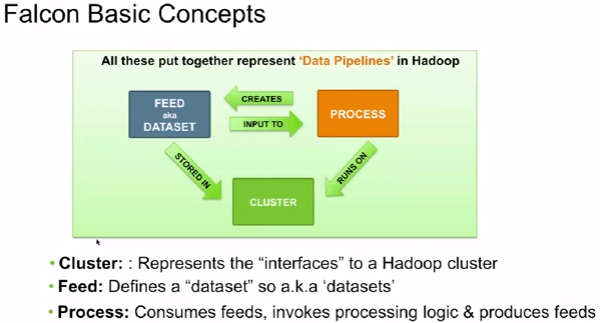
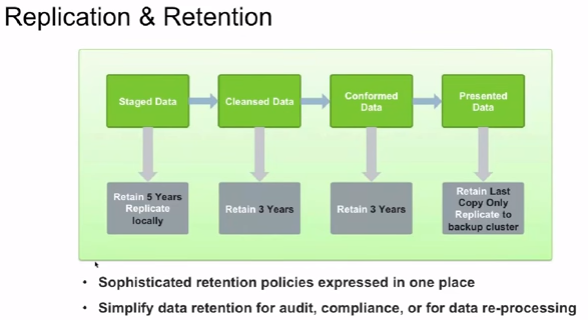
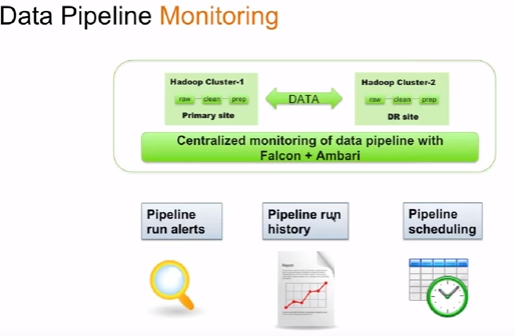
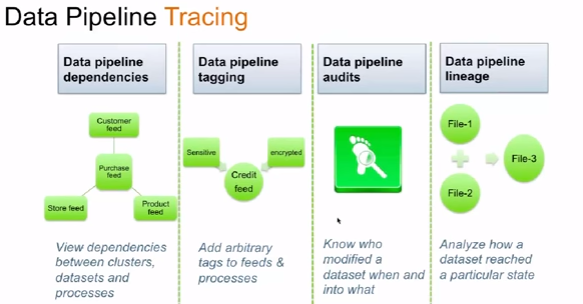
**Quick Ref: Falcon**

****

****



****

|  |  |  |
| --- | --- | --- |
| **S.No** | **Topic** | **Desc** |
|  | **Web Ref** |  |
|  | About Falcon | <https://www.youtube.com/watch?v=yLFZb41jsWM> |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  | **General Info** | **Only Ambari (Hortonworks)** |
| 1 | **About Falcon** | 1. Mainly used for Data governance 2. Using Falcon, you can start a Oozie workflow and tie a hive query, MapReduce job, pig script, even kick of another Oozie workflow, etc |
| 1.1 | About Falcon | **Apache Falcon** is a data governance engine that defines, schedules, and monitors data management policies. **Falcon** allows Hadoop administrators to centrally define their data pipelines, and then **Falcon**uses those definitions to auto-generate workflows in**Apache** Oozie. |
| 2 | **About Pipeline** | In computing, a **pipeline** is a set of **data** processing elements connected in series, where the output of one element is the input of the next one. The elements of a**pipeline** are often executed in parallel or in time-sliced fashion; in that case, some amount of buffer storage is often inserted between elements |
| 2.1 | Data Pipe line definition  *<XML based pipleline specification>* | 1. Modular – clusters, feeds & process defined separately and then linked together. 2. Easy to re-use across multiple pipelines |
| 2.2 | Data Pipe line definition  *<Out of box policies>* | # Predefined policies: For replication ,retention and late data handling easily customization of policies (using DAG)   1. **Replication:** you can set replication policy across the different steps of your pipe line 2. **Retention:** How long you want to retain data in different steps of your pipe line 3. **Late data handling:** ? |
| 2.3 | Data Pipe line definition  *<Extensible>* | 1. Plug in external solutions at any step of the pipleline. Ex: Invoke third party data obfuscation components 2. Using Falcan, you can start a Oozie workflow and tie a hive query, MapReduce job, pig script, even kick of another Oozie workflow, etc |
| 3 | **Pipe line monitor** | # Pipe line Monitor has the following steps   1. **Pipeline run alerts**: 2. **Pipeline run history**: 3. **Pipeline scheduling**: You can schedule some pipeline should run only peak or non-peak hours |
| 4 | **Pipeline Tracking** | 1. It is recommended to have pipeline tracing. So that you will have a visualization of your pipeline plan. 2. If you have visualization, it is easy to understand the dependency, so that you can update, add anything in the middle without impacting dependencies |
| 4.1 | Pipeline Tracking - steps | 1. Data pipeline dependencies 2. Data pipeline tagging 3. Data pipeline audits 4. Data pipeline lineage |
| 4.2 | Data pipeline dependencies | # View dependencies between cluster, datasets and process  # Your pipeline may deal with different set of data sets, different type of process. Pipeline tracking will help to understand those dependencies. It helps for impact analysis during changes |
| 4.3 | Data pipeline tagging | # Add arbitrary tags to feeds & process.  # Tag will help to understand certain things easily. Ex: Tag ownership of business data with description. |
| 4.4 | Data pipeline audits | # Know who modified a dataset, when it was modified and what is the change |
| 4.5 |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |