TRIFACTA

Install Guide for Azure Marketplace

Version: 6.8.1
Doc Build Date: 02/21/2020

Copyright © Trifacta Inc. 2020 - All Rights Reserved. CONFIDENTIAL

These materials (the “Documentation”) are the confidential and proprietary information of Trifacta Inc. and may not be reproduced, modified, or distributed without the prior written permission of Trifacta Inc.

EXCEPT AS OTHERWISE PROVIDED IN AN EXPRESS WRITTEN AGREEMENT, TRIFACTA INC. PROVIDES THIS DOCUMENTATION AS-IS AND WITHOUT WARRANTY AND TRIFACTA INC. DISCLAIMS ALL EXPRESS AND IMPLIED WARRANTIES TO THE EXTENT PERMITTED, INCLUDING WITHOUT LIMITATION THE IMPLIED WARRANTIES OF MERCHANTABILITY, NON-INFRINGEMENT AND FITNESS FOR A PARTICULAR PURPOSE AND UNDER NO CIRCUMSTANCES WILL TRIFACTA INC. BE LIABLE FOR ANY AMOUNT GREATER THAN ONE HUNDRED DOLLARS ($100) BASED ON ANY USE OF THE DOCUMENTATION.

For third-party license information, please select **About Trifacta** from the Help menu.

Release Notes

This section contains release notes for published versions of Trifacta® Wrangler Enterprise.

Release Notes 6.8

Contents:

Release 6.8.1

February 7, 2020

This release enables some new features and makes some relational connections generally available.

What's New

Install:

- Support for CDH 6.3. See *Supported Deployment Scenarios for Cloudera*.

> **NOTE:** Support for CDH 6.0 has been deprecated. See *End of Life and Deprecated Features*.

Import:

- Upload tabular data from PDF documents.

> **NOTE:** This feature is in Beta release.

> **NOTE:** This feature must be enabled.

  See *Import PDF Data*.
- Read support for ORC tables managed through Hive. See *Configure for Hive*.

LDAP:

- Support for initial binding to active directory using the user's account. See *Configure SSO for AD-LDAP*.

Copyright © 2020 Trifacta Inc.

Cluster Clean:

- Cluster Clean standardization feature is now available in all product editions. See *Overview of Cluster Clean*.

Documentation:

- API: Improved documentation for the asset transfer endpoint. See *Changes to the APIs*.

Changes to System Behavior

Wrangler Enterprise desktop application:

> **NOTE:** In a future release, the Wrangler Enterprise desktop application will be deprecated. Please switch to a supported version of Google Chrome or Mozilla Firefox. Support for Edge Chromium is expected in a future release. See *Desktop Requirements*.

CLI and v3 endpoints (Release 6.4):

> **NOTE:** Do not attempt to connect to the Trifacta platform using any version of the CLI or the v3 endpoints. They are no longer supported and unlikely to work.

In Release 6.4:

- The Command Line Interface (CLI) was deprecated. Customers must use the v4 API endpoints instead.
- The v3 versions of the API endpoints were deprecated. Customers must use the v4 API endpoints instead.
- Developer content was provided to assist in migrating to the v4 API endpoints.
- For more information on acquiring this content, please contact *Trifacta Support*.

General availability:

- The following relational connections are now generally available:
  - DB2 (import only)
  - Salesforce (import only)
  - Tableau Server (publish only)
    For more information, see *Connection Types*.

Key Bug Fixes

Ticket	Description
TD-45492	Publishing to Databricks Tables fails on ADLS Gen1 in user mode.

New Known Issues

Ticket	Description
TD-47263	Importing an exported flow that references a Google Sheets or Excel source breaks connection to input source. **Workaround:** If the importing user has access to the source, the user can re-import the dataset and then swap the source for the broken recipe.

Copyright © 2020 Trifacta Inc.

Release 6.8

December 6, 2019

Welcome to Release 6.8 of Trifacta® Wrangler Enterprise. This release introduces several key features around operationalizing the platform across the enterprise. Enterprise stakeholders can now receive email notifications when recurring jobs have succeeded or failed, updating data consumers outside of the platform. This release also introduces a generalized webhook interface, which facilitates push notifications to applications such as Slack when jobs have completed. When jobs fail, users can download a much richer support bundle containing configuration files, script files, and a specified set of log files.

Macros have been expanded to now be export- and import-ready across environments. In support of this feature, the Wrangle Exchange is now available through the Trifacta Community, where you can download macros created by others and import them for your own use. Like macros, you can now export and import flows across product editions and release (Release 6.8 or later only).

In the application, you can now use shortcut keys to navigate around the workspace and the Transformer page. And support for the Firefox browser has arrived. Read on for more goodness added with this release.

What's New

Install:

- Support for ADLS Gen2 blob storage. See *Enable ADLS Gen2 Access*.

Workspace:

- Individual users can now enable or disable keyboard shortcuts in the workspace or Transformer page. See *User Profile Page*.
- Configure locale settings at the workspace or user level. See *Locale Settings*.
- You can optionally duplicate the datasets from a source flow when you create a copy of it. See *Flow View Page*.
- Create a copy of your imported dataset. See *Library Page*.

Browser:

- Support for Firefox browser.

  > **NOTE:** This feature is in Beta release.

  For supported versions, see *Desktop Requirements*.

Project Management:

- Support for export and import of macros. See *Macros Page*.
  - For more information on macros, see *Overview of Macros*.
- Download and use macros available through the Wrangle Exchange. See *https://www.trifacta.com/blog/crowdsourcing-macros-trifacta-wrangle-exchange/*.

Operationalization:

- Create webhook notifications for third-party platforms based on results of your job executions. See *Create Flow Webhook Task*.
- Enable and configure email notifications based on the success or failure of job executions.

Copyright © 2020 Trifacta Inc.

> **NOTE:** This feature requires access to an SMTP server. See *Enable SMTP Email Server Integration*.

- For more information on enabling, see *Workspace Admin Page*.
  - Individual users can opt out of receiving email messages or can configure use of a different email address. See *Email Notifications Page*.
- For more information on enabling emails for individual flows, see *Manage Flow Notifications Dialog*.

Supportability:

- Download logs bundle on job success or failure now contains extensive configuration information to assist in debugging. For more information, see *Configure Support Bundling*.

Connectivity:

- Support for integration with EMR 5.8 - 5.27. For more information, see *Configure for EMR*.
- Connect to SFTP servers to read data and write datasets. See *Create SFTP Connections*.
- Create connections to Databricks Tables.

> **NOTE:** This connection is supported only when the Trifacta platform is connected to an Azure Databricks cluster.

For more information, see *Create Databricks Tables Connections*.

- Support for using non-default database for your Snowflake stage.
  - Support for ingest from read-only Snowflake databases.
  - See *Enable Snowflake Connections*.

Import:

- As of Release 6.8, you can import an exported flow into any edition or release after the build number of the export. See *Import Flow*.
- Improved monitoring of long-loading relational sources. See *Import Data Page*.

> **NOTE:** This feature must be enabled. See *Configure JDBC Ingestion*.

Transformer Page:

- Select columns, functions applied to your source, and constants to replace your current dataset. See *Select*.
- Improved Date/Time format selection. See *Choose Datetime Format Dialog*.

> **Tip:** Datetime formats in card suggestions now factor in the user's locale settings for greater relevance.

- Improved matching logic and performance when matching columns through RapidTarget.
  - Align column based on the data contained in them, in addition to column name.
  - This feature is enabled by default. For more information, see *Overview of RapidTarget*.
- Improvements to the Search panel enable faster discovery of transformations, functions, and other objects. See *Search Panel*.

Copyright © 2020 Trifacta Inc.

Job execution:

- By default, the Trifacta application permits up to four jobs from the same flow to be executed at the same time. If needed, you can configure the application to execute jobs from the same flow one at a time. See *Configure Application Limits*.
- If you enabled visual profiling for your job, you can download a JSON version of the visual profile. See *Job Details Page*.
- Support for instance pooling in Azure Databricks. See *Configure for Azure Databricks*.

Language:

- New trigonometry and statistical functions. See *Changes to the Language*.

API:

- Apply overrides at time of job execution via API.
- Define import mapping rules for your deployments that use relational sources or publish to relational targets.
- Export and import macro definitions.
- See *Changes to the APIs*.

Changes in System Behavior

Browser Support Policy:

- For supported browsers, at the time of release, the latest stable version and the two previous stable versions are supported.

  > **NOTE:** Stable browser versions released after a given release of Trifacta Wrangler Enterprise will **NOT** be supported for any prior version of Trifacta Wrangler Enterprise. A best effort will be made to support newer versions released during the support lifecycle of the release.

  For more information, see *Desktop Requirements*.

Install:

> **NOTE:** In the next release of Trifacta Wrangler Enterprise after Release 6.8, support for installation on CentOS/RHEL 6.x and Ubuntu 14.04 will be deprecated. You should upgrade the Trifacta node to a supported version of CentOS/RHEL 7.x or Ubuntu 16.04. Before performing the upgrade, please perform a full backup of the Trifacta platform and its databases. See *Backup and Recovery*.

- Support for Spark 2.1 has been deprecated. Please upgrade to a supported version of Spark.
  - Support for EMR 5.6 and eMR 5.7 has also been deprecated. Please upgrade to a supported version of EMR.
  - For more information, see *Product Support Matrix*.
- To simplify the installation distribution, the Hadoop dependencies for the recommended version only are included in the software download. For the dependencies for other supported Hadoop distributions, you must download them from the Trifacta FTP site and install them on the Trifacta node. See *Install Hadoop Dependencies*.
- Trifacta node has been upgraded to use Python 3. This instance of Python has no dependencies on any Python version external to the Trifacta node.

Copyright © 2020 Trifacta Inc.

Import/Export:

- Flows can now be exported and imported across products and versions of products. See *Changes to the Object Model*.

CLI and v3 endpoints (Release 6.4):

> **NOTE:** Do not attempt to connect to the Trifacta platform using any version of the CLI or the v3 endpoints. They are no longer supported and unlikely to work.

In Release 6.4:

- The Command Line Interface (CLI) was deprecated. Customers must use the v4 API endpoints instead.
- The v3 versions of the API endpoints were deprecated. Customers must use the v4 API endpoints instead.
- Developer content was provided to assist in migrating to the v4 API endpoints.
- For more information on acquiring this content, please contact *Trifacta Support*.

Key Bug Fixes

Ticket	Description
TD-40348	When loading a recipe in an imported flow that references an imported Excel dataset, Transformer page displays Input validation failed: (Cannot read property 'filter' of undefined) error, and the screen is blank.
TD-42080	Cannot run flow or deployment that contains more than 10 recipe jobs

New Known Issues

Ticket	Description
TD-46123	Cannot modify the type of relational target for publishing action. **Workaround:** Create a new publishing action with the desired relational target. Remove the original one if necessary. See *Run Job Page*.
TD-45923	Publishing a compressed Snappy file to SFTP fails.
TD-45922	You cannot publish TDE format to SFTP destinations.
TD-45492	Publishing to Databricks Tables fails on ADLS Gen1 in user mode.

Copyright © 2020 Trifacta Inc.

TD-45273	Artifact Storage Service fails to start on HDP 3.1. **Workaround:** The Artifact Storage Service can reference the HDP 2.6 Hadoop bundle JAR. 1. You can apply this change through the *Admin Settings Page* (recommended) or `trifacta-conf.json`. For more information, see *Platform Configuration Methods*. 2. Locate the following property: 3. Replace this value: 4. With the following: 5. Save changes and restart the platform.
TD-45122	API: re-running job using only the `wrangleDataset` identifier fails even if the original job succeeds when `writeSettings` were specified. **Workaround:** Use a full `jobGroups` job specification each time that you run a job. See *API JobGroups Create v4*.

TD-44429	Cannot publish outputs to relational targets, receiving `Encountered error while processing stream`. **Workaround:** This issue may be caused by the `trifacta` service account not having write and execute permissions to the `/tmp` directory on the Trifacta node. If so, you can do either of the following: 1. Enable write and execute permissions for the account on `/tmp`. 2. Create a new temporary account and provide the service account write and execute permissions to it. Then, add the following to `data-service.jvmOptions`:
TD-44427	Cannot publish dataset containing duplicate rows to Teradata. Error message: `Caused by: java.sql.SQLException: [Teradata Database] [TeraJDBC 15.10.00.14] [Error -2802] [SQLState 23000] Duplicate row error in abc_trifacta.tmp_218768523. at` **Workaround:** This is a known limitation on Teradata. For more information on this limitation, see *Enable Teradata Connections*.

Release Notes 6.4

Contents:

Copyright © 2020 Trifacta Inc.

Release 6.4.2

November 15, 2019

This release is primarily a bug fix release with the following new features.

What's New

API:

- Apply overrides at time of job execution via API.
- Define import mapping rules for your deployments that use relational sources or publish to relational targets.
- See *Changes to the APIs*.

Job execution:

- By default, the Trifacta application permits up to four jobs from the same flow to be executed at the same time. If needed, you can configure the application to execute jobs from the same flow one at a time. See *Configure Application Limits*.

Changes in System Behavior

None.

Key Bug Fixes

Ticket	Description
TD-44548	RANGE function returns null values if more than 1000 values in output.
TD-44494	Lists are not correctly updated in Deployment mode
TD-44311	Out of memory error when running a flow with many output objects
TD-44188	Performance is poor for SQL DW connection
TD-43877	Preview after a DATEFORMAT step does not agree with results or profile values
TD-44035	Spark job failure from Excel source
TD-43849	Export flows are broken when recipe includes Standardization or Transform by Example tasks. **NOTE:** This Advanced Feature is available in Trifacta Wrangler Enterprise under a separate, additional license. If it is not available under your current license, do not enable it for use. Please feel free to contact your representative.

Copyright © 2020 Trifacta Inc.

New Known Issues

None.

Release 6.4.1

August 30. 2019This release includes bug fixes and introduces SSO connections for Azure relational sources.

What's New

Connectivity:

- You can now leverage your Azure AD SSO infrastructure to create SSO connections to Azure relational databases. For more information, see *Enable SSO for Azure Relational Connections*.

Changes in System Behavior

Configuration changes:

- The parameter to enable custom SQL query has been moved to the Workspace Admin page.
- The parameter to disable schematized output has been moved to the Workspace Admin page.
- For more information, see *Changes to Configuration*.

Key Bug Fixes

Ticket	Description
TD-39086	Hive ingest job fails on Microsoft Azure.

New Known Issues

None.

Release 6.4

August 1, 2019

This release of Trifacta® Wrangler Enterprise features broad improvements to the recipe development experience, including multi-step operations and improved copied and paste within the Recipe panel. As a result of the panel's redesign, you can now create user-defined macros, which are sets of sequenced and parameterized steps for easy reuse and adaptation for other recipes. When jobs are executed, detailed monitoring provides enhanced information on progress of the job through each phase of the process. You can also connect to a broader ecosystem of sources and targets, including enhancements to the integration with Tableau Server and AWS Glue. New for this release: read from your Snowflake sources. Read on for additional details on new features and enhancements.

What's New

Transformer Page:

- The redesigned Recipe panel enables multi-step operations and more robust copy and paste actions. See *Recipe Panel*.
- Introducing user-defined macros, which enable saving and reusing sequences of steps. For more information, see *Overview of Macros*.
- Transform by example output values for a column of values. See *Transformation by Example Page*.
    - For an overview of this feature, see *Overview of TBE*.

Copyright © 2020 Trifacta Inc.

- Browse current flow for datasets or recipes to join into the current recipe. See *Join Panel*.
- Replace specific cell values. See *Replace Cell Values*.

Job Execution:

- Detailed job monitoring for ingest and publishing jobs. See *Overview of Job Monitoring*.
- Parameterize output paths and table and file names. See *Run Job Page*.

Install:

- Support for RHEL/CentOS 7.5 and 7.6 for the Trifacta node. See *System Requirements*.
- Support for deployment of Trifacta platform via Docker image. See *Install for Docker*.

Connectivity:

- Support for integration with Cloudera 6.2.x. See *System Requirements*.

  > **NOTE:** Support for integration with Cloudera 5.15.x and earlier has been deprecated. See *End of Life and Deprecated Features*.

  > **NOTE:** Support for integration with HDP 2.5.x and earlier has been deprecated. See *End of Life and Deprecated Features*.

- Support for Snowflake database connections.

  > **NOTE:** This feature is supported only when Trifacta Wrangler Enterprise is installed on customer-managed AWS infrastructure.

  For more information, see *Enable Snowflake Connections*.

- Support for direct publishing to Tableau Server. For more information, see *Run Job Page*.
- Support for MySQL database timezones. See *Install Databases for MySQL*.

Enhanced support for AWS Glue integration:

- Metadata catalog browsing through the application. See *AWS Glue Browser*.
- Per-user authentication to Glue. See *Configure AWS Per-User Authentication*.
- See *Enable AWS Glue Access*.

Import:

- Add timestamp parameters to your custom SQL statements to enable data import relative to the job execution time. See *Create Dataset with SQL*.

Authentication:

- Leverage your enterprise's SAML identity provider to pass through a set of IAM roles that Trifacta users can select for access to AWS resources.

  > **NOTE:** This authentication method is supported only if SSO authentication has been enabled using the platform-native SAML authentication method. For more information, see *Configure SSO for SAML*.

Copyright © 2020 Trifacta Inc.

For more information, see *Configure for AWS SAML Passthrough Authentication*.

- Support for AzureManaged Identities with Azure Databricks. See *Configure for Azure Databricks*.

Admin:

- Administrators can review, enable, disable, and delete schedules through the application. See *Schedules Page*.

Sharing:

- Share flows and connections with groups of users imported from your LDAP identity provider.

  NOTE: This feature is in Beta release.

  See *Configure Users and Groups*.

Logging:

- Tracing user information across services for logging purposes. See *Configure Logging for Services*.

Language:

- New functions. See *Changes to the Language*.
- Broader support for metadata references. For Excel files, `$filepath` references now return the location of the source Excel file. Sheet names are appended to the end of the reference. See *Source Metadata References*.

APIs:

- Admins can now generate password reset requests via API. See *Changes to the APIs*.

Databases:

- New databases:
  - Job Metadata Service database

Changes in System Behavior

NOTE: The Trifacta software must now be installed on an edge node of the cluster. Existing customers who cannot migrate to an edge node will be supported. You will be required to update cluster files on the Trifacta node whenever they change, and cluster upgrades may be more complicated. You should migrate your installation to an edge node if possible. For more information, see *System Requirements*.

NOTE: The v3 APIs are no longer supported. Please migrate immediately to using the v4 APIs.

NOTE: The command line interface (CLI) is no longer available. Please migrate immediately to using the v4 APIs.

Copyright © 2020 Trifacta Inc.

NOTE: The PNaCl browser client extension is no longer supported. Please verify that all users of Trifacta Wrangler Enterprise are using a supported version of Google Chrome, which automatically enables use of WebAssembly. For more information, see *Desktop Requirements*.

NOTE: Support for Java 7 has been deprecated in the platform. Please upgrade to Java 8 on the Trifacta node and any connected cluster. Some versions of Cloudera may install Java 7 by default.

NOTE: The **Chat with us** feature is no longer available. For Trifacta Wrangler Enterprise customers, this feature had to be enabled in the product. For more information, see *Trifacta Support*.

NOTE: The desktop version of Trifacta Wrangler will cease operations on August 31, 2019. If you are still using the product at that time, your data will be lost. Please transition to using the free Cloud version of Trifacta® Wrangler. Automated migration is not available. To register for a free account, please visit *https://cloud.trifacta.com*.

Workspace:

- Configuration for AWS authentication for platform users has been migrated to a new location. See *Configure Your Access to S3*.

API:

- The endpoint used to assign an AWSConfig object to a user has been replaced.

  NOTE: If you used the APIs to assign AWSConfig objects in a previous release, you must update your scripts to assign AWS configurations. For more information, see *Changes to the APIs*.

Documentation:

- In prior releases, the documentation listed UTF32-BE and UTF32-LE as supported file formats. These formats are not supported. Documentation has been updated to correct this error. See *Supported File Encoding Types*.

Key Bug Fixes

Ticket	Description
TD-41260	Unable to append Trifacta Decimal type into table with Hive Float type. See *Hive Data Type Conversions*.

Copyright © 2020 Trifacta Inc.

TD-40424	`UTF-32BE` and `UTF-32LE` are available as supported file encoding options. They do not work. **NOTE:** Although these options are available in the application, they have never been supported in the underlying platform. They have been removed from the interface.
TD-40299	Cloudera Navigator integration cannot locate the database name for JDBC sources on Hive.
TD-40243	API access tokens don't work with native SAML SSO authentication
TD-39513	Import of folder of Excel files as parameterized dataset only imports the first file, and sampling may fail.
TD-39455	HDI 3.6 is not compatible with Guava 26.
TD-39092	`$filepath` and `$sourcerownumber` references are not supported for Parquet file inputs. For more information, see *Source Metadata References*.
TD-31354	When creating Tableau Server connections, the Test Connection button is missing. See *Create Tableau Server Connections*.
TD-36145	Spark running environment recognizes numeric values preceded by `+` as Integer or Decimal data type. Photon running environment does not and types these values as strings.

New Known Issues

Ticket	Description
TD-42638	Publishing and ingest jobs that are short in duration cannot be canceled. **Workaround:** Allow the job to complete. You can track the progress through these phases of the jobs through the application. See *Job Details Page*.
TD-39052	Changes to signout on reverse proxy method of SSO do not take effect after upgrade.

Release Notes 6.0

Contents:

Copyright © 2020 Trifacta Inc.

Release 6.0.2

This release addresses several bug fixes.

What's New

- Support for Cloudera 6.2. For more information, see *System Requirements*.

Changes to System Behavior

> **NOTE:** As of Release 6.0, all new and existing customers must license, download, and install the latest version of the Tableau SDK onto the Trifacta node. For more information, see *Create Tableau Server Connections*.

Upload:

- In previous releases, files that were uploaded to the Trifacta platform that had an unsupported filename extension received a warning before upload.
- Beginning in this release, files with unsupported extensions are blocked from upload.
- You can change the list of supported file extensions. For more information, see *Miscellaneous Configuration*.

Documentation:

- In Release 6.0.x documentation, documentation for the API JobGroups Get Status v4 endpoint was mistakenly published. This endpoint does not exist. For more information on the v4 equivalent, see *Changes to the APIs*.

Key Bug Fixes

Ticket	Description
TD-40471	SAM auth: Logout functionality not working
TD-39318	Spark job fails with parameterized datasets sourced from Parquet files
TD-39213	Publishing to Hive table fails

New Known Issues

None.

Release 6.0.1

This release features support for several new Hadoop distributions and numerous bug fixes.

What's New

Connectivity:

Copyright © 2020 Trifacta Inc.

- Support for integration with CDH 5.16.
- Support for integration with CDH 6.1. Version-specific configuration is required.

  See *Supported Deployment Scenarios for Cloudera*.
- Support for integration with HDP 3.1. Version-specific configuration is required. See *Supported Deployment Scenarios for Hortonworks*.
  - Support for Hive 3.0 on HDP 3.0 or HDP 3.1. Version-specific configuration is required. See *Configure for Hive*.
- Support for Spark 2.4.0.

  > **NOTE:** There are some restrictions around which running environment distributions support and do not support Spark 2.4.0.

  For more information, see *Configure for Spark*.
- Support for integration with high availability for Hive.

  > **NOTE:** High availability for Hive is supported on HDP 2.6 and HDP 3.0 with Hive 2.x enabled. Other configurations are not currently supported.

  For more information, see *Create Hive Connections*.

Publishing:

- Support for automatic publishing of job metadata to Cloudera Navigator.

  > **NOTE:** For this release, Cloudera 5.16 only is supported.

  For more information, see *Configure Publishing to Cloudera Navigator*.

Changes to System Behavior

Key Bug Fixes

Ticket	Description
TD-39779	MySQL JARs must be downloaded by user. **NOTE:** If you are installing the databases in MySQL, you must download a set of JARs and install them on the Trif acta node. For more information, see *Install Databases for MySQL*.
TD-39694	Tricheck returns status code 200, but there is no response. It does not work through Admin Settings page.
TD-39455	HDI 3.6 is not compatible with Guava 26.
TD-39086	Hive ingest job fails on Microsoft Azure.

New Known Issues

Ticket	Description

Copyright © 2020 Trifacta Inc.

TD-40299	Cloudera Navigator integration cannot locate the database name for JDBC sources on Hive.
TD-40348	When loading a recipe in imported flow that references an imported Excel dataset, Transformer page displays Input validation failed: (Cannot read property 'filter' of undefined) error, and the screen is blank. **Workaround:** In Flow View, select an output object, and run a job. Then, load the recipe in the Transformer page and generate a new sample. For more information, see *Import Flow*.
TD-39969	On import, some Parquet files cannot be previewed and result in a blank screen in the Transformer page. **Workaround:** Parquet format supports row groups, which define the size of data chunks that can be ingested. If row group size is greater than 10 MB in a Parquet source, preview and initial sampling does not work. To workaround this issue, import the dataset and create a recipe for it. In the Transformer page, generate a new sample for it. For more information, see *Parquet Data Type Conversions*.

Release 6.0

This release of Trifacta® Wrangler Enterprise introduces key features around column management, including multi-select and copy and paste of columns and column values. A new Job Details page captures more detailed information about job execution and enables more detailed monitoring of in-progress jobs. Some relational connections now support publishing to connected databases. This is our largest release yet. Enjoy!

NOTE: This release also announces the deprecation of several features, versions, and supported extensions. Please be sure to review Changes to System Behavior below.

What's New

NOTE: Beginning in this release, the Wrangler Enterprise desktop application requires a 64-bit version of Microsoft Windows. For more information, see *Install Desktop Application*.

Wrangling:

- In data grid, you can select multiple columns before receiving suggestions and performing transformations on them. For more information, see *Data Grid Panel*.
    - New Selection Details panel enables selection of values and groups of values within a selected column. See *Selection Details Panel*.
- Copy and paste columns and column values through the column menus. see *Copy and Paste Columns*.
- Support for importing files in Parquet format. See *Supported File Formats*.
- Specify ranges of key values in your joins. See *Configure Range Join*.

Jobs:

Copyright © 2020 Trifacta Inc.

- Review details and monitor the status of in-progress jobs through the new Job Details page. See *Job Details Page*.
- Filter list of jobs by source of job execution or by date range. See *Jobs Page*.

Connectivity:

- Publishing (writeback) is now supported for relational connections.
  - This feature is enabled by default

    > **NOTE:** After a connection has been enabled for publishing, you cannot disable publishing for that connection. Before you enable, please verify that all user accounts accessing databases of these types have appropriate permissions.

    See *Enable Relational Connections*.
  - The following connection types are natively supported for publishing to relational systems.
    - *Oracle Data Type Conversions*
    - *Postgres Data Type Conversions*
    - *SQL Server Data Type Conversions*
    - *Teradata Data Type Conversions*
- Import folders of Microsoft Excel workbooks. See *Import Excel Data*.
- Support for integration with CDH 6.0. Version-specific configuration is required.See *Supported Deployment Scenarios for Cloudera*.
- Support for integration with HDP 3.0. Version-specific configuration is required. See *Supported Deployment Scenarios for Hortonworks*.
  - Support for Hive 3.0 on HDP 3.0 only. Version-specific configuration is required. See *Configure for Hive*.

Language:

- Track file-based lineage using `$filepath` and `$sourcerownumber` references. See *Source Metadata References*.
- In addition to directly imported files, the `$sourcerownumber` reference now works for converted files (such as Microsoft Excel workbooks) and for datasets with parameters. See *Source Metadata References*.

Workspace:

- Organize your flows into folders. See *Flows Page*.

Publishing:

- Users can be permitted to append to Hive tables when they do not have CREATE or DROP permissions on the schema.

  > **NOTE:** This feature must be enabled. See *Configure for Hive*.

Administration:

- New Workspace Admin page centralizes many of the most common admin settings. See Changes to System Behavior below.
- Download system logs through the Trifacta application. See *Admin Settings Page*.

Supportability:

- High availability for the Trifacta node is now generally available. See *Install for High Availability*.

Authentication:

Copyright © 2020 Trifacta Inc.

- Integrate SSO authentication with enterprise LDAP-AD using platform-native LDAP support.

  NOTE: This feature is in Beta release.

  NOTE: In previous releases, LDAP-AD SSO utilizes an Apache reverse proxy. While this method is still supported, it is likely to be deprecated in a future release. Please migrate to using the above SSO method. See *Configure SSO for AD-LDAP*.

- Support for SAML SSO authentication. See *Configure SSO for SAML*.

Changes to System Behavior

NOTE: The Trifacta node requires NodeJS 10.13.0. See *System Requirements*.

Configuration:

To simplify configuration of the most common feature enablement settings, some settings have been migrated to the new Workspace Admin page. For more information, see *Workspace Admin Page*.

NOTE: Over subsequent releases, more settings will be migrated to the Workspace Admin page from the Admin Settings page and from `trifacta-conf.json`. For more information, see *Changes to Configuration*.

See *Platform Configuration Methods*.

See *Admin Settings Page*.**Java 7:**

NOTE: In the next release of Trifacta Wrangler Enterprise, support for Java 7 will be end of life. The product will no longer be able to use Java 7 at all. Please upgrade to Java 8 on the Trifacta node and your Hadoop cluster.

Key Bug Fixes

Ticket	Description
TD-36332	Data grid can display wrong results if a sample is collected and dataset is unioned.
TD-36192	Canceling a step in recipe panel can result in column menus disappearing in the data grid.
TD-35916	Cannot logout via SSO
TD-35899	A deployment user can see all deployments in the instance.
TD-35780	Upgrade: Duplicate metadata in separate publications causes DB migration failure.
TD-35644	Extractpatterns with "HTTP Query strings" option doesn't work.
TD-35504	Cancel job throws 405 status code error. Clicking Yes repeatedly pops up Cancel Job dialog.
TD-35486	Spark jobs fail on LCM function that uses negative numbers as inputs.

Copyright © 2020 Trifacta Inc.

TD-35483	Differences in how WEEKNUM function is calculated in the Trifacta Photon and Spark running environments, due to the underlying frameworks on which the environments are created. **NOTE:** Trifacta Photon and Spark jobs now behave consistently. Week 1 of the year is the week that contains January 1.
TD-35481	Upgrade Script is malformed due to SplitRows not having a Load parent transform.
TD-35177	Login screen pops up repeatedly when access permission is denied for a connection.
TD-27933	For multi-file imports lacking a newline in the final record of a file, this final record may be merged with the first one in the next file and then dropped in the Trifacta Photon running environment.

New Known Issues

Ticket	Description
TD-39513	Import of folder of Excel files as parameterized dataset only imports the first file, and sampling may fail. **Workaround:** Import as separate datasets and union together.
TD-39455	HDI 3.6 is not compatible with Guava 26.
TD-39092	`$filepath` and `$sourcerownumber` references are not supported for Parquet file inputs. **Workaround:** Upload your Parquet files. Create an empty recipe and run a job to generate an output in a different file format, such as CSV or JSON. Use that output as a new dataset. See *Build Sequence of Datasets*. For more information on these references, see *Source Metadata References*.
TD-39086	Hive ingest job fails on Microsoft Azure.
TD-39053	Cannot read datasets from Parquet files generated by Spark containing nested values. **Workaround:** In the source for the job, change the data types of the affected columns to String and re-run the job on Spark.
TD-39052	Signout using reverse proxy method of SSO is not working after upgrade.
TD-38869	Upload of Parquet files does not support nested values, which appear as null values in the Transformer page. **Workaround:** Unnest the values before importing into the platform.

Copyright © 2020 Trifacta Inc.

TD-37683	Send a copy does not create independent sets of recipes and datasets in new flow. If imported datasets are removed in the source flow, they disappear from the sent version. **Workaround:** Create new versions of the imported datasets in the sent flow.
TD-36145	Spark running environment recognizes numeric values preceded by + as Integer or Decimal data type. Trifacta Photon running environment does not and types these values as strings.
TD-35867	v3 publishing API fails when publishing to alternate S3 buckets **Workaround:** You can use the corresponding v4 API to perform these publication tasks. For more information on a workflow, see *API Workflow - Manage Outputs*.

Quick Start

Install from Azure Marketplace

Contents:

This documentation applies to installation from a supported Marketplace. Please use the installation instructions provided with your deployment.

If you are installing or upgrading a Marketplace deployment, please use the available PDF content. You must use the install and configuration PDF available through the Marketplace listing.

Copyright © 2020 Trifacta Inc.

This guide steps through the requirements and process for installing Trifacta® Wrangler Enterprise from the Azure Marketplace.

Product Limitations

- This Azure Marketplace listing creates a new storage account. You can add additional storage accounts after the product is deployed.

Documentation Scope

This document guides you through the process of installing the product and getting started.

Install

Desktop Requirements

- All desktop users of the platform must have a supported browser version installed on their desktops. Please update to the latest stable version that is publicly available.
- All desktop users must be able to connect to the created Trifacta node instance through the enterprise infrastructure.

Sizing Guide

You can use the following table to guide your selection of an appropriate instance type.

> **Tip:** The current minimum requirements are 8 cores and 64 GB of memory.

Azure virtual machine type	vCPUs	RAM (GB)	Max recommended concurrent users	Avg. input data size of jobs on Trifacta Server (GB)
Standard_E8s_v3	8	64	60	5
Standard_E16s_v3	16	128	90	11
Standard_E20s_v3	20	160	120	14

Prerequisites

Before starting the installation, please complete the following steps to ensure a smooth installation.

1. **License:** When you install the software, the included license is valid for 24 hours. You must acquire a license key file from Trifacta. For more information, please contact *Trifacta Support*.
2. **Registered Application/Service Principal:** The service principal is used by the Trifacta platform for access to all Azure resources. The process of creating a new service principal and collecting the required information is covered during the installation.

   > **Tip:** You will need the Service Principal information before deploying the template.

3. You will need to acquire multiple configuration values from the Azure Portal and store them for later configuration in the platform. It may be helpful to store these settings and their values in a text file.

Copyright © 2020 Trifacta Inc.

Required permissions

- Ability to create a Registered Application (to be used as a Service Principal).
- Ability to create all resources in the solution template.

Acquire application information

Please complete the following steps in the Azure Portal.

Steps:

1. Create registered application:
    1. In the Azure Portal, navigate to **Azure Active Directory > App Registrations**.
    2. Select **New Registration**.
    3. Enter a name for the new application. Leave the other settings at their default values.
    4. Click **Register**.
    5. On the following screen, acquire the values for the following fields:
        1. **Application (client) ID**
        2. **Directory (tenant) ID**
    6. Create a client secret:
        1. Navigate to **Certificates & Secrets**.
        2. Click **Add a client secret**.
        3. Fill in your preferred values. Click **Add**.
        4. Acquire the value for: **client secret value.**
    7. Add required API permissions:
        1. Navigate to **API Permissions**.
        2. Click **Add a permission**.
        3. Select **Azure Key Vault** from the list of Microsoft APIs.
        4. Select the **user_impersonation** checkbox.
        5. Select **Add permissions**.
2. Obtain the Object ID of the service principal:
    1. Navigate to **Azure Active Directory > Enterprise Applications**.

        > **NOTE:** Do not acquire the value from the App Registration area. Please use the Enterprise Applications area.

    2. Find the app that you registered in the previous step.
    3. Open the app. Acquire the value for the **Object ID**.

Deploy the solution

With the values that you acquired in the previous steps, you're ready to launch the Solution Template.

Steps:

1. Navigate to the Trifacta Wrangler Enterprise (with Databricks) listing on the Azure Marketplace.
2. Click **Create**:
    1. Select an existing Resource Group, or create a new one.
    2. Key Properties:

Property Name	Description
Deployment Name	Used when naming the resources created by the template
Admin Username	Use it to SSH into your server.

Copyright © 2020 Trifacta Inc.

	NOTE: You cannot use `trifacta` as the Admin Username.
SSH Public Key	The public key used to authenticate with the Admin Username user
Permitted SSH Source Range	Range of IP addresses in CIDR format from which SSH is reachable. See below.
Permitted Web Access Range	Range of IP addresses in CIDR format from which the web ports are reachable. See below.
Service Principal Object ID	Object ID value you acquired earlier in the Azure Portal. This value is used to automatically grant access to resources.

Notes on IP address ranges:

1. The default Trifacta port is `3005`. By default, ports `80` and `443` are also allowed, in case you update your Trifacta service to listen on those ports.
2. The IP ranges must be specified in CIDR format. For example, to specify a single IP address, such as `8.8.8.8/32`, use `/32`.
3. These settings can be modified or updated after the application has been deployed.

3. Click **Next**.
3. Review the parameters you've specified. If all looks good, click **Create**.
4. The deployment status is displayed. When deployment is complete, click **Go to Resource**.

> **Tip:** The deployment process may take a while. **Go to Resource** only appears when the deployment is complete.

5. Take a note of the **Virtual Machine Name**, which is your default Trifacta password.
6. Locate your Databricks service URL.
7. Select the **Databricks Service**.
8. Acquire the value for the **URL**. This value is used in later steps. Example:

```
https://centralus.azuredatabricks.net
```

9. Navigate back to the resource group.
10. Locate your Key Vault DNS name value.
11. Select the **Key vault** resource.
12. Acquire the value for the **DNS Name**. This value is used in later steps. Example:

```
https://trifacta2vrjkq.vault.azure.net/
```

13. Navigate back to the resource group.
14. Acquire the value for the **Storage Account Name** in the list of resources. This value is used in later steps.

Configure the Trifacta platform

The following configuration steps are performed inside the product.

> **NOTE:** The settings in this documentation are representations of JSON objects in the configuration file. For example, `fileStorage.defaultBaseUris` corresponds to the `defaultBaseUris` line within the `fileStorage` block.

Copyright © 2020 Trifacta Inc.

Steps:

1. In a new browser window, navigate to your Trifacta instance using the IP address of the VM and port 3005. Example: http://<YOUR_IP>:3005/
2. Log in using the username *admin@trifacta.local* and the name of your virtual machine (recorded earlier) as the password.
3. Navigate to the A at the bottom left > Settings> Admin Settings

    1. You must populate the following configuration items without saving in between.

> **NOTE:** Do not save your configuration or restart the platform until you have been directed to do so. If you accidentally save, the platform may fail to restart. The remaining changes would have to edited in the configuration via SSH.

Property	Description
azure.applicationid	Set value to: **Application (client) ID** you recorded from Azure Portal.
azure.secret	Set value to: **Client secret value** you recorded from Azure Portal.
azure.directoryid	Set value to: **Directory (tenant) ID** you recorded from Azure Portal.
webapp.storageProtocol	Set value to: `abfss`
aws.s3.enabled	Deselect this checkbox.
fileStorage.whitelist	Set this value to: `sftp,abfss` **NOTE:** Do not include a space between the values above.
fileStorage.defaultBaseUris	Example value: `abfss://trifacta@<StorageAccountName>.dfs.core.windows.net/` Where `<StorageAccountName>` is the storage account name you recorded from the Azure Portal.
webapp.runInDatabricks	Select this checkbox.
databricks.serviceUrl	Set value to: the Databricks Service URL you recorded from the Azure Portal.
azure.keyVaultUrl	Set value to: the Azure Key Vault URL you recorded from the Azure Portal.
azure.adl.store	Example value: `adl://<StorageAccountName>.azuredatalakestore.net`

	Where `<StorageAccountName>` is the storage account name you recorded from the Azure Portal.
azure.adl.enabled	Select this checkbox.
feature.databricks.connection.enabled	Select this checkbox.

4. Click **Save**.

> **Tip:** The application may take several minutes to restart.

Generate Databricks Personal Access Token

While the application is restarting, you must generate a Databricks Personal Access Token.

Steps:

1. In the Azure Portal, navigate to the Databricks resource that was deployed.
2. Click **Launch Workspace.**
3. Click the user icon at the top right.
4. Click **User Settings**.
5. The default page is Access Tokens. Click **Generate new Token**.
6. Copy the value of this token to the clipboard.
7. Return to the Trifacta application.
8. Refresh the page to ensure that the reboot has completed.
9. From the menu bar, select **User menu > Preferences > Databricks**.
10. Paste in the access token value from the clipboard.

Testing

You can now test the platform by running a simple job using your Databricks cluster.

After you have installed or made changes to the platform, you should verify end-to-end operations.

> **NOTE:** The Trifacta® platform is not operational until it is connected to a supported backend datastore.

Steps:

1. Login to the application as an administrator. See *Login*.
2. Through the Admin Settings page, run Tricheck, which performs tests on the Trifacta node and any connected cluster. See *Admin Settings Page*.
3. In the application menu bar, click **Library**. Click **Import Dataset**. Select your backend datastore.
4. Navigate your datastore directory structure to locate a small CSV or JSON file.
5. Select the file. In the right panel, click **Create and Transform**.
    1. **Troubleshooting:** If the steps so far work, then you have read access to the datastore from the platform. If not, please check permissions for the Trifacta user and its access to the appropriate directories.
    2. See *Import Data Page*.
6. In the Transformer page, some steps have already been added to your recipe, so you can run the job right away. Click **Run Job.**
    1. See *Transformer Page* .
7. In the Run Job Page:

Copyright © 2020 Trifacta Inc.

1. For Running Environment, some of these options may not be available. Choose according to the running environment you wish to test.
    1. **Photon:** Runs job on the Photon running environment hosted on the Trifacta node. This method of job execution does not utilize any integrated cluster.
    2. **Spark:** Runs the job on Spark on the integrated cluster.
    3.

       Databricks: If the platform is integrated with an Azure Databricks cluster, you can test job execution on the cluster.

       > **NOTE:** Use of Azure Databricks is not supported on Marketplace installs.
2. Select CSV and JSON output.
3. Select the Profile Results checkbox.
4. **Troubleshooting:** At this point, you are able to initiate a job for execution on the selected running environment. Later, you can verify operations by running the same job on other available environments .
5. See *Run Job Page*.

8. When the job completes, you should see a success message in the Jobs tab of the Flow View page.
    1. **Troubleshooting:** Either the Transform job or the Profiling job may break. To localize the problem, mouse over the Job listing in the Jobs page. Try re-running a job by deselecting the broken job type or running the job in a different environment. You can also download the log files to try to identify the problem. See *Jobs Page*.

9. Click **View Results** in the Jobs page. In the Profile tab of the Job Details page, you can see a visual profile of the generated results.
    1. See *Job Details Page*.
10. In the Output Destinations tab, click the CSV and JSON links to download the results to your local desktop. See *Import Data Page*.

11. Load these results into a local application to verify that the content looks ok.

Troubleshooting

White screen after applying settings changes

If you save your settings and the application restarts, showing only a white page, it's likely that you've missed a configuration setting or entered an incorrect value.

To fix:

1. Review the Admin Settings page to verify that there are no missing values, unexpected spaces, or similar issues.
2. Edit the configuration file directly:
    1. Log into the server via SSH.
    2. Make a backup of the following file: `/opt/trifacta/conf/trifacta-conf.json`.

       > **NOTE:** Save your backup outside of the install directory (`/opt/trifacta`).

    3. Review the settings in `trifacta-conf.json`.
3. The settings in this documentation are representations of JSON objects in the configuration file. For example, `fileStorage.defaultBaseUris` corresponds to the `defaultBaseUris` line within the `fileStorage` block.

Copyright © 2020 Trifacta Inc.

After you apply any fixes, you can restart the Trifacta service from the command line:

```
sudo service trifacta restart
```

"Principals of type Application cannot be validly be used in role assignments" error during deploy

This error can occur when you use the Object ID from the **App Registrations** area of the Portal, instead of the **Enterprise Applications** area.

The same name appears in both locations. However, only the Enterprise Applications Object ID is a valid service principal for assigning permissions to the storage account and key vault.

To fix:

Acquire and use the Object ID from the Enterprise Applications area.

Upgrade

For more information, see *Upgrade for Azure Marketplace*.

Documentation

You can access complete product documentation online and in PDF format. From within the product, select **Help menu > Documentation**.

Upgrade for Azure Marketplace

For more information on upgrading Trifacta® Wrangler Enterprise for the Azure Marketplace, please contact *Trifacta Support*.

Configure

The following topics describe how to configure Trifacta® Wrangler Enterprise for initial deployment.

- For more information on administration tasks and admin resources, see *Admin*.

Configure for Azure

Contents:

Copyright © 2020 Trifacta Inc.

- *Configure relational connections*
- *Testing*

Please complete the following steps in the listed order to configure your installed instance of the Trifacta® platform to integrate with the running environment cluster.

Pre-requisites

1. Deploy running environment cluster and Trifacta node.

   > **NOTE:** The running environment cluster can be deployed as part of the installation process. You can also integrate the platform with a pre-existing cluster. Details are below.

2. Install Trifacta platform on the node.

For more information, see *Install for Azure*.

Configure Azure

Create registered application

You must create a Azure Active Directory (AAD) application and grant it the desired access permissions, such as read/write access to the ADLS resource and read/write access to the Azure Key Vault secrets .

This service principal is used by the Trifacta platform for access to all Azure resources. For more information, see *https://docs.microsoft.com/en-us/azure/azure-resource-manager/resource-group-create-service-principal-portal*.

After you have registered, acquire the following information:

Azure Property	Location	Use
Application ID	Acquire this value from the Registered app blade of the Azure Portal.	Applied to Trifacta platform configuration: `azure.applicationid`.
Service User Key	Create a key for the Registered app in the Azure Portal.	Applied to Trifacta platform configuration: `azure.secret`.
Directory ID	Copy the Directory ID from the Properties blade of Azure Active Directory.	Applied to Trifacta platform configuration: `azure.directoryId`.

To create an Azure Active Directory (AAD) application, please complete the following steps in the Azure console.

Steps:

1. Create registered application:

   1. In the Azure console, navigate to **Azure Active Directory > App Registrations**.
   2. Create a New App. Name it `trifacta`.

      > **NOTE:** Retain the Application ID and Directory ID for configuration in the Trifacta platform.

2. Create a client secret:
   1. Navigate to **Certificates & secrets**.

Copyright © 2020 Trifacta Inc.

2. Create a new Client secret.

   NOTE: Retain the value of the Client secret for configuration in the Trifacta platform.

3. Add API permissions:
   1. Navigate to **API Permissions**.
   2. Add Azure Key Vault with the `user_impersonation` permission.

These properties are applied later in the configuration process.

Configure the Platform

Configure for HDI

If you are integrating the Trifacta platform with a pre-existing HDI cluster, additional configuration is required. See *Configure for HDInsight*.

NOTE: If you created a new HDI cluster as part of the installation, all required is listed below.

Configure for Azure Databricks

NOTE: Use of Azure Databricks is not supported for Marketplace installations.

You can integrate the Trifacta platform with Azure Databricks. For more information, see *Configure for Azure Databricks*.

Configure base storage layer

For Azure installations, you can set your base storage layer to be HDFS or WASB.

NOTE: The base storage layer must be set after installation. After it has been configured, it cannot be modified.

Azure storage	webapp.storageProtocol setting	hdfs.protocolOverride setting
WASB	`wasbs`	(empty)
ADLS	`hdfs`	`adl`

See *Set Base Storage Layer*.

Configure for Key Vault

For authentication purposes, the Trifacta platform must be integrated with an Azure Key Vault keystore. See *Configure Azure Key Vault*.

Configure for SSO

If needed, you can integrate the Trifacta platform with Azure AD for Single-Sign On to the platform. See *Configure SSO for Azure AD*.

Copyright © 2020 Trifacta Inc.

Configure for ADLS Gen2

Enable read-only or read-write access to ADLS Gen2. For more information, see *Enable ADLS Gen2 Access*.

Configure for ADLS

Enable read-only or read-write access to ADLS. For more information, see *Enable ADLS Access*.

Configure for WASB

Enable read-only or read-write access to WASB. For more information on integrating with WASB, see *Enable WASB Access*.

Configure relational connections

If you are integrating Trifacta Wrangler Enterprise with relational datastores, please complete the following configuration sections.

Create encryption key file

An encryption key file must be created on the Trifacta node. This key file is shared across all relational connections. See *Create Encryption Key File*.

Create Hive connection

You can create a connection to the Hive instance on the HDI cluster with some modifications.

- **High Availability:** Natively, Azure supports high availability for HiveServer2 via Zookeeper. Host and port information in the JDBC URL must be replaced with a Zookeeper quorum.

In addition to the other Hive connection properties, please specify the following values for the properties listed below:

Property	Description
Host	Use your Zookeeper quorum value. For the final node of the list, omit the port number. Example: `zk1.cloudapp.net:2181,zk2.cloudapp.net:2181,zk3.cloudapp.net`
Port	Set this value to `2181`.
Connect String options	In addition to any options required for your environment, include the following option: `/;serviceDiscoveryMode=zooKeeper;zooKeeperNamespace=hiveserver2`
Database	Enter your Hive database name.

Copyright © 2020 Trifacta Inc.

Connections are created through the Connections page. See *Connections Page*.

For additional details on creating a connection to Hive, see *Create Hive Connections*.

A Hive connection can also be created using the above property substitutions via programmatic methods.

- For details on values to use, see *Connection Types*.
- See *API Connections Create v4*.

Create Azure SQL Database connection

For more information, see *Create Azure SQL Database Connections*.

Create Azure SQL DW connection

For more information, see *Create SQL DW Connections*.

Testing

1. Load a dataset from the HDI cluster through either ADLS or WASB.
2. Perform a few simple steps on the dataset.
3. Click **Run Job** in the Transformer page.
4. When specifying the job:
    1. Click the Profile Results checkbox.
    2. Select **Spark**.
5. When the job completes, verify that the results have been written to the appropriate location.

Copyright © 2020 Trifacta Inc.

Configure Azure Key Vault

Contents:

For authentication purposes, the Trifacta® platform must be integrated with an Azure Key Vault keystore.

- For more information, see *https://azure.microsoft.com/en-us/services/key-vault/*.

Please complete the following sections to create and configure your Azure Key Vault.

Create a Key Vault resource in Azure

Please complete the following steps in the Azure portal to create a Key Vault and to associate it with the Trifacta registered application.

> **NOTE:** A Key Vault is required for use with the Trifacta platform.

Create Key Vault in Azure

Steps:

1. Log into the Azure portal.
2. Goto: *https://portal.azure.com/#create/Microsoft.KeyVault*
3. Complete the form for creating a new Key Vault resource:
   1. Name: Provide a reasonable name for the resource. Example:

      ```
      <clusterName>-<applicationName>-<group/organizationName>
      ```

      Or, you can use `trifacta`.
   2. Location: Pick the location used by the HDI cluster.
   3. For other fields, add appropriate information based on your enterprise's preferences.
4. To create the resource, click **Create**.

> **NOTE:** Retain the DNS Name value for later use.

Enable Key Vault access for the Trifacta platform

Steps:

In the Azure portal, you must assign access policies for application principal of the Trifacta registered application t o access the Key Vault.

Steps:

1. In the Azure portal, select the Key Vault you created. Then, select **Access Policies**.
2. In the Access Policies window, select the Trifacta registered application.
3. Click **Add New**.
4. Assign all Key, Secret, and Certificate permissions. For Secret permissions, be sure to select the following:
    1. Get
    2. Set
    3. Delete
5. Do not select any other options.
6. Click **OK**.

Configure Key Vault for WASB

Create WASB access token

If you are enabling access to WASB, you must create this token within the Azure Portal.

For more information, see *https://docs.microsoft.com/en-us/rest/api/storageservices/delegating-access-with-a-shared-access-signature*.

You must specify the storage protocol (`wasbs`) used by the Trifacta platform.

Configure Key Vault key and secret for WASB

In the Key Vault, you can create key and secret pairs for use.

Base Storage Layer	Description
ADLS	The Trifacta platform creates its own key-secret combinations in the Key Vault. No additional configuration is required. Please skip this section and populate the Key Vault URL into the Tr ifacta platform.
WASB	For WASB, you must create key and secret values that match other values in your Azure configuration. Instructions are below.

WASB: To enable access to the Key Vault, you must specify your key and secret values as follows:

Item	Applicable Configuration
key	The value of the key must be specified as the `sasTokenId` in the Trifacta platform.
secret	The value of the secret should match the shared access signature for your storage. This value is specified as `sasToken` in the Trif acta platform.

Acquire shared access signature value:

In the Azure portal, please do the following:

1. Open your storage account.
2. Select **Shared Access Signature** .
3. Generate or view existing signatures.
4. For a new or existing signature, copy the SAS token value. Omit the leading question mark (?).
5. Paste this value into a text file for safekeeping.

Copyright © 2020 Trifacta Inc.

Create a custom key:

To create a custom key and secret pair for WASB use by the Trifacta platform, please complete the following steps:

1. On an existing or newly created Azure Key Vault resource, click **Secrets**.
2. At the top of the menu, click **Generate/Import**.
3. In the Create a secret menu:
    1. Select **Manual** for upload options.
    2. Chose an appropriate name for the key.

        > **NOTE:** Please retain the name of the key for later use, when it is applied through the Trifact a platform as the `sasTokenId` value. Instructions are provided later.

    3. Paste the SAS token value for the key into the secret field.
    4. Click **Create**.

Configure Key Vault Location

For ADLS or WASB, the location of the Azure Key Vault must be specified for the Trifacta platform. The location can be found in the properties section of the Key Vault resource in the Azure portal.

Steps:

1. Log in to the Azure portal.
2. Select the Key Vault resource.
3. Click **Properties**.
4. Locate the DNS Name field. Copy the field value.

This value is the location for the Key Vault. It must be applied in the Trifacta platform.

Steps:

1. You can apply this change through the *Admin Settings Page* (recommended) or `trifacta-conf.json`. For more information, see *Platform Configuration Methods*.
2. Specify the URL in the following parameter:

```
"azure.keyVaultURL": "<your key value URL>",
```

Apply SAS token identifier for WASB

If you are using WASB as your base storage layer, you must apply the SAS token value into the configuration of the Trifacta platform.

Steps:

1. You can apply this change through the *Admin Settings Page* (recommended) or `trifacta-conf.json`. For more information, see *Platform Configuration Methods*.
2. Paste the value of the SAS Token for the key you created in the Key Vault as the following value:

```
"azure.wasb.defaultStore.sasTokenId": "<your Sas Token Id>",
```

3. Save your changes.

Copyright © 2020 Trifacta Inc.

Configure Secure Token Service

Access to the Key Vault requires use of the secure token service (STS) from the Trifacta platform. To use STS with Azure, the following properties must be specified.

> **NOTE:** Except in rare cases, the other properties for secure token service do not need to be modified.

You can apply this change through the *Admin Settings Page* (recommended) or `trifacta-conf.json`. For more information, see *Platform Configuration Methods*.

Property	Description
" secure-token-service .autorestart"	Set this value to `true` to enable auto-restarting of the secure token service.
"secure-token-service.port"	Set this value to `8090`.
"com.trifacta.services.secure_token_service. refresh_token_encryption_key"	Enter a base64 string to serve as your encryption key for the refresh token of the secure token service. A default encryption key is inserted for you. **NOTE:** If a valid base64 string value is not provided here, the platform fails to start.
"secure-token-service.userIdHashingPepper"	Enter a base64 string.

Configure SSO for Azure AD

Contents:

When the Trifacta® platform is deployed on Azure, it can be configured to provide single sign-on (SSO) with Azure AD (Active Directory) authentication management. Use this section to enable auto-logins for Azure users.

- If auto-provisioning is not desired, after completing the basic configuration, you can disable auto-provisioning using the steps listed in the Advanced Configuration section.

Copyright © 2020 Trifacta Inc.

- Single Sign-On (SSO) authentication enables users to authenticate one time to access multiple systems. The SSO platform must translate its authentication into authentication methods executed against each system under SSO control. For more information, see *https://en.wikipedia.org/wiki/Single_sign-on*.
- When enabled, SSO also applies to the Wrangler Enterprise desktop application, if it is installed.

Supported authentication models:

Users can authenticate with the Trifacta platform using Azure AD accounts in the following scenarios:

- Azure AD is the identity provider,
- Azure AD is federated through a trust setup with a supported external identity provider,
- Azure AD is federated with on-premises Active Directory and Active Directory Federation Services (ADFS).

Azure Data Lake Store: Users can obtain OAuth access and refresh tokens from AzureAD and use the tokens to access ADLS.

Domain-Joined Clusters: Using Azure AD, the Trifacta platform can be deployed to a domain-joined HDInsight cluster and can run jobs as the authenticated AD user via secure impersonation. For more information, see *Configure for HDInsight*.

Azure Databricks Clusters: If you have integrated with an Azure Databricks cluster, please complete this configuration to enable SSO authentication for Azure. No additional configuration is required to enable SSO for Azure Databricks.

Pre-Requisites

1. You have installed the Trifacta platform on Microsoft Azure. See *Install for Azure*.
2. You have performed the basic configuration for Azure integration. See *Configure for Azure*.
3. Your enterprise uses Azure SSO for User Identity and Authentication.
4. The Trifacta platform must be registered as a Service Provider in your Azure AD tenant.
5. Please acquire the following Service Provider properties:
    1. The Service Provider Application ID (Client ID) and Key (Secret) are used for user authentication to the Azure Key Vault, Azure AD, and Azure Data Lake Store (if connected). These properties are specified in the Trifacta platform as part of the basic Azure configuration.

       > **NOTE:** The Trifacta platform must be assigned the Reader role for the Azure Key Vault. Other permissions are also required. See the Azure Key Vault Permissions section below.

    2. The Service Provider Reply URL provides the redirect URL after the user has authenticated with Azure AD.
    3. The Service Provider should be granted Delegated permissions to the Windows Azure Service Management API so it can access Azure Service Management as organization users.

Limitations

Scheduled jobs are run under the access keys for the user who initially created the schedule. They continue to run as scheduled until those keys are explicitly revoked by an admin.

> **NOTE:** With Azure SSO enabled, use of custom dictionaries is not supported.

Configure Azure AD for Trifacta platform

Please verify or perform the following configurations through Azure.

Copyright © 2020 Trifacta Inc.

Azure Key Vault Permissions

For the Azure Key Vault:

- The Trifacta application must be assigned the Reader permission to the key vault.
- For the Key Vault Secrets, the application must be assigned the Set, Get, and Delete permissions.

Configure Trifacta platform for Azure AD

Azure AD Properties

Please configure the following properties.

You can apply this change through the *Admin Settings Page* (recommended) or `trifacta-conf.json`. For more information, see *Platform Configuration Methods*.

Property	Description
azure.sso.enabled	Set this value to `true` to enable Azure AD Single Sign-On. The Trifacta platform authenticates users through enterprise Azure AD.
azure.sso.redirectUrl	Set this value to the redirect URL callback configured for this Azure AD application in the Azure portal. The URL is in the following format: `https://<trifacta-app-host>/sign-in/azureCallback`
azure.sso.allowHttpForRedirectUrl	When `true`, the `redirectUrl` can be specified as an insecure, non-HTTPS value. Default is `false`.
azure.sso.enableAutoRegistration	Set this value to `true` to enable SSO users to automatically register and login to the Trifacta application when they connect.
azure.resourceURL	This value defines the Azure AD resource for which to obtain an access token. **NOTE:** By default, this value is `https://graph.windows.net/` . You can select other values from the drop-down in the Admin Settings page. When using Azure Data Lake: 1. In the Azure Portal, grant to the Trifacta application ID the Azure Data Lake API permission. 2. Set this value to `https://datalake.azure.net/` . 3. Sign out of the Trifacta application and sign in again.

User Management

Tip: After SSO is enabled, the first AD user to connect to the platform is automatically registered as an admin user.

Configure auto-registration

Enabling auto-registration:

Auto-registration must be enabled for the Trifacta platform and for Azure AD SSO specifically.

You can apply this change through the *Admin Settings Page* (recommended) or `trifacta-conf.json`. For more information, see *Platform Configuration Methods*.

Property	Description
`webapp.sso.enableAutoRegistration`	This property has no effect in Azure.
`azure.sso.enableAutoRegistration`	Set this value to `true`. For more information, see Azure AD Properties above.

How users are managed depends on whether auto-registration is enabled:

- If auto-registration is enabled, after users provide their credentials, the account is automatically created for them.
- If auto-registration is disabled, a Trifacta administrator must still provision a user account before it is available. See below.

Enabled:

After SSO with auto-registration has been enabled, you can still manage users through the Admin Settings page, with the following provisions:

- The Trifacta platform does not recheck for attribute values on each login. If attribute values change in LDAP, they must be updated in the User Management page, or the user must be deleted and recreated through auto-provisioning.
- If the user has been removed from AD, the user cannot sign in to the platform.
- If you need to remove a user from the platform, you should consider just disabling the user through the User Management area.

For more information, see *Manage Users*.

Disabled:

To disable auto-provisioning in the platform, please verify the following property:

1. You can apply this change through the *Admin Settings Page* (recommended) or `trifacta-conf.json`. For more information, see *Platform Configuration Methods*.
2. Set the following property:

```
"webapp.sso.enableAutoRegistration" : false,
```

3. Save your changes and restart the platform.
4. New users of the Trifacta platform must be provisioned by a Trifacta administrator. See below.

Provision new users under SSO without auto-registration

If SSO auto-registration is disabled, admin users can provision new users of the platform through the following URL:

```
https://<hostname>/register
```

`http://<host_name>:<port_number>/register`

Copyright © 2020 Trifacta Inc.

- The user's password is unnecessary in an SSO environment. You must provide the SSO principal value, which is typically the Active Directory login for the user.
- If you are connected to a Hadoop cluster, you must provision the Hadoop principal value.
- See *Create User Account*.
- Admin accounts can be created through the application. See *Create Admin Account*.

Disable user

> **If a user has been disabled in Azure AD, a Trifacta administrator must disable the user in the Trifacta application. Otherwise, the user can still access the Trifacta application until the user's access token expires.**

For more information on disabling user accounts, see *Manage Users*.

User Access

Users access the application through the Trifacta login page:

```
https://<hostname>
```

SSO Relational Connections

For more information, see *Enable SSO for Azure Relational Connections*.

Enable ADLS Gen2 Access

Contents:

Microsoft Azure deployments can integrate with with the next generation of Azure Data Lake Store (ADLS Gen2).

- **Microsof Azure Data Lake Store Gen2 (ADLS Gen2)** combines the power of a high-performance file system with massive scale and economy. Azure Data Lake Storage Gen2 extends Azure Blob Storage capabilities and is optimized for analytics workloads.
- For more information, see *https://azure.microsoft.com/en-us/services/storage/data-lake-storage/*.

Copyright © 2020 Trifacta Inc.

Limitations of ADLS Gen2 Integration

- This version requires a specific CDH 6.1 bundle JAR. Details are described later.

Read-only access

If the base storage layer has been set to WASB, you can follow these instructions to set up read-only access to ADLS Gen2.

> **NOTE:** To enable read-only access to ADLS Gen2, do not set the base storage layer to `abfss`.

Pre-requisites

General

- The Trifacta platform has already been installed and integrated with an Azure Databricks cluster. See *Configure for Azure Databricks*.
- For each combination of blob host and container, a separate Azure Key Vault Store entry must be created. For more information, please contact your Azure admin.
- When running against ADLS Gen2, the product requires that you create a filesystem object in your ADLS Gen2 storage account.
- ABFSS must be set as the base storage layer for the Trifacta platform instance. See *Set Base Storage Layer*.

Create a registered application

Before you integrate with Azure ADLS Gen2, you must create the Trifacta platform as a registered application. See *Configure for Azure*.

Azure properties

The following properties should already be specified in the Admin Settings page. Please verify that the following have been set:

- `azure.applicationId`
- `azure.secret`
- `azure.directoryId`

The above properties are needed for this configuration.

> **Tip:** ADLS Gen2 also works if you are using Azure Managed Identity.

Registered application role

> **NOTE:** The Storage Blob Data Contributor role or its equivalent roles must be assigned in the ADLS Gen2 storage account.

For more information, see *Configure for Azure*.

Key Vault Setup

An Azure Key Vault has already been set up and configured for use by the Trifacta platform. Properties must be specified in the platform, if they have not been configured already.

Copyright © 2020 Trifacta Inc.

For more information on configuration for Azure key vault, see *Configure for Azure*.

Configure the Trifacta platform

Specify CDH 6.1 bundle JARs

> **NOTE:** For this release, use of the CDH 6.1 bundle JARs is required for ADLS Gen2 integration. Please specify all of these properties, even if you are not integrating with Hive.

Please complete the following steps to review and modify if necessary the bundle JAR properties and dependency locations.

Steps:

1. You can apply this change through the *Admin Settings Page* (recommended) or `trifacta-conf.json`. For more information, see *Platform Configuration Methods*.
2. Locate the following parameters and set them to the values listed below:

   > **NOTE:** If you have integrated with Databricks Tables, do not overwrite the value for `data-service.hiveJdbcJar` with the following value, even if it's set to a different distribution JAR file.

   ```
   "hadoopBundleJar": "hadoop-deps/cdh-6.1/build/libs/cdh-6.1-bundle.
   jar",
   "spark-job-service.hiveDependenciesLocation": %(topOfTree)s/hadoop-
   deps/cdh-6.1/build/libs"
   "data-service.hiveJdbcJar": "hadoop-deps/cdh-6.1/build/libs/cdh-6.1-
   hive-jdbc.jar",
   ```

3. Save your changes and restart the platform.

Define base storage layer

Per earlier configuration:

- `webapp.storageProtocol` must be set to `abfss`.

  > **NOTE:** Base storage layer must be configured when the platform is first installed and cannot be modified later.

- `hdfs.protocolOverride` is ignored.

See *Set Base Storage Layer*.

Review Java VFS Service

Use of ADLS Gen2 requires the Java VFS service in the Trifacta platform.

> **NOTE:** This service is enabled by default.

For more information on configuring this service, see *Configure Java VFS Service*.

Copyright © 2020 Trifacta Inc.

Configure file storage protocols and locations

The Trifacta platform must be provided the list of protocols and locations for accessing ADLS Gen2 blob storage.

Steps:

1. You can apply this change through the *Admin Settings Page* (recommended) or `trifacta-conf.json`. For more information, see *Platform Configuration Methods*.
2. Locate the following parameters and set their values according to the table below:

```
"fileStorage.whitelist": ["abfss"],
"fileStorage.defaultBaseUris": ["abfss://filesystem@storageaccount.
dfs.core.windows.net/"],
```

Parameter	Description
filestorage.whitelist	A comma-separated list of protocols that are permitted to read and write with ADLS Gen2 storage. **NOTE:** The protocol identifier `"abfss"` must be included in this list.
filestorage.defaultBaseUris	For each supported protocol, this param must contain a top-level path to the location where Trifacta platform files can be stored. These files include uploads, samples, and temporary storage used during job execution. **NOTE:** A separate base URI is required for each supported protocol. You may only have one base URI for each protocol.

3. Save your changes and restart the platform.

Configure access mode

Authentication to ADLS Gen2 storage is supported for `system` mode only.

Mode	Description
System	All users authenticate to ADLS using a single system key/secret combination. This combination is specified in the following parameters, which you should have already defined: • `azure.applicationId` • `azure.secret` • `azure.directoryId` These properties define the registered application in Azure Active Directory. System authentication mode uses the registered application identifier as the service principal for authentication to ADLS. All users have the same permissions in ADLS. For more information on these settings, see *Configure for Azure*.
User	

NOTE: User mode, Azure AD, and Azure SSO are not supported for use with ADLS Gen2.

Steps:

Please verify the following steps to specify the ADLS access mode.

1. You can apply this change through the *Admin Settings Page* (recommended) or `trifacta-conf.json`. For more information, see *Platform Configuration Methods*.
2. Verify that the following parameter to `system`:

```
"azure.adlsgen2.mode": "system",
```

3. Save your changes.

Testing

Restart services. See *Start and Stop the Platform*.

After the configuration has been specified, an ADLS Gen2 connection appears in the Import Data page. Select it to begin navigating for data sources.

NOTE: If you have multiple ADLS Gen2 file systems or storage accounts, you can access the secondary ones through the ADLS Gen2 browser. Edit the URL path in the browser and paste in the URI for other locations.

Try running a simple job from the Trifacta application. For more information, see *Verify Operations*.

- Except as noted above, the ADLS Gen2 browser is identical to the ADLS one. See *\ADLS Gen2 Browser*.
- Except as noted above, the basic usage of ADLS Gen2 is identical to the ADLS (Gen1) version. See *Using ADLS*.

Enable ADLS Access

Contents:

Copyright © 2020 Trifacta Inc.

By default, Microsoft Azure deployments integrate with Azure Data Lake Store (ADLS). Optionally, you can configure your deployment to integrate with WASB.

- **Microsof Azure Data Lake Store (ADLS)** is a scalable repository for big data analytics.
- ADLS is accessible from Microsoft HDI and Azure Databricks.
- For more information, see *https://docs.microsoft.com/en-us/azure/data-lake-store/data-lake-store-overview*.

Limitations of ADLS Integration

- In this release, the Trifacta platform supports integration with the default store only. Extra stores are not supported.

Read-only access

If the base storage layer has been set to WASB, you can follow these instructions to set up read-only access to ADLS.

> **NOTE:** To enable read-only access to ADLS, do not set the base storage layer to `hdfs`. The base storage layer for ADLS read-write access must remain `wasbs`.

Pre-requisites

General

- The Trifacta platform has already been installed and integrated with an Azure Databricks cluster. See *Configure for Azure Databricks*.
- HDFS must be set as the base storage layer for the Trifacta platform instance. See *Set Base Storage Layer*.
- For each combination of blob host and container, a separate Azure Key Vault Store entry must be created. For more information, please contact your Azure admin.

Create a registered application

Before you integrate with Azure ADLS, you must create the Trifacta platform as a registered application. See *Configure for Azure*.

Azure properties

The following properties should already be specified in the Admin Settings page. Please verify that the following have been set:

- `azure.applicationId`
- `azure.secret`
- `azure.directoryId`

The above properties are needed for this configuration. For more information, see *Configure for Azure*.

Key Vault Setup

An Azure Key Vault has already been set up and configured for use by the Trifacta platform. For more information, see *Configure for Azure*.

Configure ADLS Authentication

Authentication to ADLS storage is supported for the following modes, which are described in the following section.

Mode	Description
System	All users authenticate to ADLS using a single system key/secret combination. This combination is specified in the following parameters, which you should have already defined: • `azure.applicationId` • `azure.secret` • `azure.directoryId` These properties define the registered application in Azure Active Directory. System authentication mode uses the registered application identifier as the service principal for authentication to ADLS. All users have the same permissions in ADLS. For more information on these settings, see *Configure for Azure*.
User	Per-user mode allows individual users to authenticate to ADLS through their Azure Active Directory login. **NOTE:** Additional configuration for AD SSO is required. Details are below.

Steps:

Please complete the following steps to specify the ADLS access mode.

1. You can apply this change through the *Admin Settings Page* (recommended) or `trifacta-conf.json`. For more information, see *Platform Configuration Methods*.
2. Set the following parameter to the preferred mode (`system` or `user`):

```
"azure.adl.mode": "<your_preferred_mode>",
```

3. Save your changes.

System mode access

When access to ADLS is requested, the platform uses the combination of Azure directory ID, Azure application ID, and Azure secret to complete access.

After defining the properties in the Trifacta platform, system mode access requires no additional configuration.

User mode access

In user mode, a user ID hash is generated from the Key Vault key/secret and the user's AD login. This hash is used to generate the access token, which is stored in the Key Vault.

Set up for Azure AD SSO

NOTE: User mode access to ADLS requires Single Sign On (SSO) to be enabled for integration with Azure Active Directory. For more information, see *Configure SSO for Azure AD*.

Configure the Trifacta platform

Define default storage location and access key

In platform configuration, you must define the following properties:

```
"azure.adl.store": "<your_value_here>",
```

This property defines the ADLS storage to which all output data is delivered. Example:

```
adl://<YOUR_STORE_NAME>.azuredatalakestore.net
```

Per earlier configuration:

- `webapp.storageProtocol` must be set to `hdfs`.
- `hdfs.protocolOverride` must be set to `adl`.

Configure HDFS properties

In the Trifacta platform, you must configure the following properties for effective communication with HDFS.

```
"hdfs": {
 "username": "[hadoop.user]",
 "enabled": true,
 "webhdfs": {
  "httpfs": false,
  "maprCompatibilityMode": false,
  "ssl": {
   "enabled": true,
   "certificateValidationRequired": false,
   "certificatePath": "<YOUR_PATH_HERE>"
  },
  "host": "[ADLS].azuredatalakestore.net",
  "version": "/webhdfs/v1",
  "proxy": {
   "host": "proxy",
   "enabled": false,
   "port": 8080
  },
  "credentials": {
   "username": "[hadoop.user]",
   "password": ""
  },
  "port": 443
 },
 "protocolOverride": "adl",
 "highAvailability": {
  "serviceName": "[ADLS].azuredatalakestore.net",
  "namenodes": {}
 },
 "namenode": {
  "host": "[ADLS].azuredatalakestore.net",
  "port": 443
 }
}
```

Copyright © 2020 Trifacta Inc.

Property	Description
`hdfs.username`	Set this value to the name of the user that the Trifacta platform uses to access the cluster.
`hdfs.enabled`	Set to `true`.
`hdfs.webhdfs.httpfs`	Use of HttpFS in this integration is not supported. Set this value to `false`.
`hdfs.webhdfs.maprCompatibilityMode`	This setting does not apply to ADLS. Set this value to `false`.
`hdfs.webhdfs.ssl.enabled`	SSL is always used for ADLS. Set this value to `true`.
`hdfs.webhdfs.ssl.certificateValidationRequired`	Set this value to `false` .
`hdfs.webhdfs.ssl.certificatePath`	This value is not used for ADLS.
`hdfs.webhdfs.host`	Set this value to the address of your ADLS datastore.
`hdfs.webhdfs.version`	Set this value to `/webhdfs/v1`.
`hdfs.webhdfs.proxy.host`	This value is not used for ADLS.
`hdfs.webhdfs.proxy.enabled`	A proxy is not used for ADLS. Set this value to `false` .
`hdfs.webhdfs.proxy.port`	This value is not used for ADLS.
`hdfs.webhdfs.credentials.username`	Set this value to the name of the user that the Trifacta platform uses to access the cluster.
`hdfs.webhdfs.credentials.password`	Leave this value empty for ADLS.
`hdfs.webhdfs.port`	Set this value to `443`.
`hdfs.protocolOverride`	Set this value to `adl`.
`hdfs.highAvailability.serviceName`	Set this value to the address of your ADLS datastore.
`hdfs.highAvailability.namenodes`	Set this value to an empty value.
`hdfs.namenode.host`	Set this value to the address of your ADLS datastore.
`hdfs.namenode.port`	Set this value to `443`.

Enable

Steps:

1. You can apply this change through the *Admin Settings Page* (recommended) or `trifacta-conf.json`. For more information, see *Platform Configuration Methods*.
2. Locate the following parameter and change its value to `true`:

```
"azure.adl.enabled": true,
```

3. Configure use of the appropriate Hadoop bundle JAR:

Copyright © 2020 Trifacta Inc.

```
"hadoopBundleJar": "hadoop-deps/hdp-2.6/build/libs/hdp-2.6-bundle.
jar",
```

4. Save your changes.

Testing

Restart services. See *Start and Stop the Platform*.

After the configuration has been specified, an ADLS connection appears in the Import Data page. Select it to begin navigating for data sources.

Try running a simple job from the Trifacta application. For more information, see *Verify Operations*.

- See *ADLS Browser*.
- See *Using ADLS*.

Enable WASB Access

Contents:

By default, Microsoft Azure deployments integrate with Azure Data Lake Store (ADLS). Optionally, you can configure your deployment to integrate with WASB.

- **Windows Azure Storage Blob (WASB)** is an abstraction layer on top of HDFS, which enables persistence of storage, access without a Hadoop cluster presence, and access from multiple Hadoop clusters.

Limitations of WASB Integration

- If a directory is created on the HDI cluster through WASB, the directory includes a Size=0 blob. The Trifact a platform does not list them and does not support interaction with Size=0 blobs.

Read-only access

If the base storage layer has been set to ADLS, you can follow these instructions to set up read-only access to WASB.

Copyright © 2020 Trifacta Inc.

NOTE: If you are adding WASB as a secondary integration to ADLS, your WASB blob container or containers must contain at least one folder. This is a known issue.

NOTE: To enable read-only access to WASB, do not set the base storage layer to `wasbs`. The base storage layer for ADLS read-write access must remain `hdfs`.

Pre-requisites

General

- The Trifacta platform has already been installed and integrated with an Azure HDI or Azure Databricks cluster.
- WASB must be set as the base storage layer for the Trifacta platform instance. See *Set Base Storage Layer*.
- For each combination of blob host and container, a separate Azure Key Vault Store entry must be created. For more information, please contact your Azure admin.

Create a registered application

Before you integrate with Azure ADLS, you must create the Trifacta platform as a registered application. See *Configure for Azure*.

Other Azure properties

The following properties should already be specified in the Admin Settings page. Please verify that the following have been set:

- `azure.applicationId`
- `azure.secret`
- `azure.directoryId`

The above properties are needed for this configuration. For more information, see *Configure for Azure*.

Key Vault Setup

For new installs, an Azure Key Vault has already been set up and configured for use by the Trifacta platform.

NOTE: An Azure Key Vault is required. Upgrading customers who do not have a Key Vault in their environment must create one.

For more information, see *Configure for Azure*.

Configure WASB Authentication

Authentication to WASB storage is managed by specifying the appropriate host, container, and token ID in the Trif acta platform configuration. When access to WASB is requested, the platform passes the information through the Secure Token Service to query the specified Azure Key Vault Store using the provided values. The keystore returns the value for the secret. The combination of the key (token ID) and secret is used to access WASB.

NOTE: Per-user authentication is not supported for WASB.

Copyright © 2020 Trifacta Inc.

For more information on creating the Key Vault Store and accessing it through the Secure Token Service, see

Configure for Azure.

Configure the Trifacta platform

Define location of SAS token

The SAS token required for accessing Azure can be accessed from either of the following locations:

1. Key Vault
2. Trifacta configuration

whether SAS token is to be retrieved from Azure Key Vault or Configuration

SAS token from Key Vault

To store the SAS token in the key vault, specify the following parameters in platform configuration. You can apply this change through the *Admin Settings Page* (recommended) or `trifacta-conf.json`. For more information, see *Platform Configuration Methods*.

Secret names used for extra stores

If you are enabling extra WASB stores, specify the secret name to be used to access the SAS token from the Key Vault per extra Store.

> **NOTE:** Additional configuration is required for enabling extra WASB stores. See below.

```
"azure.wasb.extraStores": [ {
        ...
        "keyVaultSasTokenSecretName": "<secret_name>"
  }, {
        ...
        "keyVaultSasTokenSecretName": "<secret_name>"
  }
]
```

Parameter	Description
`"azure.wasb. fetchSasTokensFromKeyVault": true,`	Instructs the Trifacta platform to query the Key Vault for SAS tokens **NOTE:** The Key Vault must already be set up. See "Key Vault Setup" above.
`"azure.wasb.defaultStore.`	The default store's SAS token secret name to retrieve the SAS

`keyVaultSasTokenSecretName": "<your_value_here>",`	token for the default store from the Azure Key Value Store.

SAS token from Trifacta configuration

To specify the SAS token in the Trifacta platform configuration, set the following flag to false and then specify the SAS token per container.

Parameter	Description
`"azure.wasb. fetchSasTokensFromKeyVault": false,`	Instructs the Trifacta platform to acquire per-container SAS tokens from the platform configuration.
`"azure.wasb.defaultStore.sasToken": "<your_value_here>",`	Specify the SAS token here for the default store, if `azure.wasb.fetchSasTokensFromKeyVault` is set to `false`.

SAS token for extra WASB stores

If you are enabling extra WASB stores and `azure.wasb.fetchSasTokensFromKeyVault` is set to `false`, specify the sasToken for each extra store.

NOTE: Additional configuration is required for enabling extra WASB stores. See below.

```
"azure.wasb.extraStores": [ {
        ...
        "sasToken": "<your_value_here>"
  }, {
        ...
        "sasToken": "<your_value_here>"
  }
]
```

Define default storage location and access key

In platform configuration, you must define the following properties. When these properties are specified, the platform acquires the secret for the specified token ID, which is used to gain access to WASB.

Storage account

Azure path to the location where your data is to be stored.

```
"azure.wasb.defaultStore.blobHost": "<your_value_here>",
```

Container

Within your storage location, this value defines the default container for storing data.

Copyright © 2020 Trifacta Inc.

```
"azure.wasb.defaultStore.container": "<your_value_here>",
```

Define extra stores

If you have additional WASB stores, you can specify access to them for the Trifacta platform. Users of the platform can use them for reading sources and writing results.

Steps:

1. To apply this configuration change, login as an administrator to the Trifacta node. Then, edit `trifacta-conf.json`. Some of these settings may not be available through the *Admin Settings Page*. For more information, see *Platform Configuration Methods*.
2. Locate the `azure.wasb.extraStores` configuration block and add the following parameters:

```
"azure.wasb.extraStores":
    "extraStores": [
     {
      "sasToken": "<VALUE1_HERE",
      "keyVaultSasTokenSecretName": "<VALUE1_HERE>",
      "container": "<VALUE1_HERE>",
      "blobHost": "<VALUE1_HERE>"
     },
     {
      "sasToken": "VALUE2_HERE",
      "keyVaultSasTokenSecretName": "<VALUE2_HERE>",
      "container": "<VALUE2_HERE>",
      "blobHost": "<VALUE2_HERE>"
     }
    ]
   },
 },
```

Parameter	Description
`sasToken`	Set this value to `SAS token`, if applicable.
`keyVaultSasTokenSecretName`	To use the same SAS token as used in default storage, set this value to the same SAS token ID. If needed, you can generate and apply a per-container SAS token for use in this field for this specific store. Details are below.
`container`	Apply the name of the WASB container. **NOTE:** If you are specifying different blob host and container combinations for your extra stores, you must create a new Key Vault store. See above for details.
`blobHost`	Specify the blob host for the extra store.

NOTE: If you are specifying different blob host and container combinations for your extra stores, you must create a new Key Vault store. See above for details.

3. Save your changes and restart the platform.

Generate per-container SAS token

Execute the following command at the command line to generate a SAS token for a specific container:

```
Set-AzureRmStorageAccount -Name 'name'
$sasToken = New-AzureStorageContainerSASToken -Permission r -ExpiryTime
(Get-Date).AddHours(2.0) -Name '<container_name>'
```

Configure storage protocol

You must configure the platform to use the WASBS (secure) storage protocol when accessing.

Steps:

1. You can apply this change through the *Admin Settings Page* (recommended) or `trifacta-conf.json`. For more information, see *Platform Configuration Methods*.
2. Locate the following parameter and change its value `wasbs` for secure access:

```
"webapp.storageProtocol": "wasbs",
```

3. Save your changes and restart the platform.

Enable

Steps:

1. You can apply this change through the *Admin Settings Page* (recommended) or `trifacta-conf.json`. For more information, see *Platform Configuration Methods*.
2. Locate the following parameter and change its value to `true`:

```
"azure.wasb.enabled": true,
```

3. Configure use of the appropriate Hadoop bundle JAR:

```
"hadoopBundleJar": "hadoop-deps/hdp-2.6/build/libs/hdp-2.6-bundle.
jar";
```

4. Save your changes and restart the platform.

Testing

Restart services. See *Start and Stop the Platform*.

Copyright © 2020 Trifacta Inc.

After the configuration has been specified, a WASB connection appears in the Import Data page. Select it to begin navigating through the WASB Browser for data sources.

Try running a simple job from the Trifacta application. For more information, see *Verify Operations*.

- See *WASB Browser*.
- See *Using WASB*.

Configure for Azure Databricks

Contents:

This section describes how to configure the Trifacta® platform to integrate with Databricks hosted in Azure.

- Azure Databricks is an Apache Spark implementation that has been optimized for use on the Azure platform. For more information, see *https://databricks.com/product/azure*.

NOTE: For each user, a separate cluster is created. It may take a few minutes to spin up a new cluster.

Pre-requisites

- The Trifacta platform must be deployed in Microsoft Azure.

Limitations

- Supported for Azure Databricks versions 5.3 - 5.5 LTS.
- The Trifacta platform must be installed on Microsoft Azure.
- Nested folders are not supported when running jobs from Azure Databricks.
- When a job is started and no cluster is available, a cluster is initiated, which can take up to four minutes. If the job is canceled during cluster startup:
  - The job is terminated, and the cluster remains.
  - The job is reported in the application as Failed, instead of Canceled.
- Azure Databricks integration works with Spark 2.4.x only.

Copyright © 2020 Trifacta Inc.

NOTE: The version of Spark for Azure Databricks must be applied to the platform configuration through the `databricks.sparkVersion` property. Details are provided later.

- Azure Databricks integration does not work with Hive.

Job counts

By default, the number of jobs permitted on an Azure Databricks cluster is set to `1000`.

- The number of jobs that can be created per workspace in an hour is limited to `1000`.
- These limits apply to any jobs run for workspace data on the cluster.
- The number of actively concurrent job runs in a workspace is limited to `150`.

NOTE: To enable retrieval and auditing of job information after a job has been completed, the Trifacta platform does not delete jobs from the cluster. As a result, jobs can accumulate over time to exceeded the number of jobs permitted on the cluster. You should periodically delete jobs on your Azure Databricks cluster to prevent reaching these limits and receiving a `Quota for number of jobs has been reached` limit.

For more information, see *https://docs.databricks.com/user-guide/jobs.html*.

Enable

To enable Azure Databricks, please perform the following configuration changes.

Steps:

1. You can apply this change through the *Admin Settings Page* (recommended) or `trifacta-conf.json`. For more information, see *Platform Configuration Methods*.
2. Locate the following parameters. Set them to the values listed below, which enable the Trifacta Photon (smaller jobs) and Azure Databricks (small to extra-large jobs) running environments:

```
"webapp.runInTrifactaServer": true,
"webapp.runInDatabricks": true,
"webapp.runWithSparkSubmit": false,
"webapp.runinEMR": false,
"webapp.runInDataflow": false,
"photon.enabled": true,
```

3. Do not save your changes until you have completed the following configuration section.

Configure

Configure Platform

Please review and modify the following configuration settings.

NOTE: When you have finished modifying these settings, save them and restart the platform to apply.

Parameter	Description	Value
feature.parameterization. maxNumberOfFilesForExecution.	Maximum number of parameterized source files that are permitted to be executed as	

databricksSpark	part of an Azure Databricks job.	
feature.parameterization. matchLimitOnSampling.databricksSpark	Maximum number of parameterized source files that are permitted for matching in a single dataset with parameters.	
databricks.workerNodeType	Type of node to use for the Azure Databricks Workers/Executors. There are 1 or more Worker nodes per cluster.	Default: `Standard_D3_v2` **NOTE:** This property is unused when instance pooling is enabled. For more information, see Configure instance pooling below. For more information, see the sizing guide for Azure Databricks.
databricks.sparkVersion	Azure Databricks cluster version which also includes the Spark Version.	**NOTE:** Please verify that this value is set to the following: `5.5.x-scala2.11`. Please do not use other values. For more information, see *Configure for Spark*.
databricks.serviceUrl	URL to the Azure Databricks Service where Spark jobs will be run (Example: *https://westus2.azuredatabricks.net*)	
databricks.minWorkers	Initial number of Worker nodes in the cluster, and also the minimum number of Worker nodes that the cluster can scale down to during auto-scale-down	Minimum value: `1` Increasing this value can increase compute costs.
databricks.maxWorkers	Maximum number of Worker nodes the cluster can create during auto scaling	Minimum value: Not less than `databricks.minWorkers`. Increasing this value can increase compute costs.
databricks.poolId	If you have enabled instance pooling in Azure Databricks, you can specify the pool identifier here. For more information, see Configure instance pooling below.	
databricks.driverNodeType	Type of node to use for the Azure Databricks Driver. There is only 1 Driver node per cluster.	Default: `Standard_D3_v2` For more information, see the sizing guide for Databricks. **NOTE:** This property is unused when instance pooling is enabled. For more information, see Configure instance pooling below.
databricks.logsDestination	DBFS location that cluster logs will be sent to every 5 minutes	Leave this value as `/trifacta/logs`.
databricks.enableAutotermination	Set to true to enable auto-termination of a user cluster after N minutes of idle time, where N is the value of the autoterminationMinutes property.	Unless otherwise required, leave this value as `true`.
databricks. clusterStatePollerDelayInSeconds	Number of seconds to wait between polls for Azure Databricks cluster status when a cluster is starting up	

Copyright © 2020 Trifacta Inc.

databricks.clusterStartupWaitTimeInMinutes	Maximum time in minutes to wait for a Cluster to get to Running state before aborting and failing an Azure Databricks job	
databricks. clusterLogSyncWaitTimeInMinutes	Maximum time in minutes to wait for a Cluster to complete syncing its logs to DBFS before giving up on pulling the cluster logs to the Trifacta node.	Set this to `0` to disable cluster log pulls.
databricks. clusterLogSyncPollerDelayInSeconds	Number of seconds to wait between polls for a Databricks cluster to sync its logs to DBFS after job completion	
databricks.autoterminationMinutes	Idle time in minutes before a user cluster will auto-terminate.	Do not set this value to less than the cluster startup wait time value.
spark.useVendorSparkLibraries	When `true`, the platform bypasses shipping its installed Spark libraries to the cluster with each job's execution.	Default is `false`. Do not modify unless you are experiencing failures in Azure Databricks job execution. For more information, see Troubleshooting below.

Configure instance pooling

Instance pooling reduces cluster node spin-up time by maintaining a set of idle and ready instances. The Trifacta platform can be configured to leverage instance pooling on the Azure Databricks cluster.

Pre-requisites:

- All cluster nodes used by the Trifacta platform are taken from the pool. If the pool has an insufficient number of nodes, cluster creation fails.
- Each user must have access to the pool and must have at least the `ATTACH_TO` permission.
- Each user must have a personal access token from the same Azure Databricks workspace. See Configure personal access token below.

To enable:

1. Acquire your pool identifier from Azure Databricks.
2. You can apply this change through the *Admin Settings Page* (recommended) or `trifacta-conf.json`. For more information, see *Platform Configuration Methods*.
3. Set the following parameter to the Azure Databricks pool identifier:

```
"databricks.poolId": "<my_pool_id>",
```

4. Save your changes and restart the platform.

NOTE: When instance pooling is enabled, the following parameters are not used:

`databricks.driverNodeType`

`databricks.workerNodeType`

For more information, see *https://docs.azuredatabricks.net/clusters/instance-pools/index.html*.

Configure personal access token

Each user must insert a Databricks Personal Access Token to access Databricks resources. For more information, see *Databricks Personal Access Token Page*.

Copyright © 2020 Trifacta Inc.

Additional Configuration

Enable SSO for Azure Databricks

To enable SSO authentication with Azure Databricks, you enable SSO integration with Azure AD. For more information, see *Configure SSO for Azure AD*.

Enable Azure Managed Identity access

For enhanced security, you can configure the Trifacta platform to use an Azure Managed Identity. When this feature is enabled, the platform queries the Key Vault for the secret holding the applicationId and secret to the service principal that provides access to the Azure services.

> **NOTE:** This feature is supported for Azure Databricks only.

> **NOTE:** Your Azure Key Vault must already be configured, and the applicationId and secret must be available in the Key Vault. See *Configure for Azure*.

To enable, the following parameters for the Trifacta platform must be specified.

You can apply this change through the *Admin Settings Page* (recommended) or `trifacta-conf.json`. For more information, see *Platform Configuration Methods*.

Parameter	Description
azure.managedIdentities.enabled	Set to `true` to enable use of Azure managed identities.
azure.managedIdentities.keyVaultApplicationidSecretName	Specify the name of the Azure Key Vault secret that holds the service principal Application Id.
azure.managedIdentities.keyVaultApplicationSecretSecretName	Specify the name of the Key Vault secret that holds the service principal secret.

Save your changes.

Pass additional Spark properties

As needed, you can pass additional properties to the Spark running environment through the `spark.props` configuration area.

> **NOTE:** These properties are passed to Spark for all jobs.

Steps:

1. You can apply this change through the *Admin Settings Page* (recommended) or `trifacta-conf.json`. For more information, see *Platform Configuration Methods*.
2. Search for the following property: `spark.props`.
3. Insert new Spark properties. For example, you can specify the `spark.props.spark.executor.memory` property, which changes the memory allocated to the Spark executor on each node by using the following in the `spark.props` area:

Copyright © 2020 Trifacta Inc.

```
"spark": {
  ...
  "props": {
    "spark.executor.memory": "6GB"
  }
  ...
}
```

4. Save your changes and restart the platform.

For more information on modifying these settings, see *Configure for Spark*.

Use

Run job from application

When the above configuration has been completed, you can select the running environment through the application. See *Run Job Page*.

Run job via API

You can use API calls to execute jobs.

Please make sure that the request body contains the following:

```
"execution": "databricksSpark",
```

For more information, see *API JobGroups Create v4*.

Troubleshooting

Spark job on Azure Databricks fails with "Invalid spark version" error

When running a job using Spark on Azure Databricks, the job may fail with the above invalid version error. In this case, the Databricks version of Spark has been deprecated.

Solution:

Since an Azure Databricks cluster is created for each user, the solution is to identify the cluster version to use, configure the platform to use it, and then restart the platform.

1. You can apply this change through the *Admin Settings Page* (recommended) or `trifacta-conf.json`. For more information, see *Platform Configuration Methods*.
2. Acquire the value for `databricks.sparkVersion`.
3. In Azure Databricks, compare your value to the list of supported Azure Databricks version. If your version is unsupported, identify a new version to use.

   NOTE: Please make note of the version of Spark supported for the version of Azure Databricks that you have chosen.

4. In the Trifacta platform configuration:
   1. Set `databricks.sparkVersion` to the new version to use.
   2. Set `spark.version` to the appropriate version of Spark to use.

Copyright © 2020 Trifacta Inc.

5. Restart the Trifacta platform.
6. The platform is restarted. A new Azure Databricks cluster is created for each user using the specified values, when the user runs a job.

Contact Support

Do you need further assistance? Check out the resources below:

Search Support

In Trifacta® Wrangler Enterprise, click the Help icon and select **Search Help** to search our help content.

If your question is not answered through search, you can file a support ticket through the Support Portal (see below).

Trifacta Community and Support Portal

The Trifacta Community and Support Portal can be reached at:
https://community.trifacta.com

Within our Community, you can:

- Manage your support cases
- Get free Wrangler certifications
- Post questions to the community
- Search our AI-driven knowledgebase
- Answer questions to earn points and get on the leaderboard
- Watch tutorials, access documentation, learn about new features, and more

Email

support@trifacta.com

Legal

Third-Party License Information

Copyright © 2020 Trifacta Inc.

This product also includes the following libraries which are covered by The (MIT AND BSD-3-Clause):

- sha.js

This product also includes the following libraries which are covered by The 2-clause BSD License:

- double_metaphone

This product also includes the following libraries which are covered by The 3-Clause BSD License:

- com.google.protobuf.protobuf-java
- com.google.protobuf.protobuf-java-util

This product also includes the following libraries which are covered by The ASL:

- funcsigs

- org.json4s.json4s-ast_2.10
- org.json4s.json4s-core_2.10
- org.json4s.json4s-native_2.10

This product also includes the following libraries which are covered by The ASL 2.0:

- pykerberos

This product also includes the following libraries which are covered by The Amazon Redshift ODBC and JDBC Driver License Agreement:

- RedshiftJDBC41-1.1.7.1007
- com.amazon.redshift.RedshiftJDBC41

This product also includes the following libraries which are covered by The Apache 2.0 License:

- com.uber.jaeger.jaeger-b3
- com.uber.jaeger.jaeger-core
- com.uber.jaeger.jaeger-thrift
- com.uber.jaeger.jaeger-zipkin
- org.apache.spark.spark-catalyst_2.11
- org.apache.spark.spark-core_2.11
- org.apache.spark.spark-hive_2.11
- org.apache.spark.spark-kvstore_2.11
- org.apache.spark.spark-launcher_2.11
- org.apache.spark.spark-network-common_2.11
- org.apache.spark.spark-network-shuffle_2.11
- org.apache.spark.spark-sketch_2.11
- org.apache.spark.spark-sql_2.11
- org.apache.spark.spark-tags_2.11
- org.apache.spark.spark-unsafe_2.11
- org.apache.spark.spark-yarn_2.11

This product also includes the following libraries which are covered by The Apache License:

- com.chuusai.shapeless_2.11
- commons-httpclient.commons-httpclient
- org.apache.httpcomponents.httpclient
- org.apache.httpcomponents.httpcore

This product also includes the following libraries which are covered by The Apache License (v2.0):

- com.vlkan.flatbuffers

This product also includes the following libraries which are covered by The Apache License 2.0:

- @google-cloud/pubsub
- @google-cloud/resource
- @google-cloud/storage
- arrow
- avro
- aws-sdk
- azure-storage
- bootstrap
- browser-request
- bytebuffer
- cglib.cglib-nodep
- com.amazonaws.aws-java-sdk
- com.amazonaws.aws-java-sdk-bundle
- com.amazonaws.aws-java-sdk-core

- com.amazonaws.aws-java-sdk-dynamodb
- com.amazonaws.aws-java-sdk-emr
- com.amazonaws.aws-java-sdk-iam
- com.amazonaws.aws-java-sdk-kms
- com.amazonaws.aws-java-sdk-s3
- com.amazonaws.aws-java-sdk-sts
- com.amazonaws.jmespath-java
- com.carrotsearch.hppc
- com.clearspring.analytics.stream
- com.cloudera.navigator.navigator-sdk
- com.cloudera.navigator.navigator-sdk-client
- com.cloudera.navigator.navigator-sdk-model
- com.codahale.metrics.metrics-core
- com.cronutils.cron-utils
- com.databricks.spark-avro_2.11
- com.fasterxml.classmate
- com.fasterxml.jackson.dataformat.jackson-dataformat-cbor
- com.fasterxml.jackson.datatype.jackson-datatype-joda
- com.fasterxml.jackson.jaxrs.jackson-jaxrs-base
- com.fasterxml.jackson.jaxrs.jackson-jaxrs-json-provider
- com.fasterxml.jackson.module.jackson-module-jaxb-annotations
- com.fasterxml.jackson.module.jackson-module-paranamer
- com.fasterxml.jackson.module.jackson-module-scala_2.11
- com.fasterxml.uuid.java-uuid-generator
- com.github.nscala-time.nscala-time_2.10
- com.github.stephenc.jcip.jcip-annotations
- com.google.api-client.google-api-client
- com.google.api-client.google-api-client-jackson2
- com.google.api-client.google-api-client-java6
- com.google.api.grpc.grpc-google-cloud-bigquerystorage-v1beta1
- com.google.api.grpc.grpc-google-cloud-bigtable-admin-v2
- com.google.api.grpc.grpc-google-cloud-bigtable-v2
- com.google.api.grpc.grpc-google-cloud-pubsub-v1
- com.google.api.grpc.grpc-google-cloud-spanner-admin-database-v1
- com.google.api.grpc.grpc-google-cloud-spanner-admin-instance-v1
- com.google.api.grpc.grpc-google-cloud-spanner-v1
- com.google.api.grpc.grpc-google-common-protos
- com.google.api.grpc.proto-google-cloud-bigquerystorage-v1beta1
- com.google.api.grpc.proto-google-cloud-bigtable-admin-v2
- com.google.api.grpc.proto-google-cloud-bigtable-v2
- com.google.api.grpc.proto-google-cloud-datastore-v1
- com.google.api.grpc.proto-google-cloud-monitoring-v3
- com.google.api.grpc.proto-google-cloud-pubsub-v1
- com.google.api.grpc.proto-google-cloud-spanner-admin-database-v1
- com.google.api.grpc.proto-google-cloud-spanner-admin-instance-v1
- com.google.api.grpc.proto-google-cloud-spanner-v1
- com.google.api.grpc.proto-google-common-protos
- com.google.api.grpc.proto-google-iam-v1
- com.google.apis.google-api-services-bigquery
- com.google.apis.google-api-services-clouddebugger
- com.google.apis.google-api-services-cloudresourcemanager
- com.google.apis.google-api-services-dataflow
- com.google.apis.google-api-services-iam
- com.google.apis.google-api-services-oauth2
- com.google.apis.google-api-services-pubsub
- com.google.apis.google-api-services-storage
- com.google.auto.service.auto-service
- com.google.auto.value.auto-value-annotations
- com.google.cloud.bigdataoss.gcsio
- com.google.cloud.bigdataoss.util

Copyright © 2020 Trifacta Inc.

- com.google.cloud.google-cloud-bigquery
- com.google.cloud.google-cloud-bigquerystorage
- com.google.cloud.google-cloud-bigtable
- com.google.cloud.google-cloud-bigtable-admin
- com.google.cloud.google-cloud-core
- com.google.cloud.google-cloud-core-grpc
- com.google.cloud.google-cloud-core-http
- com.google.cloud.google-cloud-monitoring
- com.google.cloud.google-cloud-spanner
- com.google.code.findbugs.jsr305
- com.google.code.gson.gson
- com.google.errorprone.error_prone_annotations
- com.google.flogger.flogger
- com.google.flogger.flogger-system-backend
- com.google.flogger.google-extensions
- com.google.guava.failureaccess
- com.google.guava.guava
- com.google.guava.guava-jdk5
- com.google.guava.listenablefuture
- com.google.http-client.google-http-client
- com.google.http-client.google-http-client-apache
- com.google.http-client.google-http-client-appengine
- com.google.http-client.google-http-client-jackson
- com.google.http-client.google-http-client-jackson2
- com.google.http-client.google-http-client-protobuf
- com.google.inject.extensions.guice-servlet
- com.google.inject.guice
- com.google.j2objc.j2objc-annotations
- com.google.oauth-client.google-oauth-client
- com.google.oauth-client.google-oauth-client-java6
- com.googlecode.javaewah.JavaEWAH
- com.googlecode.libphonenumber.libphonenumber
- com.hadoop.gplcompression.hadoop-lzo
- com.jakewharton.threetenabp.threetenabp
- com.jamesmurty.utils.java-xmlbuilder
- com.jolbox.bonecp
- com.mapr.mapr-root
- com.microsoft.azure.adal4j
- com.microsoft.azure.azure-core
- com.microsoft.azure.azure-storage
- com.microsoft.windowsazure.storage.microsoft-windowsazure-storage-sdk
- com.nimbusds.lang-tag
- com.nimbusds.nimbus-jose-jwt
- com.ning.compress-lzf
- com.opencsv.opencsv
- com.squareup.okhttp.okhttp
- com.squareup.okhttp3.logging-interceptor
- com.squareup.okhttp3.okhttp
- com.squareup.okhttp3.okhttp-urlconnection
- com.squareup.okio.okio
- com.squareup.retrofit2.adapter-rxjava
- com.squareup.retrofit2.converter-jackson
- com.squareup.retrofit2.retrofit
- com.trifacta.hadoop.cloudera4
- com.twitter.chill-java
- com.twitter.chill_2.11
- com.twitter.parquet-hadoop-bundle
- com.typesafe.akka.akka-actor_2.11
- com.typesafe.akka.akka-cluster_2.11
- com.typesafe.akka.akka-remote_2.11

Copyright © 2020 Trifacta Inc.

- com.typesafe.akka.akka-slf4j_2.11
- com.typesafe.config
- com.univocity.univocity-parsers
- com.zaxxer.HikariCP
- com.zaxxer.HikariCP-java7
- commons-beanutils.commons-beanutils
- commons-beanutils.commons-beanutils-core
- commons-cli.commons-cli
- commons-codec.commons-codec
- commons-collections.commons-collections
- commons-configuration.commons-configuration
- commons-dbcp.commons-dbcp
- commons-dbutils.commons-dbutils
- commons-digester.commons-digester
- commons-el.commons-el
- commons-fileupload.commons-fileupload
- commons-io.commons-io
- commons-lang.commons-lang
- commons-logging.commons-logging
- commons-net.commons-net
- commons-pool.commons-pool
- dateinfer
- de.odysseus.juel.juel-api
- de.odysseus.juel.juel-impl
- de.odysseus.juel.juel-spi
- express-opentracing
- google-benchmark
- googleapis
- io.airlift.aircompressor
- io.dropwizard.metrics.metrics-core
- io.dropwizard.metrics.metrics-graphite
- io.dropwizard.metrics.metrics-json
- io.dropwizard.metrics.metrics-jvm
- io.grpc.grpc-all
- io.grpc.grpc-alts
- io.grpc.grpc-auth
- io.grpc.grpc-context
- io.grpc.grpc-core
- io.grpc.grpc-grpclb
- io.grpc.grpc-netty
- io.grpc.grpc-netty-shaded
- io.grpc.grpc-okhttp
- io.grpc.grpc-protobuf
- io.grpc.grpc-protobuf-lite
- io.grpc.grpc-protobuf-nano
- io.grpc.grpc-stub
- io.grpc.grpc-testing
- io.netty.netty
- io.netty.netty-all
- io.netty.netty-buffer
- io.netty.netty-codec
- io.netty.netty-codec-http
- io.netty.netty-codec-http2
- io.netty.netty-codec-socks
- io.netty.netty-common
- io.netty.netty-handler
- io.netty.netty-handler-proxy
- io.netty.netty-resolver
- io.netty.netty-tcnative-boringssl-static
- io.netty.netty-transport

Copyright © 2020 Trifacta Inc.

- io.opentracing.contrib.opentracing-concurrent
- io.opentracing.contrib.opentracing-globaltracer
- io.opentracing.contrib.opentracing-web-servlet-filter
- io.opentracing.opentracing-api
- io.opentracing.opentracing-noop
- io.opentracing.opentracing-util
- io.prometheus.simpleclient
- io.prometheus.simpleclient_common
- io.prometheus.simpleclient_servlet
- io.reactivex.rxjava
- io.spray.spray-can_2.11
- io.spray.spray-http_2.11
- io.spray.spray-httpx_2.11
- io.spray.spray-io_2.11
- io.spray.spray-json_2.11
- io.spray.spray-routing_2.11
- io.spray.spray-util_2.11
- io.springfox.springfox-core
- io.springfox.springfox-schema
- io.springfox.springfox-spi
- io.springfox.springfox-spring-web
- io.springfox.springfox-swagger-common
- io.springfox.springfox-swagger-ui
- io.springfox.springfox-swagger2
- io.swagger.swagger-annotations
- io.swagger.swagger-models
- io.undertow.undertow-core
- io.undertow.undertow-servlet
- io.undertow.undertow-websockets-jsr
- io.zipkin.java.zipkin
- io.zipkin.reporter.zipkin-reporter
- io.zipkin.reporter.zipkin-sender-urlconnection
- io.zipkin.zipkin2.zipkin
- it.unimi.dsi.fastutil
- jaeger-client
- javax.inject.javax.inject
- javax.jdo.jdo-api
- javax.validation.validation-api
- joda-time.joda-time
- kerberos
- less
- libcuckoo
- log4j.apache-log4j-extras
- log4j.log4j
- long
- mathjs
- mx4j.mx4j
- net.bytebuddy.byte-buddy
- net.hydromatic.eigenbase-properties
- net.java.dev.eval.eval
- net.java.dev.jets3t.jets3t
- net.jcip.jcip-annotations
- net.jpountz.lz4.lz4
- net.minidev.accessors-smart
- net.minidev.json-smart
- net.sf.opencsv.opencsv
- net.snowflake.snowflake-jdbc
- opentracing
- org.activiti.activiti-bpmn-converter
- org.activiti.activiti-bpmn-layout

Copyright © 2020 Trifacta Inc.

- org.activiti.activiti-bpmn-model
- org.activiti.activiti-common-rest
- org.activiti.activiti-dmn-api
- org.activiti.activiti-dmn-model
- org.activiti.activiti-engine
- org.activiti.activiti-form-api
- org.activiti.activiti-form-model
- org.activiti.activiti-image-generator
- org.activiti.activiti-process-validation
- org.activiti.activiti-rest
- org.activiti.activiti-spring
- org.activiti.activiti-spring-boot-starter-basic
- org.activiti.activiti-spring-boot-starter-rest-api
- org.activiti.activiti5-compatibility
- org.activiti.activiti5-engine
- org.activiti.activiti5-spring
- org.activiti.activiti5-spring-compatibility
- org.apache.arrow.arrow-format
- org.apache.arrow.arrow-memory
- org.apache.arrow.arrow-vector
- org.apache.atlas.atlas-client
- org.apache.atlas.atlas-typesystem
- org.apache.avro.avro
- org.apache.avro.avro-ipc
- org.apache.avro.avro-mapred
- org.apache.beam.beam-model-job-management
- org.apache.beam.beam-model-pipeline
- org.apache.beam.beam-runners-core-construction-java
- org.apache.beam.beam-runners-direct-java
- org.apache.beam.beam-runners-google-cloud-dataflow-java
- org.apache.beam.beam-sdks-java-core
- org.apache.beam.beam-sdks-java-extensions-google-cloud-platform-core
- org.apache.beam.beam-sdks-java-extensions-protobuf
- org.apache.beam.beam-sdks-java-extensions-sorter
- org.apache.beam.beam-sdks-java-io-google-cloud-platform
- org.apache.beam.beam-sdks-java-io-parquet
- org.apache.beam.beam-vendor-grpc-1_13_1
- org.apache.beam.beam-vendor-guava-20_0
- org.apache.calcite.calcite-avatica
- org.apache.calcite.calcite-core
- org.apache.calcite.calcite-linq4j
- org.apache.commons.codec
- org.apache.commons.commons-collections4
- org.apache.commons.commons-compress
- org.apache.commons.commons-configuration2
- org.apache.commons.commons-crypto
- org.apache.commons.commons-csv
- org.apache.commons.commons-dbcp2
- org.apache.commons.commons-email
- org.apache.commons.commons-exec
- org.apache.commons.commons-lang3
- org.apache.commons.commons-math
- org.apache.commons.commons-math3
- org.apache.commons.commons-pool2
- org.apache.curator.curator-client
- org.apache.curator.curator-framework
- org.apache.curator.curator-recipes
- org.apache.derby.derby
- org.apache.directory.api.api-asn1-api
- org.apache.directory.api.api-util

Copyright © 2020 Trifacta Inc.

- org.apache.directory.server.apacheds-i18n
- org.apache.directory.server.apacheds-kerberos-codec
- org.apache.hadoop.avro
- org.apache.hadoop.hadoop-annotations
- org.apache.hadoop.hadoop-auth
- org.apache.hadoop.hadoop-aws
- org.apache.hadoop.hadoop-azure
- org.apache.hadoop.hadoop-azure-datalake
- org.apache.hadoop.hadoop-client
- org.apache.hadoop.hadoop-common
- org.apache.hadoop.hadoop-hdfs
- org.apache.hadoop.hadoop-hdfs-client
- org.apache.hadoop.hadoop-mapreduce-client-app
- org.apache.hadoop.hadoop-mapreduce-client-common
- org.apache.hadoop.hadoop-mapreduce-client-core
- org.apache.hadoop.hadoop-mapreduce-client-jobclient
- org.apache.hadoop.hadoop-mapreduce-client-shuffle
- org.apache.hadoop.hadoop-yarn-api
- org.apache.hadoop.hadoop-yarn-client
- org.apache.hadoop.hadoop-yarn-common
- org.apache.hadoop.hadoop-yarn-registry
- org.apache.hadoop.hadoop-yarn-server-common
- org.apache.hadoop.hadoop-yarn-server-nodemanager
- org.apache.hadoop.hadoop-yarn-server-web-proxy
- org.apache.htrace.htrace-core
- org.apache.htrace.htrace-core4
- org.apache.httpcomponents.httpmime
- org.apache.ivy.ivy
- org.apache.kerby.kerb-admin
- org.apache.kerby.kerb-client
- org.apache.kerby.kerb-common
- org.apache.kerby.kerb-core
- org.apache.kerby.kerb-crypto
- org.apache.kerby.kerb-identity
- org.apache.kerby.kerb-server
- org.apache.kerby.kerb-simplekdc
- org.apache.kerby.kerb-util
- org.apache.kerby.kerby-asn1
- org.apache.kerby.kerby-config
- org.apache.kerby.kerby-pkix
- org.apache.kerby.kerby-util
- org.apache.kerby.kerby-xdr
- org.apache.kerby.token-provider
- org.apache.logging.log4j.log4j-1.2-api
- org.apache.logging.log4j.log4j-api
- org.apache.logging.log4j.log4j-api-scala_2.11
- org.apache.logging.log4j.log4j-core
- org.apache.logging.log4j.log4j-jcl
- org.apache.logging.log4j.log4j-jul
- org.apache.logging.log4j.log4j-slf4j-impl
- org.apache.logging.log4j.log4j-web
- org.apache.orc.orc-core
- org.apache.orc.orc-mapreduce
- org.apache.orc.orc-shims
- org.apache.parquet.parquet-avro
- org.apache.parquet.parquet-column
- org.apache.parquet.parquet-common
- org.apache.parquet.parquet-encoding
- org.apache.parquet.parquet-format
- org.apache.parquet.parquet-hadoop

Copyright © 2020 Trifacta Inc.

- org.apache.parquet.parquet-jackson
- org.apache.parquet.parquet-tools
- org.apache.pig.pig
- org.apache.pig.pig-core-spork
- org.apache.thrift.libfb303
- org.apache.thrift.libthrift
- org.apache.tomcat.embed.tomcat-embed-core
- org.apache.tomcat.embed.tomcat-embed-el
- org.apache.tomcat.embed.tomcat-embed-websocket
- org.apache.tomcat.tomcat-annotations-api
- org.apache.tomcat.tomcat-jdbc
- org.apache.tomcat.tomcat-juli
- org.apache.xbean.xbean-asm5-shaded
- org.apache.xbean.xbean-asm6-shaded
- org.apache.zookeeper.zookeeper
- org.codehaus.jackson.jackson-core-asl
- org.codehaus.jackson.jackson-mapper-asl
- org.codehaus.jettison.jettison
- org.datanucleus.datanucleus-api-jdo
- org.datanucleus.datanucleus-core
- org.datanucleus.datanucleus-rdbms
- org.eclipse.jetty.jetty-client
- org.eclipse.jetty.jetty-http
- org.eclipse.jetty.jetty-io
- org.eclipse.jetty.jetty-security
- org.eclipse.jetty.jetty-server
- org.eclipse.jetty.jetty-servlet
- org.eclipse.jetty.jetty-util
- org.eclipse.jetty.jetty-util-ajax
- org.eclipse.jetty.jetty-webapp
- org.eclipse.jetty.jetty-xml
- org.fusesource.jansi.jansi
- org.hibernate.hibernate-validator
- org.htrace.htrace-core
- org.iq80.snappy.snappy
- org.jboss.jandex
- org.joda.joda-convert
- org.json4s.json4s-ast_2.11
- org.json4s.json4s-core_2.11
- org.json4s.json4s-jackson_2.11
- org.json4s.json4s-scalap_2.11
- org.liquibase.liquibase-core
- org.lz4.lz4-java
- org.mapstruct.mapstruct
- org.mortbay.jetty.jetty
- org.mortbay.jetty.jetty-util
- org.mybatis.mybatis
- org.objenesis.objenesis
- *org.osgi.org*.osgi.core
- org.parboiled.parboiled-core
- org.parboiled.parboiled-scala_2.11
- org.quartz-scheduler.quartz
- org.roaringbitmap.RoaringBitmap
- org.sonatype.oss.oss-parent
- org.sonatype.sisu.inject.cglib
- org.spark-project.hive.hive-exec
- org.spark-project.hive.hive-metastore
- org.springframework.boot.spring-boot
- org.springframework.boot.spring-boot-autoconfigure
- org.springframework.boot.spring-boot-starter

Copyright © 2020 Trifacta Inc.

- org.springframework.boot.spring-boot-starter-aop
- org.springframework.boot.spring-boot-starter-data-jpa
- org.springframework.boot.spring-boot-starter-jdbc
- org.springframework.boot.spring-boot-starter-log4j2
- org.springframework.boot.spring-boot-starter-logging
- org.springframework.boot.spring-boot-starter-tomcat
- org.springframework.boot.spring-boot-starter-undertow
- org.springframework.boot.spring-boot-starter-web
- org.springframework.data.spring-data-commons
- org.springframework.data.spring-data-jpa
- org.springframework.plugin.spring-plugin-core
- org.springframework.plugin.spring-plugin-metadata
- org.springframework.retry.spring-retry
- org.springframework.security.spring-security-config
- org.springframework.security.spring-security-core
- org.springframework.security.spring-security-crypto
- org.springframework.security.spring-security-web
- org.springframework.spring-aop
- org.springframework.spring-aspects
- org.springframework.spring-beans
- org.springframework.spring-context
- org.springframework.spring-context-support
- org.springframework.spring-core
- org.springframework.spring-expression
- org.springframework.spring-jdbc
- org.springframework.spring-orm
- org.springframework.spring-tx
- org.springframework.spring-web
- org.springframework.spring-webmvc
- org.springframework.spring-websocket
- org.uncommons.maths.uncommons-maths
- org.wildfly.openssl.wildfly-openssl
- org.xerial.snappy.snappy-java
- org.yaml.snakeyaml
- oro.oro
- parquet
- pbr
- pig-0.11.1-withouthadoop-23
- pig-0.12.1-mapr-noversion-withouthadoop
- pig-0.14.0-core-spork
- piggybank-amzn-0.3
- piggybank-cdh5.0.0-beta-2-0.12.0
- python-iptables
- request
- requests
- stax.stax-api
- thrift
- tomcat.jasper-compiler
- tomcat.jasper-runtime
- xerces.xercesImpl
- xml-apis.xml-apis
- zipkin
- zipkin-transport-http

This product also includes the following libraries which are covered by The Apache License 2.0 + Eclipse Public License 1.0:

- spark-assembly-thinner

This product also includes the following libraries which are covered by The Apache License Version 2:

- org.mortbay.jetty.jetty-sslengine

This product also includes the following libraries which are covered by The Apache License v2.0:

- net.java.dev.jna.jna
- net.java.dev.jna.jna-platform

This product also includes the following libraries which are covered by The Apache License, version 2.0:

- com.nimbusds.oauth2-oidc-sdk
- org.jboss.logging.jboss-logging

This product also includes the following libraries which are covered by The Apache Software Licenses:

- org.slf4j.log4j-over-slf4j

This product also includes the following libraries which are covered by The BSD license:

- alabaster
- antlr.antlr
- asm.asm-parent
- babel
- click
- com.google.api.api-common
- com.google.api.gax
- com.google.api.gax-grpc
- com.google.api.gax-httpjson
- com.jcraft.jsch
- com.thoughtworks.paranamer.paranamer
- dk.brics.automaton.automaton
- dom4j.dom4j
- enum34
- flask
- itsdangerous
- javolution.javolution
- jinja2
- jline.jline
- microee
- mock
- networkx
- numpy
- org.antlr.ST4
- org.antlr.antlr-runtime
- org.antlr.antlr4-runtime
- org.antlr.stringtemplate
- org.codehaus.woodstox.stax2-api
- org.ow2.asm.asm
- org.scala-lang.jline
- pandas
- pluginbase
- psutil
- pygments
- python-enum34
- python-json-logger
- scipy
- snowballstemmer
- sphinx
- sphinxcontrib-websupport
- strptime

Copyright © 2020 Trifacta Inc.

- websocket-stream
- xlrd
- xmlenc.xmlenc
- xss-filters

This product also includes the following libraries which are covered by The BSD 2-Clause License:

- com.github.luben.zstd-jni

This product also includes the following libraries which are covered by The BSD 3-Clause:

- org.scala-lang.scala-compiler
- org.scala-lang.scala-library
- org.scala-lang.scala-reflect
- org.scala-lang.scalap

This product also includes the following libraries which are covered by The BSD 3-Clause "New" or "Revised" License (BSD-3-Clause):

- *org.abego.treelayout.org*.abego.treelayout.core

This product also includes the following libraries which are covered by The BSD 3-Clause License:

- org.antlr.antlr4

This product also includes the following libraries which are covered by The BSD 3-clause:

- org.scala-lang.modules.scala-parser-combinators_2.11
- org.scala-lang.modules.scala-xml_2.11
- org.scala-lang.plugins.scala-continuations-library_2.11
- org.threeten.threetenbp

This product also includes the following libraries which are covered by The BSD New license:

- com.google.auth.google-auth-library-credentials
- com.google.auth.google-auth-library-oauth2-http

This product also includes the following libraries which are covered by The BSD-2-Clause:

- cls-bluebird
- node-polyglot
- org.postgresql.postgresql
- terser
- uglify-js

This product also includes the following libraries which are covered by The BSD-3-Clause:

- @sentry/node
- d3-dsv
- datalib
- datalib-sketch
- markupsafe
- md5
- node-forge
- protobufjs
- qs
- queue-async
- sqlite3
- werkzeug

Copyright © 2020 Trifacta Inc.

This product also includes the following libraries which are covered by The BSD-Style:

- com.jsuereth.scala-arm_2.11

This product also includes the following libraries which are covered by The BSD-derived (*http://www.repoze.org/LICENSE.txt*):

- meld3
- supervisor

This product also includes the following libraries which are covered by The BSD-like:

- dnspython
- idna
- org.scala-lang.scala-actors
- org.scalamacros.quasiquotes_2.10

This product also includes the following libraries which are covered by The BSD-style license:

- bzip2

This product also includes the following libraries which are covered by The Boost Software License:

- asio
- boost
- cpp-netlib-uri
- expected
- jsbind
- poco

This product also includes the following libraries which are covered by The Bouncy Castle Licence:

- org.bouncycastle.bcprov-jdk15on

This product also includes the following libraries which are covered by The CDDL:

- javax.mail.mail
- javax.mail.mailapi
- javax.servlet.jsp-api
- javax.servlet.jsp.jsp-api
- javax.servlet.servlet-api
- javax.transaction.jta
- javax.xml.stream.stax-api
- org.glassfish.external.management-api
- org.glassfish.gmbal.gmbal-api-only
- org.jboss.spec.javax.annotation.jboss-annotations-api_1.2_spec

This product also includes the following libraries which are covered by The CDDL + GPLv2 with classpath exception:

- javax.annotation.javax.annotation-api
- javax.jms.jms
- javax.servlet.javax.servlet-api
- javax.transaction.javax.transaction-api
- javax.transaction.transaction-api
- org.glassfish.grizzly.grizzly-framework
- org.glassfish.grizzly.grizzly-http
- org.glassfish.grizzly.grizzly-http-server
- org.glassfish.grizzly.grizzly-http-servlet
- org.glassfish.grizzly.grizzly-rcm

Copyright © 2020 Trifacta Inc.

- org.glassfish.hk2.external.aopalliance-repackaged
- org.glassfish.hk2.external.javax.inject
- org.glassfish.hk2.hk2-api
- org.glassfish.hk2.hk2-locator
- org.glassfish.hk2.hk2-utils
- org.glassfish.hk2.osgi-resource-locator
- org.glassfish.javax.el
- org.glassfish.jersey.core.jersey-common

This product also includes the following libraries which are covered by The CDDL 1.1:

- com.sun.jersey.contribs.jersey-guice
- com.sun.jersey.jersey-client
- com.sun.jersey.jersey-core
- com.sun.jersey.jersey-json
- com.sun.jersey.jersey-server
- com.sun.jersey.jersey-servlet
- com.sun.xml.bind.jaxb-impl
- *javax.ws.rs.javax.ws*.rs-api
- javax.xml.bind.jaxb-api
- org.jvnet.mimepull.mimepull

This product also includes the following libraries which are covered by The CDDL License:

- *javax.ws*.rs.jsr311-api

This product also includes the following libraries which are covered by The CDDL/GPLv2+CE:

- com.sun.mail.javax.mail

This product also includes the following libraries which are covered by The CERN:

- colt.colt

This product also includes the following libraries which are covered by The COMMON DEVELOPMENT AND DISTRIBUTION LICENSE (CDDL) Version 1.0:

- javax.activation.activation
- javax.annotation.jsr250-api

This product also includes the following libraries which are covered by The Doug Crockford's license that allows this module to be used for Good but not for Evil:

- jsmin

This product also includes the following libraries which are covered by The Dual License:

- python-dateutil

This product also includes the following libraries which are covered by The Eclipse Distribution License (EDL), Version 1.0:

- org.hibernate.javax.persistence.hibernate-jpa-2.1-api

This product also includes the following libraries which are covered by The Eclipse Public License:

- com.github.oshi.oshi-core

This product also includes the following libraries which are covered by The Eclipse Public License - v 1.0:

- org.aspectj.aspectjweaver

This product also includes the following libraries which are covered by The Eclipse Public License, Version 1.0:

- com.mchange.mchange-commons-java

This product also includes the following libraries which are covered by The GNU General Public License v2.0 only, with Classpath exception:

- org.jboss.spec.javax.servlet.jboss-servlet-api_3.1_spec
- org.jboss.spec.javax.websocket.jboss-websocket-api_1.1_spec

This product also includes the following libraries which are covered by The GNU LGPL:

- nose

This product also includes the following libraries which are covered by The GNU Lesser General Public License:

- org.hibernate.common.hibernate-commons-annotations
- org.hibernate.hibernate-core
- org.hibernate.hibernate-entitymanager

This product also includes the following libraries which are covered by The GNU Lesser General Public License Version 2.1, February 1999:

- org.jgrapht.jgrapht-core

This product also includes the following libraries which are covered by The GNU Lesser General Public License, Version 2.1:

- com.fasterxml.jackson.core.jackson-annotations
- com.fasterxml.jackson.core.jackson-core
- com.fasterxml.jackson.core.jackson-databind
- com.mchange.c3p0

This product also includes the following libraries which are covered by The GNU Lesser Public License:

- com.google.code.findbugs.annotations

This product also includes the following libraries which are covered by The GPLv3:

- yamllint

This product also includes the following libraries which are covered by The Google Cloud Software License:

- com.google.cloud.google-cloud-storage

This product also includes the following libraries which are covered by The ICU License:

- icu

This product also includes the following libraries which are covered by The ISC:

- browserify-sign
- iconify
- inherits
- lru-cache
- request-promise
- request-promise-native
- requests-kerberos

- rimraf
- sax
- semver
- split-ca

This product also includes the following libraries which are covered by The JGraph Ltd - 3 clause BSD license:

- org.tinyjee.jgraphx.jgraphx

This product also includes the following libraries which are covered by The LGPL:

- chardet
- com.sun.jna.jna

This product also includes the following libraries which are covered by The LGPL 2.1:

- org.codehaus.jackson.jackson-jaxrs
- org.codehaus.jackson.jackson-xc
- org.javassist.javassist
- xmlhttprequest

This product also includes the following libraries which are covered by The LGPLv3 or later:

- com.github.fge.json-schema-core
- com.github.fge.json-schema-validator

This product also includes the following libraries which are covered by The MIT license:

- amplitude-js
- analytics-node
- args4j.args4j
- async
- avsc
- backbone
- backbone-forms
- basic-auth
- bcrypt
- bluebird
- body-parser
- browser-filesaver
- buffer-crc32
- bufferedstream
- busboy
- byline
- bytes
- cachetools
- chai
- chalk
- cli-table
- clipboard
- codemirror
- colors
- com.github.tommyettinger.blazingchain
- com.microsoft.azure.azure-data-lake-store-sdk
- com.microsoft.sqlserver.mssql-jdbc
- commander
- common-tags
- compression
- console.table
- cookie

Copyright © 2020 Trifacta Inc.

- cookie-parser
- cookie-session
- cronstrue
- crypto-browserify
- csrf
- csurf
- definitely
- eventEmitter
- express
- express-http-context
- express-params
- express-zip
- forever
- form-data
- fs-extra
- function-rate-limit
- future
- fuzzy
- generic-pool
- google-auto-auth
- iconv-lite
- imagesize
- int24
- is-my-json-valid
- jade
- jq
- jquery
- jquery-serializeobject
- *jquery.ba*-serializeobject
- jquery.event.drag
- jquery.form
- jquery.ui
- json-stable-stringify
- jsonfile
- jsonschema
- jsonwebtoken
- jszip
- keygrip
- keysim
- knex
- less-middleware
- lodash
- lunr
- matic
- memcached
- method-override
- mini-css-extract-plugin
- minilog
- mkdirp
- moment
- moment-jdateformatparser
- moment-timezone
- mongoose
- morgan
- morgan-json
- mysql
- net.razorvine.pyrolite
- netifaces
- nock
- nodemailer

Copyright © 2020 Trifacta Inc.

- org.checkerframework.checker-compat-qual
- org.checkerframework.checker-qual
- org.mockito.mockito-core
- org.slf4j.jcl-over-slf4j
- org.slf4j.jul-to-slf4j
- org.slf4j.slf4j-api
- org.slf4j.slf4j-log4j12
- pace
- passport
- passport-azure-ad
- passport-http
- passport-http-bearer
- passport-ldapauth
- passport-local
- passport-saml
- passport-saml-metadata
- passport-strategy
- password-validator
- pegjs
- pg
- pg-hstore
- pip
- png-img
- promise-retry
- prop-types
- punycode
- py-cpuinfo
- python-crfsuite
- python-six
- pytz
- pyyaml
- query-string
- querystring
- randexp
- rapidjson
- react
- react-day-picker
- react-dom
- react-hot-loader
- react-modal
- react-router-dom
- react-select
- react-switch
- react-table
- react-virtualized
- recursive-readdir
- redefine
- requestretry
- require-jade
- require-json
- retry
- retry-as-promised
- rotating-file-stream
- safe-json-stringify
- sequelize
- setuptools
- simple-ldap-search
- simplejson
- singledispatch
- six

Copyright © 2020 Trifacta Inc.

- slick.core
- slick.grid
- slick.headerbuttons
- slick.headerbuttons.css
- slick.rowselectionmodel
- snappy
- sphinx-rtd-theme
- split-pane
- sql
- stream-meter
- supertest
- tar-fs
- temp
- to-case
- tv4
- ua-parser-js
- umzug
- underscore.string
- unicode-length
- universal-analytics
- urijs
- uritools
- url
- urllib3
- user-agent-parser
- uuid
- validator
- wheel
- winston
- winston-daily-rotate-file
- ws
- yargs
- zxcvbn

This product also includes the following libraries which are covered by The MIT License; BSD 3-clause License:

- antlr4-cpp-runtime

This product also includes the following libraries which are covered by The MIT license:

- org.codehaus.mojo.animal-sniffer-annotations

This product also includes the following libraries which are covered by The MIT/X:

- vnc2flv

This product also includes the following libraries which are covered by The MIT/X11 license:

- optimist

This product also includes the following libraries which are covered by The MPL 2.0:

- pathspec

This product also includes the following libraries which are covered by The MPL 2.0 or EPL 1.0:

- com.h2database.h2

This product also includes the following libraries which are covered by The MPL-2.0:

- certifi

This product also includes the following libraries which are covered by The Microsoft Software License:

- com.microsoft.sqlserver.sqljdbc4

This product also includes the following libraries which are covered by The Mozilla Public License, Version 2.0:

- org.mozilla.rhino

This product also includes the following libraries which are covered by The New BSD License:

- asm.asm
- backbone-queryparams
- bloomfilter
- com.esotericsoftware.kryo-shaded
- com.esotericsoftware.minlog
- com.googlecode.protobuf-java-format.protobuf-java-format
- d3
- domReady
- double-conversion
- gflags
- glog
- gperftools
- gtest
- org.antlr.antlr-master
- org.codehaus.janino.commons-compiler
- org.codehaus.janino.janino
- org.hamcrest.hamcrest-core
- pcre
- protobuf
- re2
- require-text
- sha1
- termcolor
- topojson
- triflow
- vega
- websocketpp

This product also includes the following libraries which are covered by The New BSD license:

- com.google.protobuf.nano.protobuf-javanano

This product also includes the following libraries which are covered by The Oracle Technology Network License:

- com.oracle.ojdbc6

This product also includes the following libraries which are covered by The PSF:

- futures
- heap
- typing

This product also includes the following libraries which are covered by The PSF license:

- functools32
- python

This product also includes the following libraries which are covered by The PSF or ZPL:

- wsgiref

This product also includes the following libraries which are covered by The Proprietary:

- com.trifacta.connect.trifacta-TYcassandra
- com.trifacta.connect.trifacta-TYdb2
- com.trifacta.connect.trifacta-TYgreenplum
- com.trifacta.connect.trifacta-TYhive
- com.trifacta.connect.trifacta-TYinformix
- com.trifacta.connect.trifacta-TYmongodb
- com.trifacta.connect.trifacta-TYmysql
- com.trifacta.connect.trifacta-TYopenedgewp
- com.trifacta.connect.trifacta-TYoracle
- com.trifacta.connect.trifacta-TYoraclesalescloud
- com.trifacta.connect.trifacta-TYpostgresql
- com.trifacta.connect.trifacta-TYredshift
- com.trifacta.connect.trifacta-TYrightnow
- com.trifacta.connect.trifacta-TYsforce
- com.trifacta.connect.trifacta-TYsparksql
- com.trifacta.connect.trifacta-TYsqlserver
- com.trifacta.connect.trifacta-TYsybase
- jsdata

This product also includes the following libraries which are covered by The Public Domain:

- aopalliance.aopalliance
- org.jboss.xnio.xnio-api
- org.jboss.xnio.xnio-nio
- org.tukaani.xz
- protobuf-to-dict
- pycrypto
- simple-xml-writer

This product also includes the following libraries which are covered by The Public domain:

- net.iharder.base64

This product also includes the following libraries which are covered by The Python Software Foundation License:

- argparse
- backports-abc
- ipaddress
- python-setuptools
- regex

This product also includes the following libraries which are covered by The Tableau Binary Code License Agreement:

- com.tableausoftware.common.tableaucommon
- com.tableausoftware.extract.tableauextract

This product also includes the following libraries which are covered by The Apache License, Version 2.0:

- com.fasterxml.woodstox.woodstox-core
- com.google.cloud.bigtable.bigtable-client-core
- com.google.cloud.datastore.datastore-v1-proto-client
- io.opencensus.opencensus-api
- io.opencensus.opencensus-contrib-grpc-metrics
- io.opencensus.opencensus-contrib-grpc-util
- io.opencensus.opencensus-contrib-http-util

Copyright © 2020 Trifacta Inc.

- org.spark-project.spark.unused
- software.amazon.ion.ion-java

This product also includes the following libraries which are covered by The BSD 3-Clause License:

- org.fusesource.leveldbjni.leveldbjni-all

This product also includes the following libraries which are covered by The GNU General Public License, Version 2:

- mysql.mysql-connector-java

This product also includes the following libraries which are covered by The Go license:

- com.google.re2j.re2j

This product also includes the following libraries which are covered by The MIT License (MIT):

- com.microsoft.azure.azure
- com.microsoft.azure.azure-annotations
- com.microsoft.azure.azure-client-authentication
- com.microsoft.azure.azure-client-runtime
- com.microsoft.azure.azure-keyvault
- com.microsoft.azure.azure-keyvault-core
- com.microsoft.azure.azure-keyvault-webkey
- com.microsoft.azure.azure-mgmt-appservice
- com.microsoft.azure.azure-mgmt-batch
- com.microsoft.azure.azure-mgmt-cdn
- com.microsoft.azure.azure-mgmt-compute
- com.microsoft.azure.azure-mgmt-containerinstance
- com.microsoft.azure.azure-mgmt-containerregistry
- com.microsoft.azure.azure-mgmt-containerservice
- com.microsoft.azure.azure-mgmt-cosmosdb
- com.microsoft.azure.azure-mgmt-dns
- com.microsoft.azure.azure-mgmt-graph-rbac
- com.microsoft.azure.azure-mgmt-keyvault
- com.microsoft.azure.azure-mgmt-locks
- com.microsoft.azure.azure-mgmt-network
- com.microsoft.azure.azure-mgmt-redis
- com.microsoft.azure.azure-mgmt-resources
- com.microsoft.azure.azure-mgmt-search
- com.microsoft.azure.azure-mgmt-servicebus
- com.microsoft.azure.azure-mgmt-sql
- com.microsoft.azure.azure-mgmt-storage
- com.microsoft.azure.azure-mgmt-trafficmanager
- com.microsoft.rest.client-runtime

This product also includes the following libraries which are covered by The MIT License: *http://www.opensource.org/licenses/mit-license.php*:

- probableparsing
- usaddress

This product also includes the following libraries which are covered by The New BSD License:

- net.sf.py4j.py4j
- org.jodd.jodd-core

This product also includes the following libraries which are covered by The Unicode/ICU License:

- com.ibm.icu.icu4j

This product also includes the following libraries which are covered by The Unlicense:

- moment-range
- stream-buffers

This product also includes the following libraries which are covered by The WTFPL:

- org.reflections.reflections

This product also includes the following libraries which are covered by The *http://www.apache.org/licenses/LICENSE-2.0*:

- tornado

This product also includes the following libraries which are covered by The new BSD:

- scikit-learn

This product also includes the following libraries which are covered by The new BSD License:

- decorator

This product also includes the following libraries which are covered by The provided without support or warranty:

- org.json.json

This product also includes the following libraries which are covered by The public domain, Python, 2-Clause BSD, GPL 3 (see COPYING.txt):

- docutils

This product also includes the following libraries which are covered by The zlib License:

- zlib

Copyright © 2020 - Trifacta, Inc.
All rights reserved.