|  |  |  |
| --- | --- | --- |
| **Domaine** | **Techno** | **Documentations** |
| **OUTILS TRES INTERESSANT** |  | <http://www.tutorialspoint.com/index.htm>  <http://www.pdftowordconverter.net/fr.aspx>  <https://www.draw.io/>  <http://www.graphviz.org/> |
| **INTERVIEW !!!!** |  | <http://www.tutorialspoint.com/java/java_interview_questions.htm>  <http://www.javatpoint.com/corejava-interview-questions>  http://www.javatpoint.com/hadoop-interview-questions |
| **Veille techno** | **Radar** | <https://www.thoughtworks.com/radar> |
| **Java J2E** | **Technos** | GC: <http://www.jmdoudoux.fr/java/dej/chap-gestion_memoire.htm>,  <http://jobprod.com/le-fonctionnement-du-garbage-collector/>,  <http://patatos.over-blog.com/article-comment-fonctionne-le-garbage-collector-de-la-jvm-51105952.html>  Gestion mémoire, type des objets en mémoire: <http://blog.sodifrance.fr/java-et-la-gestion-de-la-memoire/>  type des objets: <http://www.commentcamarche.net/contents/564-java-les-types-de-donnees>, <http://www.vulgarisation-informatique.com/java-types-donnees.php>  Mots réservés Java: <http://baptiste-wicht.developpez.com/tutoriels/java/mots-reserves/>  Polymorphisme: <http://www.commentcamarche.net/contents/811-poo-le-polymorphisme>,  <https://fr.wikibooks.org/wiki/Programmation_Java/Polymorphisme>,  <https://fr.wikipedia.org/wiki/Polymorphisme_(informatique)>  Encapsulation,héritage: <http://www.chicoree.fr/w/Encapsulation,_h%C3%A9ritage_et_polymorphisme>, <https://fr.wikibooks.org/wiki/Programmation_Java/Encapsulation>  Classes abstraites: <https://fr.wikibooks.org/wiki/Programmation_Java/Classes_abstraites>  Interface: <https://fr.wikibooks.org/wiki/Programmation_Java/Interfaces>  Classe interne: <https://fr.wikibooks.org/wiki/Programmation_Java/Classes_internes>  transtypage: <https://fr.wikibooks.org/wiki/Programmation_Java/Transtypage>  Type générique: <https://fr.wikibooks.org/wiki/Programmation_Java/Types_g%C3%A9n%C3%A9riques>  Instanciation: <https://fr.wikibooks.org/wiki/Programmation_Java/Instanciation_et_cycle_de_vie>  Collections MAP: <https://openclassrooms.com/courses/apprenez-a-programmer-en-java/les-collections-d-objets> |
| **Analyses mémoire** | <http://www.infini-software.com/Encyclopedie/Developpement/Infini/Ellipse/French/HowTo_DetectMemoryLeaks.wp>  <http://www.javaperformancetuning.com/tools/gcviewer/> |
| **Architectures micro services** | <http://blog.xebia.fr/2015/03/09/microservices-des-architectures/>  <http://blog.xebia.fr/2015/03/02/microservices-les-concepts/>  <https://fr.wikipedia.org/wiki/Enterprise_service_bus>  <http://blog.inovia-conseil.fr/?p=155>  <http://microservices.io/patterns/microservices.html>  <http://blog.inovia-conseil.fr/?p=155>  <http://eugenedvorkin.com/seven-micro-services-architecture-problems-and-solutions/>  <http://www.javaworld.com/article/2683277/architecture-scalability/what-microservices-architecture-really-means.html> |
| **Caches** | <https://labs.consol.de/java-caches/>  <http://memcached.org/>  <http://www.ehcache.org/> |
| **IBM Web Sphere** | <https://www-01.ibm.com/support/knowledgecenter/SSAW57_8.0.0/com.ibm.websphere.nd.doc/info/ae/ae/covr_arch.html> |
| **Base de données** | **Transaction** | <https://fr.wikipedia.org/wiki/Transaction_informatique> |
| **Jointure** | <http://sql.sh/cours/jointures>  <https://openclassrooms.com/courses/introduction-aux-jointures-sql> |
| **Index** | <http://www.dbnewz.com/2008/06/27/les-index-mysql-types-placements-efficacite/comment-page-1/>  <https://fr.wikipedia.org/wiki/Index_(base_de_donn%C3%A9es)>  <http://cerig.pagora.grenoble-inp.fr/tutoriel/bases-de-donnees/chap03.htm> |
| **Architecture** | **LAMBDA Architecture** | <http://lambda-architecture.net/> <https://www.manning.com/books/big-data#downloads>  <http://blog.xebia.fr/2014/05/07/devoxxfr-lambda-architecture-choose-tools-for-real-time-big-data/>  <http://lambda-architecture.net/>  <https://www.manning.com/books/big-data>  <http://www.drdobbs.com/database/applying-the-big-data-lambda-architectur/240162604> |
| **Architecture et technologies** | <http://blog.octo.com/category/architecture-et-technologies/> |
| **Langage de programmation** | **Scala** | <http://logic.cse.unt.edu/tarau/teaching/scala_docs/scala-for-the-impatient.pdf> |
| **TOUT SUR JAVA** | <https://fr.wikibooks.org/wiki/Programmation_Java> |
| **Java 8 et 7** | <http://www.infoq.com/fr/articles/java-8-vs-scala>  <http://programmers.stackexchange.com/questions/193630/summary-of-differences-between-java-versions>  <http://javarevisited.blogspot.fr/2014/04/10-jdk-7-features-to-revisit-before-you.html>  <http://blog.takipi.com/5-features-in-java-8-that-will-change-how-you-code/>  <http://www.ibm.com/developerworks/library/j-java8lambdas/index.html>  <http://blog.netapsys.fr/java-8-lere-des-expressions-lambda/> |
| **DEVOPS** | **Déploiement automatique** | <http://blog.octo.com/devops-de-lintegration-continue-au-deploiement-continu/> |
| **Concepts** | <http://blog.xebia.fr/wp-content/uploads/2015/05/Programmez-DevOps.pdf>  <http://blog.octo.com/devops/>  <https://try.newrelic.com/rs/newrelic/images/NewRelic-Kickstarting-Devops-eBook.pdf>  <http://thenewstack.io/six-core-capabilities-of-a-devops-practice/>  <http://bfmbusiness.bfmtv.com/01-business-forum/devops-les-3-concepts-fondamentaux-621600.html>  <https://www.youtube.com/watch?v=bTGj4PeRPjs> |
| **Docker** | <http://douche.name/presentation-docker/>  <http://www.journaldunet.com/solutions/cloud-computing/docker-definition-avantages-inconvenients.shtml>  <http://sametmax.com/le-deploiement-par-conteneurs-avec-docker/>  <http://blog.ippon.fr/2014/04/14/docker-presentation-part-1/>  [http://blog.nicolargo.com/2014/06/virtualisation-legere-docker.html](http://fr.hortonworks.com/hadoop/) |
| **« Application Portfolio Management » (APM), ou « Gestion de Portefeuille Applicatif » (GPA)** | **CENTREON**  **NAGIOS**  **APPDYNAMICS** | comparaison:  <http://www.fromdev.com/2015/03/application-performance-management-tools.html>  <https://assets.nagios.com/datasheets/compare/How_Nagios_Compares_To_Centreon.pdf>  <http://decrypt.ysance.com/2011/02/comparatif-outils-monitoring-metrologie-supervision-2-zabbix-centreon-nagios-cacti-munin/>  AppDynamics:  <https://docs.appdynamics.com/display/PRO14S/Architecture>  <https://www.appdynamics.com>  <http://www.redsen-consulting.com/fr/inspired/architecture-entreprise/demarche-apm>  <http://www.visualware.com/resources/technical/apm_paper0404.pdf>  <https://msdn.microsoft.com/en-us/library/bb896054.aspx>  <https://books.google.fr/books?id=UHWjgcn3BsQC&pg=PA265&lpg=PA265&dq=architecture+APM&source=bl&ots=Ea8byf508O&sig=KcebbtZ-9zWOA5V5kZOwELY7PO0&hl=fr&sa=X&ved=0ahUKEwjJ1Ij7_MTKAhXEWiwKHXDxD1c4ChDoAQhUMAg#v=onepage&q=architecture%20APM&f=false>  <http://www.lemondeinformatique.fr/actualites/lire-silicon-valley-2014-appdynamics-une-solution-apm-taillee-pour-le-cloud-et-la-virtualisation-57886.html>  Solutions BigData pour APM  <http://www.ictjournal.ch/fr-CH/News/2014/07/01/Les-cinq-fonctions-cles-dune-solution-Big-Data-pour-APM.aspx> |
| **Big Data** | **CODE !!!!** | <https://github.com/sequenceiq> |
| **INTRODUCTION IMPORTANT** | <https://www.youtube.com/watch?v=xYnS9PQRXTg>  <http://fr.slideshare.net/Datadopter/the-three-generations-of-big-data-processing>  <http://blog.octo.com/hadoop-summit-2014-un-compte-rendu-partie-13/>  <http://www.thomas-robert.fr/mon-resume-des-concepts-du-big-data/>  <https://www.cetic.be/Installer-en-3-etapes-un-cluster> |
| **LESTOUTES TECHNOLOGIES BIG DATA** | <http://www.bigdata-careers.com/?page_id=99> |
| **Replication HDFS géographique** | <http://fr.slideshare.net/Hadoop_Summit/selective-data-replication-with-geographically-distributed-hadoop>  <http://allthingshadoop.com/2014/12/29/multi-datacenter-replication-with-apache-kafka/>  <http://www.datastax.com/wp-content/uploads/2012/09/WP-DataStax-MultiDC.pdf> |
| **Lien avec le digital** | <http://www.smart-seven.fr/big-data-au-service-du-marketing-digital-definitions/> |
| **Comparaison distribution** | <https://www.youtube.com/watch?v=WRfMrwyniqQ>  <http://www.networkworld.com/article/2369327/software/comparing-the-top-hadoop-distributions.html>  <http://www.journaldunet.com/solutions/saas-logiciel/comparatif-4-distributions-hadoop/tableau-de-synthese.shtml>  [**https://www.experfy.com/blog/cloudera-vs-hortonworks-comparing-hadoop-distributions/**](https://www.experfy.com/blog/cloudera-vs-hortonworks-comparing-hadoop-distributions/)  [**http://www.infoq.com/fr/articles/BigDataPlatform**](http://www.infoq.com/fr/articles/BigDataPlatform) |
| **HDFS architecture** | <http://hortonworks.com/hadoop/hdfs/#section_1>  <https://hadoop.apache.org/docs/r1.2.1/hdfs_design.html> |
| **Comparaison Mongo Hbase Cassendra** | IMPORTANT: <http://fr.slideshare.net/EdurekaIN/no-sql-databases-35591065>  <http://www.datastax.com/nosql-databases/benchmarks-cassandra-vs-mongodb-vs-Hbase>  <http://www.infoworld.com/article/2848722/nosql/mongodb-cassandra-hbase-three-nosql-databases-to-watch.html>  <http://kkovacs.eu/cassandra-vs-mongodb-vs-couchdb-vs-redis>  <http://db-engines.com/en/system/Cassandra%3BHBase%3BMongoDB>  <http://www.planetcassandra.org/nosql-performance-benchmarks/> |
| **Zeppelin** | <https://github.com/NFLabs/z-manager> |
| **Hadoop**  **Comment dimensionner un cluster**  **comment ça marche** | <http://fr.hortonworks.com/hadoop/>  <http://blog.octo.com/hadoop-dans-ma-dsi-comment-dimensionner-un-cluster/>  <https://dzone.com/articles/how-hadoop-mapreduce-works> |
| [**S**](http://fr.hortonworks.com/hadoop/)**tack ELK** | <http://blog.ippon.fr/2014/10/28/un-elk-pour-les-gouverner-tous-les-logs/>  <http://www.oxalide.com/2015/06/hands-on-workshop-elasticsearch-stack-elk-quoi-de-neuf/>  <http://www.rittmanmead.com/2014/10/monitoring-obiee-with-elasticsearch-logstash-and-kibana/>  <http://blog.xebia.fr/2013/12/12/logstash-elasticsearch-kibana-s01e02-analyse-orientee-business-de-vos-logs-applicatifs/>  <http://blog.xebia.fr/2013/12/06/logstash-elasticsearch-kibana-s01e01-analyse-de-reputation-sur-twitter/>  <https://wooster.checkmy.ws/2014/04/elk-elasticsearch-logstash-kibana/> |
| **Talend** | <https://fr.talend.com/> |
| [**Hadoop MapReduce**](https://fr.talend.com/) | <http://www.journaldunet.com/developpeur/outils/les-solutions-du-big-data/principe-de-fonctionnement-de-mapreduce.shtml>  TRES IMPORTANT <http://blog.soat.fr/2015/05/comprendre-mapreduce/>  COMMENT CA MARCHE <http://www.bart-konieczny.com/fr/blog/hadoop/fonctionnement-du-mapreduce>  <http://martinfowler.com/articles/collection-pipeline/map.html>  <http://www.supergloo.com/fieldnotes/apache-spark-examples-of-transformations/>  http://ippon.developpez.com/tutoriels/big-data/apache-flink-park/redondance/ |
| [**Spark**](https://fr.talend.com/) | <http://spark.apache.org/docs/latest/programming-guide.html>  <https://www.quora.com/What-is-the-difference-between-Apache-Spark-and-Apache-Hadoop-Map-Reduce>  <https://www.mapr.com/blog/apache-spark-vs-mapreduce-whiteboard-walkthrough>  <https://www.xplenty.com/blog/2014/11/apache-spark-vs-hadoop-mapreduce/>  IMPORTANT <http://www.infoq.com/fr/articles/apache-spark-introduction>  RDD et DF <https://ogirardot.wordpress.com/2015/05/29/rdds-are-the-new-bytecode-of-apache-spark/>  Avoid GroupByKey <https://databricks.gitbooks.io/databricks-spark-knowledge-base/content/best_practices/prefer_reducebykey_over_groupbykey.html> |
| **Solution bigData** | <http://www.journaldunet.com/developpeur/outils/les-solutions-du-big-data/> |
| **Kafka** | IMPORTANT <http://fr.slideshare.net/carolineboison/apache-kafka-40973171>  <http://www.timeislife.eu/2015/08/07/big-data-kafka-concepts-generaux/>  <http://www.infoq.com/fr/news/2014/01/apache-afka-messaging-system>  <https://www.linkedin.com/pulse/flume-kafka-real-time-event-processing-lan-jiang> |
| **Hortonworks** | <http://fr.slideshare.net/hortonworks/hortonworks-technical-workshop-hdp-everywhere-cloud-considerations-using-cloudbreak-2015-june>  <http://hortonworks.com/hadoop/cloudbreak/> |
| [**Différence HBASE HDFS**](http://hortonworks.com/hadoop/cloudbreak/) | <http://fr.slideshare.net/enissoz/hbase-and-hdfs-understanding-filesystem-usage>  <http://fr.slideshare.net/cloudera/tokyo-nosqlslidesonly> |
|  | [**Comparaison Flume et Kafka**](http://hortonworks.com/hadoop/cloudbreak/) | <https://www.quora.com/What-are-the-most-significant-differences-between-Flume-and-Kafka>  [**Flume**:](http://hortonworks.com/hadoop/cloudbreak/)   * [Data is pushed to the the destination.](http://hortonworks.com/hadoop/cloudbreak/) * [Built around Hadoop ecosystem for the primary purpose of sending messages to HDFS & HBase](http://hortonworks.com/hadoop/cloudbreak/) * [Flume does not replicate events - in case of flume-agent failure, you will lose events in the channel](http://hortonworks.com/hadoop/cloudbreak/)   [**Kafka**:](http://hortonworks.com/hadoop/cloudbreak/)   * [A general purpose distributed publish-subscribe messaging system - you can have a topic which many listeners can subscribe to so the processing of messages can happen in parallel on various channels.](http://hortonworks.com/hadoop/cloudbreak/) * [Can be used for any system to connect to other systems that requires enterprise level messaging (website activity tracking, operational metrics, stream processing etc)](http://hortonworks.com/hadoop/cloudbreak/) * [High availability of events(recoverable in case of failures)](http://hortonworks.com/hadoop/cloudbreak/)   <http://blog.cloudera.com/blog/2014/11/flafka-apache-flume-meets-apache-kafka-for-event-processing/> |