmation système en C sur Unix/Linux	LC010	195
Programmation noyau et drivers en C sur Linux	LC011	198
Développement d'applications Web avec PHP	LH001	199
Programmation Perl	LP001	201
Le langage Python	LY001	203
Développement avancé avec Python	LY002	205
Développement web avec Django	LY003	207
XML Développement	AS120	209
Développement d'applications Android	UX128	210
Filières Langages Java et JEE		



Concepts Objets et programmation Java SE 8	LJ001	213
Programmation avancée Java SE 8	LJ002	216
Sécurité Java et JEE	LJ005	218
JEE : Développement d'applications web	AS004	220
JEE : les EJB	AS005	222
Développement Web Services	AS006	224
Serveurs d'application JEE	AS011	226
Administration WebSphere	AS131	228
Administration Tomcat	AS133	230
Administration JBoss	AS134	233
Java EE avec JSF	AS151	235
Persistence Java avec Hibernate	AS152	237
Framework Spring : mise en oeuvre	AS153	239

