**Instalation Guide for Capturean**

In order to have a fully complete installation of Capture four software modules/requirements should be installed:

* *Infrastructure Module*:
  + Hbase: hbase-1.1.0.1/
  + Solr: solr-5.2.0/
    - Copy the core schemas from:
      * <https://gitlab.atosresearch.eu/ari/capture/tree/master/CaptureREST/install/cores-solr-5/capture-dc-fast>
      * https://gitlab.atosresearch.eu/ari/capture/tree/master/CaptureREST/install/cores-solr-5/capture-tweets-fast
  + Kafka. We are now running with kafka\_2.10-0.10.0.0/
    - Once kafka is installed and running, created following topics:
      * capture0-search0
      * capture0-search1
      * capture0-search2
      * capture0-stream0
      * capture0-stream1
      * capture0-stream2
  + Jetty: In teide we have two different jettys. One for CaptureREST and other one for capture-gui and AnalyticREST. The goal was to avoid having traces from each war in the same log. If just one jetty will be installed take into account jettyHome=jettyHome2
  + Git: To download the code from repository
    - /home/kuser/sources-new-capture
* *Core Module*: To install the core components do folling step
  + CaptureREST
    - Compilation and deployment
      * cd captureGitHOME/capture/capture-master
      * mvn -U clean install
      * cd captureGitHOME/capture/CaptureREST
      * mvn -U clean install -P production-iter (in the case of a standalone installation of Capturean the profile will be devel-localhost)
      * cp target/CaptureREST.war jettyHome1/webapps
  + AnalyticREST
    - Configuration:
      * In the file captureGitHOME/capture/AnalyticREST/src/main/resources/fingerkey.txt : add lines for each machine ssh-rsa where aggregate process will be run.
        + Log in, run the command “ssh-keyscan localhost” and copy all the content in fingerkey.txt
      * For example:
        + *hostname,capture03, ssh-rsa \*\*\*\*\*\*\*\*\*\*\*\**
        + *hostname,otherHostOrIP, ssh-rsa \*\*\*\*\*\*\*\*\*\*\*\**
      * In the file captureGitHOME/capture/AnalyticREST/src/main/resources/restfulservices.properties : update with endpoint data and connection data where aggregate process will be run.
      * For example:
        + *#endpoints*
        + *services.endpoint=http://ip:port/AnalyticREST/rest/*
        + *services.username=\*\*\*\**
        + *services.password=\*\*\*\**
        + *#host config*
        + *connection.host=capture03*
        + *connection.user=\*\*\*\**
        + *connection.password=\*\*\*\**
        + *connection.dirAgg=/data/batch-aggregates/*
        + *connection.dirBrand=/data/batch-brand/*
    - Compilation and deployment:
      * cd captureGitHOME/capture/AnalyticREST
      * mvn -U clean install -P production-iter (in the case of a standalone installation of Capturean the profile will be devel-localhost)
      * cp target/AnalyticREST.war jettyHome2/webapps
  + capture-gui:
    - Configuration
      * Change in file: captureGitHOME/capture/capture-gui/src/main/resources/restfulservices.properties services.endpoint and services.endpointCapture to the proper IP and PORT.
    - Compilation and deployment
      * cd captureGitHOME/capture/capture/capture-gui
      * mvn -U clean install -P production-iter (in the case of a standalone installation of Capturean the profile will be devel-localhost)
      * cp target/capture-gui.war jettyHome2/webapps
    - Create the h2 database for first time
      * Once the capture-gui.war is deployed, execute in browser following url:
        + <http://localhost:8080/capture-gui/createmodel> (change localhost by the public ip and the 8080 port by the port where the jettyHome2 will be listening)
      * Access via web to the db
        + <http://localhost:8080/capture-gui/console/>
        + Jdbc URL: jdbc:h2:~/capture-db
        + username: capture
        + password:capture
* *Pipeline streaming Module*:
  + In the directory where the pipeline will be deployed download: https://gitlab.atosresearch.eu/ari/capture/blob/master/CaptureREST/doc/streaming-pipeline.tar.gz
  + tar -zxvf streaming-pipeline.tar.gz
  + For each process
    - streaming-sentiment
      * Configuration
        + Adjust the metadata.broker.list host (not port if kafka run in the default port), kafka.zooKeeperEndpoint host (not port if zoo run in the default port) to the machine dns in the streamingPipelineHOME/streaming-pipeline/streaming-sentiment/kafka.properties file
      * Compilation and deployment:
        + cd captureGitHOME/capture/capture-stream-analysis
        + mvn -U clean install -P production-iter (in the case of a standalone installation of Capturean the profile will be devel-localhost)
        + cp target/capture-stream-analysis-0.0.1-SNAPSHOT-jar-with-dependencies.jar streamingPipelineHOME/streaming-pipeline/streaming-sentiment/
    - streaming-dpfilter
      * Configuration
        + Adjust the metadata.broker.list host (not port if kafka run in the default port), kafka.zooKeeperEndpoint host (not port if zoo run in the default port) to the machine dns in the streamingPipelineHOME/streaming-pipeline/streaming-dpfilter/kafka.properties file
      * Compilation and deployment:
        + cd captureGitHOME/capture/capture-stream-analysis
        + mvn -U clean install -P production-iter (in the case of a standalone installation of Capturean the profile will be devel-localhost)
        + cp target/capture-stream-analysis-0.0.1-SNAPSHOT-jar-with-dependencies.jar streamingPipelineHOME/streaming-pipeline/ streaming-dpfilter/
    - streaming-dpfilterV2/V3/V4 (actual V4)
      * Configuration
        + Adjust the metadata.broker.list host (not port if kafka run in the default port), kafka.zooKeeperEndpoint host (not port if zoo run in the default port) to the machine dns in the streamingPipelineHOME/streaming-pipeline/streaming-dpfilterV4/kafka.properties file
        + Adjust the **Flink CONSTANTS** in the streamingPipelineHOME/streaming-pipeline/streaming-dpfilterV4/kafka.properties file:

flink.webserver: true/false. Start flink webserver

flink.taskManagerSlots: number

dpFilter.checkEvery: number. Check every X milliseconds if datapools has been updated

* + - * Compilation and deployment:
        + cd captureGitHOME/capture/capture-stream-analysis
        + mvn -U clean install -P production-iter (in the case of a standalone installation of Capturean the profile will be devel-localhost)
        + cp target/capture-stream-analysis-0.0.1-SNAPSHOT-jar-with-dependencies.jar streamingPipelineHOME/streaming-pipeline/ streaming-dpfilterV4/
    - streaming-volume/
      * Configuration
        + Adjust the metadata.broker.list host (not port if kafka run in the default port), kafka.zooKeeperEndpoint host (not port if zoo run in the default port) to the machine dns in the streamingPipelineHOME/streaming-pipeline/streaming-volume/kafka.properties file
      * Compilation and deployment:
        + cd captureGitHOME/capture/capture-stream-aggregation
        + mvn -U clean install -P production-iter (in the case of a standalone installation of Capturean the profile will be devel-localhost)
        + cp target/capture-stream-aggregation-0.0.1-SNAPSHOT-jar-with-dependencies.jar streamingPipelineHOME/streaming-pipeline/ streaming-volume/
    - streaming-tagcloud/
      * Configuration
        + Adjust the metadata.broker.list host (not port if kafka run in the default port), kafka.zooKeeperEndpoint host (not port if zoo run in the default port) to the machine dns in the streamingPipelineHOME/streaming-pipeline/streaming-tagcloud /kafka.properties file
      * Compilation and deployment:
        + cd captureGitHOME/capture/capture-stream-aggregation
        + mvn -U clean install -P production-iter (in the case of a standalone installation of Capturean the profile will be devel-localhost)
        + cp target/capture-stream-aggregation-0.0.1-SNAPSHOT-jar-with-dependencies.jar streamingPipelineHOME/streamingPipelineHome/streaming-pipeline/streaming-tagcloud/
    - streaming-persistence
      * Configuration
        + Adjust the metadata.broker.list host (not port if kafka run in the default port), kafka.zooKeeperEndpoint host (not port if zoo run in the default port) to the machine dns in the streamingPipelineHOME/streaming-pipeline/streaming-persistence/kafka.properties file
      * Compilation and deployment:
        + cd captureGitHOME/capture/flink-persistence
        + mvn -U clean install -P production-iter (in the case of a standalone installation of Capturean the profile will be devel-localhost)
        + cp target/flink-persistence-0.0.1-SNAPSHOT-jar-with-dependencies.jar streamingPipelineHOME/streaming-pipeline/ streaming-persistence/
  + In order to launch each process, go to the pertinent folder in streamingPipelineHOME/streaming-pipeline/ and “./run.sh”
* *Pipeline search Module* 
  + In the directory where the pipeline will be deployed download: https://gitlab.atosresearch.eu/ari/capture/blob/master/CaptureREST/doc/search-pipeline.tar.gz
  + tar -zxvf search-pipeline.tar.gz
  + For each process
    - search-sentiment
      * Configuration
        + Adjust the metadata.broker.list host (not port if kafka run in the default port), kafka.zooKeeperEndpoint host (not port if zoo run in the default port) to the machine dns in the searchPipelineHOME/search-pipeline/search-sentiment/kafka.properties file
      * Compilation and deployment:
        + cd captureGitHOME/capture/capture-stream-analysis
        + mvn -U clean install -P production-iter (in the case of a standalone installation of Capturean the profile will be devel-localhost)
        + cp target/capture-stream-analysis-0.0.1-SNAPSHOT-jar-with-dependencies.jar searchPipelineHOME/search-pipeline/search-sentiment/
    - search-dpfilter
      * Configuration
        + Adjust the metadata.broker.list host (not port if kafka run in the default port), kafka.zooKeeperEndpoint host (not port if zoo run in the default port) to the machine dns in the searchPipelineHOME/search-pipeline/search-dpfilter/kafka.properties file
      * Compilation and deployment:
        + cd captureGitHOME/capture/capture-stream-analysis
        + mvn -U clean install -P production-iter (in the case of a standalone installation of Capturean the profile will be devel-localhost)
        + cp target/capture-stream-analysis-0.0.1-SNAPSHOT-jar-with-dependencies.jar searchPipelineHOME/search-pipeline/search-dpfilter/
    - search- dpfilterV2/V3/V4 (V4 actual)
      * Configuration
        + Adjust the metadata.broker.list host (not port if kafka run in the default port), kafka.zooKeeperEndpoint host (not port if zoo run in the default port) to the machine dns in the searchPipelineHOME/search-pipeline/search-dpfilterV4/kafka.properties file
        + Adjust the **Flink CONSTANTS** in the streamingPipelineHOME/streaming-pipeline/streaming-dpfilterV4/kafka.properties file:

flink.webserver: true/false. Start flink webserver

flink.taskManagerSlots: number

dpFilter.checkEvery: number. Check every X milliseconds if datapools has been updated

* + - * Compilation and deployment:
        + cd captureGitHOME/capture/capture-stream-analysis
        + mvn -U clean install -P production-iter (in the case of a standalone installation of Capturean the profile will be devel-localhost)
        + cp target/capture-stream-analysis-0.0.1-SNAPSHOT-jar-with-dependencies.jar searchPipelineHOME/search-pipeline/search-dpfilterV4/
    - search-persistence
      * Configuration
        + Adjust the metadata.broker.list host (not port if kafka run in the default port), kafka.zooKeeperEndpoint host (not port if zoo run in the default port) to the machine dns in the searchPipelineHOME/search-pipeline/search-persistence/kafka.properties file
      * Compilation and deployment:
        + cd captureGitHOME/capture/flink-persistence
        + mvn -U clean install -P production-iter (in the case of a standalone installation of Capturean the profile will be devel-localhost)
        + cp target/flink-persistence-0.0.1-SNAPSHOT-jar-with-dependencies.jar searchPipelineHOME/search-pipeline/search-persistence/
  + In order to launch each process, go to the pertinent folder in searchPipelineHOME/search-pipeline/ and “./run.sh”.

**Update of some core or (stream/search) pipeline process**

Once a process have been already installed and deployed one time, re-deploy following times not needed to perform the entire process. Just go to the “Compilation and deployment” bullet of the desired process to update.

**Paths**

captureGitHOME  path where git download capture.

jettyHome  path to Jetty home dir

jettyHome2  In teide we have two different jettys. One for CaptureREST and other one for capture-gui and AnalyticREST. This was like that to avoid having traces from each war in the same log.txt

streamingPipelineHOME  the directory where the pipeline will be deployed download. But if there are several nodes, each of them (each of the folder once installed) can be distributed (place) in distinct nodes. In teide we have all of them in the same frontend node

searchPipelineHOME  the directory where the seach pipeline will be installed. In a standalond distriburion all will be together. But if there are several nodes, each of them (each of the folder once installed) can be distributed (place) in distinct nodes. In teide, we have each of them in several node: Sentiment in capture02, dpfilter in capture01 and persistence in capture03