Configure User-Specific Props for Cluster Jobs

Contents:

For individual users of the Trifacta platform, an administrator may like to submit a custom set of properties to the Hadoop cluster for each job execution. For example, you may wish to change the available Java heap size for users who submit large jobs. This section describes how to define and deploy user-specific Java properties files for cluster jobs.

> (i) **NOTE:** User-specific custom properties override or append any other custom properties that are passed to Hadoop. Suppose the Java properties file contains a single property.
>
>   - If the property is not specified elsewhere in the job definition, it is appended to any other properties that are passed.
>   - If the property is specified elsewhere in the job definition, the Java properties file overrides the other custom property value.

> (i) **NOTE:** You cannot specify user-specific properties for S3 jobs.

Enable Java properties directory

This feature is enabled by defining the values for the Java properties directory for Spark in the platform configuration.

You can apply this change through the *Admin Settings Page* (recommended) or

`trifacta-conf.json`

. For more information, see *Platform Configuration Methods*.

Steps:

1. Set the following properties to the directories where the user-specific property files can be stored on the Trifacta node. Example:

```
"spark.userPropsDirectory": "/opt/trifacta/conf/usr",
```

2. Save your changes and restart the platform.

Required Permissions

For the above locations, the Trifacta service requires the following permissions:

- Execute permission on the directory defined in `spark.userPropsDirectory`.

- Read permission on the files in them.

Define user-specific properties files

For each user that is passing in custom properties, a separate file must be created in the appropriate directory with the following filename pattern:

```
userEmail-user.properties
```

where:

`userEmail` is the email address for the user that is registered with the Trifacta platform. For example, for userId `joe@example.com`, the filename is `joe@example.com-user.properties`.

File Format:

Each file must follow Java Properties file format, which is the following:

```
property.a=value.a
property.b=value.b
```

> ⓘ **NOTE:** Do not include `spark.props` in your property names. For example, if you are modifying the Spark YARN queue (`spark.props.spark.yarn.queue`) for the user, the property name must be `spark.yarn.queue`.

For more information on this format, see
https://docs.oracle.com/cd/E23095_01/Platform.93/ATGProgGuide/html/s0204propertiesfileformat01.html.