pyspark-plaso

Tool for Distributed Extraction of Timestamped Events from Files

Installation Guide

Marek Rychlý and Radek Burget

TARZAN Project Documentation
Faculty of Information Technology, Brno University of Technology

Last changed: December 28, 2019

PySpark Plaso Installation Guide

The PySpark Plaso is

- utilizing extractors adapted from the Plaso Project
- running at Docker containers from the TARZAN Docker Infrastructure Project

The Plaso extractors were adopted to an Apache Spark infrastructure and the extraction process is controlled by a Web service via a REST API. The following instructions describe how to build all required components, build the software artefacts for the Web service and run the service in the Spark infrastructure in Docker containers.

Building

```
cd ./deployment
# create a Python virtual environment including a required Python packages
./010-make-python-virtualenv.sh
# pack the Python packages into a ZIP file ready to use in PySpark
./020-make-site-packages-zip.sh
# create JAR packages for Java dependencies
./030-make-java-helpers.sh
```

The build process consists of several steps:

1. **to create a Python virtual environment including a required Python packages** with required dependencies:

   - Python 2.7 and above
   - Python Package Index (PyPI)
   - Virtual Python Environment builder (virtualenv)

   All required Python packages, that can be found in `misc/dependencies.txt`, will be installed by the script.

2. **to pack the Python packages into a ZIP file ready to use in PySpark** with required dependencies:

   - zip or p7zip

   Binary platform-dependent builds (`*.so` files) are not included in the ZIP file, so it can be distributed to nodes of various platforms. To use the binary parts of Python packages, install them by a package manager in each of the target nodes (e.g., by `INSTALL_PKGS` environment variable of the Spark Docker image.

3. **to create JAR packages for Java dependencies** which require:

    - Java JDK version 8 or above
    - Gradle Build Tool

    The Java dependencies are implemented by PySpark Plaso Java Helpers which is included as a sub-module in the PySpark Plaso repository.

After the build process, there should be the following files in `deployment/build`:

- `site-packages.zip` -- Python-based components
- `pyspark-java-helpers.jar` and `timeline-analyzer-spark.jar` -- Java-base components

Deployment

```
cd ./deployment
# run the PySpark Plaso infrastructure as Docker containers by docker-compose
./040-run-docker-webapp.sh
# reinitialise the PySpark Plaso Docker container with new build artefacts
./050-redeploy-docker-webapp.sh
```

To deploy the PySpark Plaso, the script creates and runs an infrastructure of Docker containers as defined in `deployment/docker-compose/webapp.yml`. The PySpark Plaso application will be deployed into Docker Spark Application container by copying the build artefacts into its `/app/lib` directory and the initialisation Python script `deployment/src/webapp_main.py` into the container's `/app` directory. See the docker-compose file.

In the case of modification of the build artefacts (a re-build) the Docker Spark Application container can be reinitialised by the corresponding script.