Quick\_Ref: Pig

|  |  |  |
| --- | --- | --- |
| **S.No** | **Topic** | **Desc** |
|  | **Weblink** |  |
|  | Pig Admin | <https://pig.apache.org/docs/r0.14.0/admin.html> |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  | **General Information** |  |
| 1 | About Pig | Pig is a MapReduce based tool to process both structured and un-structured data. Pig is termed as data flow language and its syntax and semantic resembles functional programming languages like python, perl, etc. |
| 2 | Hive vs Pig | 1. Hive is like SQL 2. Pig is like function programming languages (python, perl, etc) |
| 3 | Pig parameter | Cloudera Manager will not support to update pig properties which is highly unlikely. So go to  >cd /etc/pig/conf/pig.properites  >cd /var/run/cloudera-scm-agent/process |
| 4 | Pig log file | # Log files will be created in the location where ever we launch Command Line Interface –CLI (grunt) and run commands.  So it is very important to decide from where you need to run the pig command |
| 5 | **Pig Admin commands** | 1. Output location strict check 2. Disabling Pig commands and operators 3. Blacklisting 4. Whitelisting 5. Note |
| 5.1 | Output location strict check | Pig scripts could contain multiple STORE statements. There are cases when one would like to avoid writing to the same output location. Pig provides admins/script writers with a property to check if multiple STORE statements make an attempt to write to the same output directory. And fail fast letting the user know of the same.  Specifically this makes sense for file-based output locations (HDFS, Local FS, S3..) to avoid Pig script from failing when multiple MR jobs write to the same location.  To enforce strict checking of output location, set **pig.location.check.strict=true** |
| 5.2 | Disabling Pig commands and operators | This is an admin feature providing ability to blacklist or/and whitelist certain commands and operations. Pig exposes a few of these that could be not very safe in a multitenant environment. For example, "sh" invokes shell commands, "set" allows users to change non-final configs. While these are tremendously useful in general, having an ability to disable would make Pig a safer platform. The goal is to allow administrators to be able to have more control over user scripts. Default behaviour would still be the same - no filters applied on commands and operators.  There are two properties you can use to control what users are able to do   * pig.blacklist * pig.whitelist |
| 5.3 | Blacklisting | Set "pig.blacklist" to a comma-delimited set of operators and commands. For eg, **pig.blacklist=rm,kill,cross** would disable users from executing any of "rm", "kill" commands and "cross" operator. |
| 5.4 | Whitelisting | This is an even safer approach to disallowing functionality in Pig. Using this you will be able to disable all commands and operators that are not a part of the whitelist. For eg, **pig.whitelist=load,filter,store** will disallow every command and operator other than "load", "filter" and "store". |
| 5.5 | Note | There should not be any conflicts between blacklist and whitelist. Make sure to have them entirely distinct or Pig will complain |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  | **Commands** | **After Pig Installation**   1. Cloudera Manager will not show Pig in service list . Because there is No Daemon process for pig in CM 2. It is just like a client based tool which will be deployed in slaves at the time of installation 3. So no need to ‘start’ or ‘stop’ the pig service |
| 1 | Check Pig availability | # To check pig is installed. It will enter into ‘grunt’ shell  >pig  Grunt> |
| 2 | Pig jar availability | # It is not mandatory….. Just to find the location where pig software is installed  >sudo find / -name “\*pig\*.jar” |
| 3 | Pig configuration path | # The default pig configuration path  >cd /etc/pig/config |
| 4 | Pig properties | ## There is no pig-site.xml, it uses properties like sqoop  /etc/pig/config/Pig.properties |
| 5 | HCatalog | # To be verified  >pig –useHCatalog |
| 6 | Ls , cat commands | # To list all the directory. No need to prefix ls with haoop or hdfs. Chk help  Grant> ls /user/hduser  Grant> fs cat /user/hduser/filename.txt -- cat to view the file |
| 7 | Help | # To get help  >pig –help |
| 8 | Help Properties | # To get help on properties  >pig -help properties   1. verbose=true|false; default is false. This property is the same as -v switch 2. brief=true|false; default is false. This property is the same as -b switch 3. debug=OFF|ERROR|WARN|INFO|DEBUG; default is INFO. This property is the same as -d switch 4. aggregate.warning=true|false; default is true. If true, prints count of warnings |
| 9 | Sample pig program | # Just for reference, not required for admin   1. grunt>cards\_rows = LOAD ‘user/hive/warehouse/cards.db/single\_deck/single\_deck.txt’; 2. grunt>cards\_group=GROUP cards\_rows ALL; 3. grunt>cards\_count = FOREACH cards\_group GENERATE COUNT\_STAR(cards\_rows) as ct 4. # All the above 3 commands are assigning values to variable, so it will not execute. But use the below command to execute … i 5. Grunt>illustrate cards\_count; 6. Grunt>dump cards\_count; 7. Grunt>describe cards\_count; |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |