TRIFACTA

Install Guide for AWS

Version: 6.4.1
Doc Build Date: 08/30/2019

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

Install Overview

Contents:

Basic Install Workflow

1. Review the pre-installation checklist and other system requirements. See *Install Preparation*.
2. Review the requirements for your specific installation scenario in the following sections.
3. Install the software. See *Install Software*.
4. Install the databases. See *Install Databases*.
5. Configure your installation.
6. Verify operations.

Notation

In this guide, JSON settings are provided in dot notation. For example, `webapp.selfRegistration` refers to a JSON block `selfRegistration` under `webapp`:

```
{
...
  "webapp": {
    "selfRegistration": true,
    ...
  }
...
}
```

Install for High Availability

Contents:

The Trifacta® platform can be installed across multiple nodes for high availability failover. This section describes the general process for installing the platform across multiple, highly available nodes.

- The Trifacta platform can also integrate with a highly available Hadoop cluster. For more information, see *Enable Integration with Cluster High Availability*.

Limitations

The following limitations apply to this feature:

- This form of high availability is not supported for Marketplace installations.

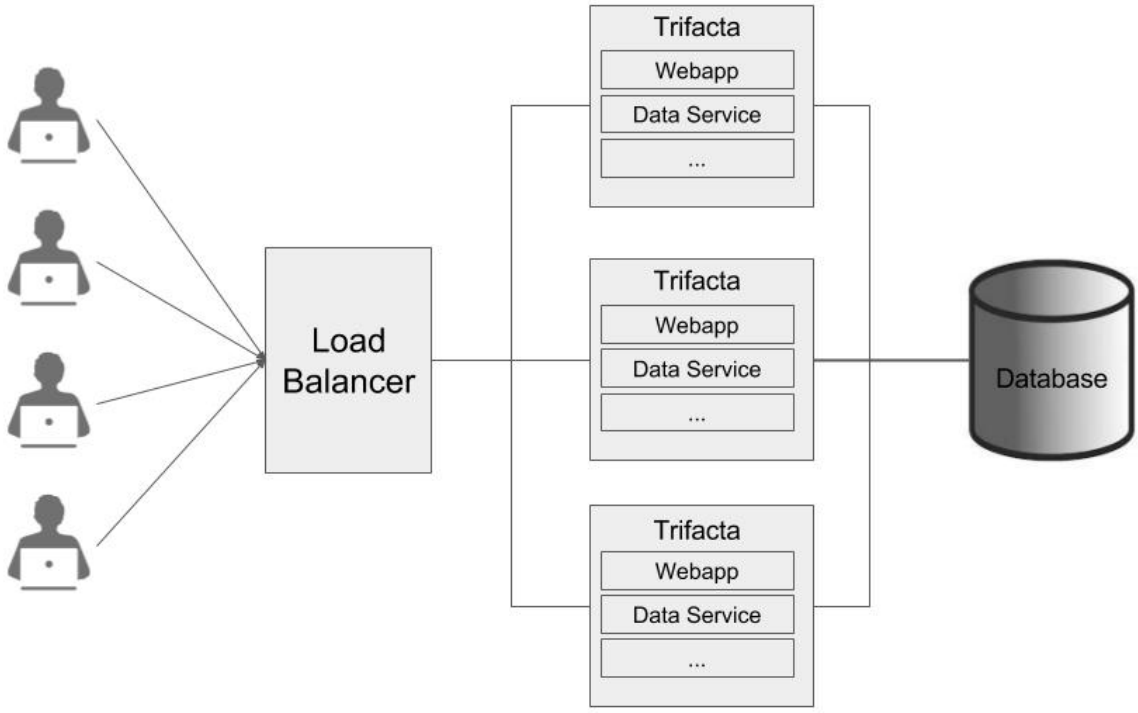
- Job canceling does not work.
- When HA is enabled, the restart feature in the Admin Settings page does not work. You must restart using the command line.
- The platform must be installed on `/opt/trifacta` on every failover node.
- This feature does not apply to the following components:
  - Hadoop cluster (See previous link.)
  - webhdfs/httpfs
  - Sentry
  - Navigator
  - Atlas
  - any other application/infrastructure with which the Trifacta platform can integrate

For more information, see *Configure for High Availability*.

Overview

The Trifacta platform supports an Active-Active HA deployment model, which works well at scale. The architecture features a single load balancer sitting in front of multiple nodes running the Trifacta platform. Each node:

- communicates with the same database
- shares the `/opt/trifacta/conf` and `/opt/trifacta/logs` directories through NFS.



- **Database:** PostGreSQL supports HA. The HA-enabled database runs outside of the cluster of platform nodes and appears to each node as a single database. No application code changes are required.
- **Load balancer:** HAProxy is used for its capabilities on health checking the other HA nodes. This load balance periodically checks the health of the other nodes in the setup.
  - If the health for a given node fails, then the load balancer stops routing traffic to that node while continuing to poll its health.
  - If the node recovers, the load balancer resumes sending traffic to it.
  - Node health is described below.
- **Synchronized configuration:** All nodes share the `/opt/trifacta/conf` mount point, which allows the same configuration files to be visible and accessible on each node.

Job interruption

In case of a failover event, any in-progress job should be marked as failed.

Failover events/scenarios around jobs:

#	Job	Event	Resulting job state
1	In progress	The batch job runner is fine, but executor running the job fails.	Failed
2	In progress	The batch job runner or the node dies.	In Progress
3	Queued	The batch job runner or the node dies.	In Progress[1]
4	Pending	The batch job runner or the node dies.	In Progress[1 2]

[1] It may not be "In Progress". However, the job has not truly failed.

2 A nuance around #3. There is a feature flag that can be enabled and is enabled by default, which causes pending jobs to be marked as failed on (re)start of batch job runner. However, because this feature indiscriminately marks *all* pending jobs as failed, it cannot be safely enabled in an environment that has multiple running batch job runners.

Installation Topography

The Trifacta platform supports a single load balancer placed in front of multiple nodes, each of which runs the same version of Trifacta Wrangler Enterprise. Content between nodes is shared using an NFS resource mount.

- **master node:** This node is the default one used for hosting and serving the Trifacta platform. Example node information:

```
NFS Server Hostname: server.local
NFS Server IP Address: 192.168.1.101
```

- **client node(s):** These nodes are failover nodes in case the master node is unavailable. Example node information:

```
NFS Client Hostname: client.local
NFS Client IP Address: 192.168.1.102
```

- **load balancer:** This documentation references set up for HAProxy as an example. If you are using a different load balancer, please consult the documentation that came with your product.

Shared resources:

Each node shares the following resources:

- Trifacta databases
- Directories shared via NFS mount:

```
/opt/trifacta/logs
/opt/trifacta/conf
```

Order of Installation

Steps:

1. All nodes must meet the system requirements. See *System Requirements*.
2. All nodes must have the appropriate ports opened. See *System Ports*.
3. Install the databases.

> **NOTE:** The databases must be installed in a location that is accessible to all nodes.

> **NOTE:** When installing databases for high availability access, you should deploy standard access and replication techniques that are consistent with the policies of your enterprise.

See *Install Databases*.

4. Complete the installation process for the server node.

   > **NOTE:** After install, do not start the Trifacta node.

   See *Install Software*.
5. Repeat the above process for each of the client nodes.
6. The software is installed on all nodes. No node is running the software.

Configuration

Additional configuration is required.

> **NOTE:** Starting and stopping the platform in high availability mode requires additional steps.

For more information, see *Configure for High Availability*.

Install for AWS

Contents:

This install process applies to installing Trifacta® Wrangler Enterprise on an AWS infrastructure that you manage.

AWS Marketplace deployments:

> **NOTE:** Content in this section does not apply to deployments from the AWS Marketplace, which provide fewer deployment and configuration options. For more information, see the AWS Marketplace.

Scenario Description

> **NOTE:** All hardware in use for supporting the platform is maintained within the enterprise infrastructure on AWS.

- Installation of Trifacta Wrangler Enterprise on an EC2 server in AWS
- Installation of Trifacta databases on AWS
- Integration with a supported EMR cluster.
- Base storage layer and backend datastore of S3

> **NOTE:** When the above installation and configuration steps have been completed, the platform is operational. Additional configuration may be required, which is referenced at the end of this section.

For more information on deployment scenarios, see *Supported Deployment Scenarios for AWS*.

Product Limitations

The following limitations apply to installations of Trifacta Wrangler Enterprise on AWS:

- No support for high availability and failover
- Job cancellation is not supported on EMR.
- When publishing single files to S3, you cannot apply an `append` publishing action.
- The following limitations apply to EMR integration only:
  - No support for Hive integration
  - No support for secure impersonation or Kerberos

Pre-requisites

Desktop Requirements

- All desktop users of the platform should have a supported version of Google Chrome installed on their desktops.
  - For more information. see *Desktop Requirements*.
  - If a supported browser is not available within your enterprise, desktop users can install the Trifacta enterprise application as a separate application. For more information, see *Install Desktop Application*.
- All desktop users must be able to connect to the EC2 instance through the enterprise infrastructure.

AWS Pre-requisites

Depending on which of the following AWS components you are deploying, additional pre-requisites and limitations may apply. Please review these sections as well.

- *Configure for EMR*
- *Enable S3 Access*
- *Create Redshift Connections*

Prep

Before you begin, please verify that you have completed the following:

1. **Review Planning Guide:** Please review and verify *Install Preparation* and sub-topics.
    1. **Limitations:** For more information on limitations of this scenario, see *Product Limitations* in the *Install Preparation* area.
2. **Read:** Please read this entire document before you create the EMR cluster or install the Trifacta platform.
3. **Acquire Assets:** Acquire the installation package for your operating system and your license key. For more information, contact *Trifacta Support*.
    1. If you are completing the installation without Internet access, you must also acquire the offline versions of the system dependencies. See *Install Dependencies without Internet Access*.
4. **VPC:** Enable and deploy a working AWS VPC.
5. **S3:** Enable and deploy an AWS S3 bucket to use as the base storage layer for the platform. In the bucket, the platform stores metadata in the following location:

    ```
    <S3_bucket_name>/trifacta
    ```

    See *https://s3.console.aws.amazon.com/s3/home*.
6. **IAM Policies:** Create IAM policies for access to the S3 bucket. Required permissions are the following:
    - The system account or individual user accounts must have full permissions for the S3 bucket:

        ```
        Delete*, Get*, List*, Put*, Replicate*, Restore*
        ```

    - These policies must apply to the bucket and its contents. Example:

        ```
        "arn:aws:s3:::my-trifacta-bucket-name"
        "arn:aws:s3:::my-trifacta-bucket-name/*"
        ```

    - See *https://console.aws.amazon.com/iam/home#/policies*
7. **EC2 instance:** Deploy an AWS EC2 with SELinux where the Trifacta software can be installed.
    1. The required set of ports must be enabled for listening. See *System Ports*.
    2. This node should be dedicated for Trifacta use.

    > **NOTE:** The EC2 node must meet the system requirements. For more information, see *System Requirements*.

8. **EC2 instance role:** Create an EC2 instance role for your S3 bucket policy. See *https://console.aws.amazon.com/iam/home#/roles*.
9. **EMR cluster:** An existing EMR cluster is required.
    1. **Cluster sizing:** Before you begin, you should allocate sufficient resources for sizing the cluster. For guidance, please contact your Trifacta representative.
    2. See Deploy the Cluster below.
10. **Databases:**

1. The platform utilizes a set of databases that must be accessed from the Trifacta node. Databases are installed as part of the workflow described later.
2. For more information on the supported databases and versions, see *System Requirements*.
3. For more information on database installation requirements, see *Install Databases*.
4. If installing databases on Amazon RDS an admin account to RDS is required. For more information, see *Install Databases on Amazon RDS*.

AWS Information

Before you begin installation, please acquire the following information from AWS:

- **EMR:**
  - AWS region for the EMR cluster, if it exists.
  - ID for EMR cluster, if it exists
    - If you are creating an EMR cluster as part of this process, please retain the ID.
    - The EMR cluster must allow access from the Trifacta node. This configuration is described later.
- **Subnet:** Subnet within your virtual private cloud (VPC) where you want to launch the Trifacta platform.
  - This subnet should be in the same VPC as the EMR cluster.
  - Subnet can be private or public.
  - If it is private and it cannot access the Internet, additional configuration is required. See below.
- **S3:**
  - Name of the S3 bucket that the platform can use
  - Path to resources on the S3 bucket
- **EC2:**
  - Instance type for the Trifacta node

Internet access

From AWS, the Trifacta platform requires Internet access for the following services:

> **NOTE:** Depending on your AWS deployment, some of these services may not be required.

- AWS S3
- Key Management System [KMS] (if sse-kms server side encryption is enabled)
- Secure Token Service [STS] (if temporary credential provider is used)
- EMR (if integration with EMR cluster is enabled)

> **NOTE:** If the Trifacta platform is hosted in a VPC where Internet access is restricted, access to S3, KMS and STS services must be provided by creating a VPC endpoint. If the platform is accessing an EMR cluster, a proxy server can be configured to provide access to the AWS ElasticMapReduce regional endpoint.

Deploy the Cluster

In your AWS infrastructure, you must deploy a supported version of EMR across a recommended number of nodes to support the expected data volumes of your Trifacta jobs.

- For more information on suggested sizing, see *Sizing Guidelines* in the *Install Preparation* area.

For more information on the supported EMR distributions, see *Supported Deployment Scenarios for AWS*.

When you configure the platform to integrate with the cluster, you must acquire some information about the cluster resources. For more information on the set of information to collect, see *Pre-Install Checklist* in the *Install Preparation* area.

Deploy the EC2 Node

An EC2 node of the cluster must be deployed to host the Trifacta platform software. For more information on the requirements of this node, see *System Requirements*.

When you configure the platform to integrate with the cluster, you must acquire some information about the cluster resources. For more information on the set of information to collect, see *Pre-Install Checklist* in the *Install Preparation* area.

Here are some guidelines for deploying the EC2 cluster from the EC2 cluster:

1. **Instance size:** Select the instance size.
2. **Network:** Configure the VPC, subnet, firewall and other configuration settings necessary to communicate with the instance.
3. **Auto-assigned Public IP:** You must create a public IP to access the Trifacta platform.
4. **EC2 role:** Select the EC2 role that you created.
5. **Local storage:** Select a local EBS volume. The default volume includes 100GB storage.

   > **NOTE:** The local storage environment contains the Trifacta databases, the product installation, and its log files. No source data is ever stored within the product.

6. **Security group:** Use a security group that exposes access to port 3005, which is the default port for the platform.
7. **Create an AWS key-pair for access:** This key is used to provide SSH access to the platform, which may be required for some admin tasks.
8. Save your changes.

Install Workflow

> **NOTE:** These steps are covered in greater detail later in this section.

After you have completed, the above, please complete these steps listed in order:

1 - Install software

Install the Trifacta platform software on the EC2 node you created. See *Install Software*.

2 - Install databases

The platform requires several databases for storing metadata.

NOTE: The software assumes that you are installing the databases on a PostgreSQL server on the same node as the software. If you are not or are changing database names or ports, additional configuration is required as part of this installation process.

For more information, see *Install Databases* in the Databases Guide.

3 - Login to the application

After software and databases are installed, you can login to the application to complete configuration. See *Login*.

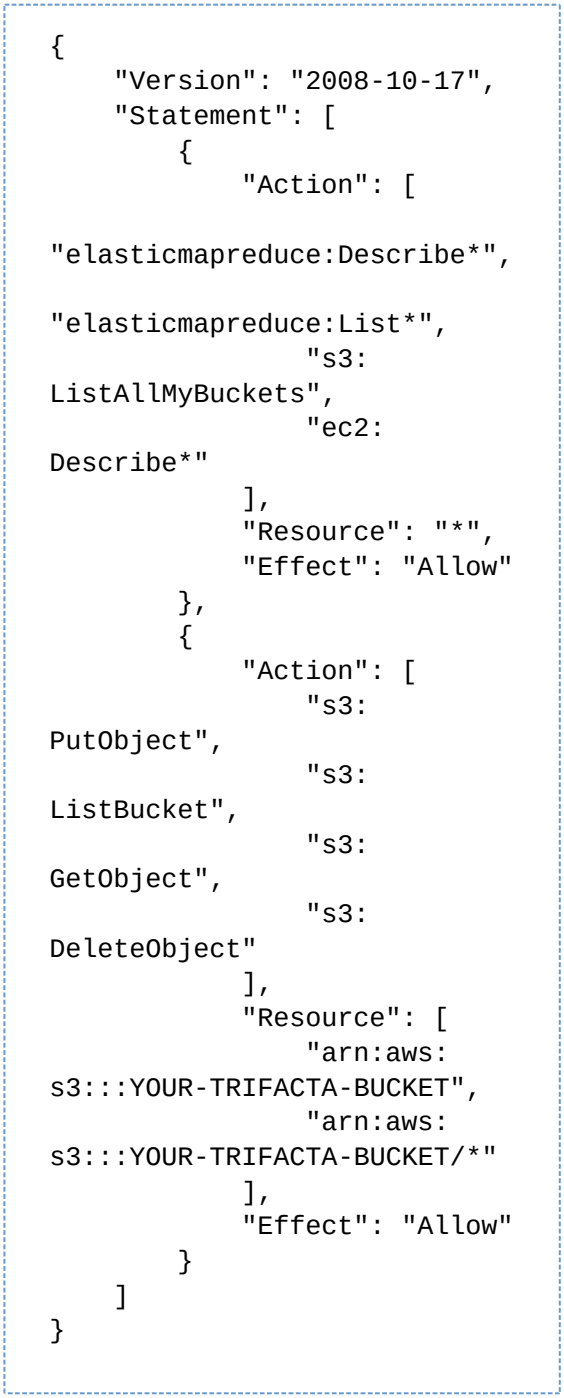
As soon as you login, you should change the password on the admin account. In the left menu bar, select **s > Admin Settings**

Please complete the following configuration to enable access to your pre-existing EMR cluster from the platform

You must make changes to your IAM and Security Group changes to enable the
communicate with your existing EMR cluster and your EMR cluster to read/write to the
are the requirements and suggested implementation details. Please adapt these suggestions to fit your environment as long as the requirements are satisfied.

For additional documentation around these changes:

Trifacta EC2 instance role must be permitted to use your EMR cluster.	{
	"Version": "2008-10-17",
	"Statement": [
	{
	"Action": [
	
	"elasticmapreduce:
	DescribeStep",
	
	"elasticmapreduce:
	ListBootstrapActions",
	
	"elasticmapreduce:
	ListClusters",
	
	"elasticmapreduce:
	DescribeCluster",
	
	"elasticmapreduce:
	AddJobFlowSteps",
	
	"elasticmapreduce:
	DescribeJobFlows",
	
	"elasticmapreduce:
	ListInstanceGroups"
	],
	"Resource": "*",
	"Effect": "Allow"
	}
	]
	}

EMR EC2 instance role must be permitted to use the Trifacta data bucket.	

Your EMR Service Role should permit access to the Trifacta bucket.
 	
NOTE: This example is not a complete policy. You should update your existing policy with these statements.
	<pre>{
	"Action": [
	"s3:HeadBucket",
	"s3:ListAllMyBuckets"
	],
	"Resource": "*",
	"Effect": "Allow"
	},
	{
	"Action": [
	"s3:PutObject",
	"s3:GetObject",
	"s3:ListBucket",
	"s3:DeleteObject"
	],
	"Resource": [
	"arn:aws:s3:::YOUR-TRIFACTA-BUCKET",
	"arn:aws:s3:::YOUR-TRIFACTA-BUCKET/*"
	],
	"Effect": "Allow"
	},</pre>

Your EMR cluster master node must permit the Trifacta EC2 instance to access it.	• The Trifacta EC2 instance must be able to communicate with your EMR master node on TCP ports 18080 and 8088. • You should create a security group and then associate it with your EMR master node using the "additional security groups" functionality. • For future ease of use, you should specify the security group associated with your Trifacta EC2 instance as the source.

Additional configuration must be applied within the platform. These steps are described later.

Additional Required Configuration for AWS Installs

Apply license key to EC2 node

Steps:

1. Acquire the `license.json` license key file that was provided to you by your Trifacta representative.
2. Transfer the license key file to the EC2 node that is hosting the Trifacta platform. Navigate to the directory where you stored it.
3. Make the Trifacta user the owner of the file:

```
sudo chown trifacta:trifacta license.json
```

4. Make sure that the Trifacta user has read permissions on the file:

```
sudo chmod 644 license.json
```

5. Copy the license key file to the proper location:

```
cp license.json /opt/trifacta/license/
```

Start the platform

For more information on how to launch the platform, see *Start and Stop the Platform*.

When the instance is spinning up for the first time, performance may be slow. When the instance is up, navigate to the following:

```
http://<public_hostname>:3005
```

When the login screen appears, enter the default admin credentials provided to you.

> **NOTE:** As soon as you login as an admin for the first time, you should immediately change the password. From the left nav bar, select **Settings > Settings > User Profile**. Change the password and click **Save** to restart the platform.

Configure for EMR clusters

The following steps apply to configure the platform to integrate with the EMR cluster:

1. From the application menu, select the Settings menu. Then, click **Settings > Admin Settings**.
2. In the Admin Settings page, you can configure many aspects of the platform, including user management tasks, and perform restarts to apply the changes.
    1. In the Search bar, enter the following:

    ```
    aws.s3.bucket.name
    ```

    2. Set the value of this setting to be S3 bucket name.
3. Check the following setting. Verify that it is set to `2.3.0`:

```
"spark.version": "2.3.0",
```

4. The following setting must be specified.

```
"aws.mode":"system",
```

You can set the above value to either of the following:

aws.mode value	Description
`system`	Set the mode to `system` to enable use of EC2 instance-based authentication for access.
`user`	Set the mode to `user` to utilize user-based credentials. This mode requires additional configuration.

Details on the above configuration are described later.

5. Set the following parameter to `true`, which instructs the Trifacta application to run jobs on the integrated EMR cluster:

```
"webapp.runinEMR" = true,
```

6. In the Admin Settings page, locate the External Service Settings section.
7. In the Admin Settings page, locate the External Service Settings section.

    1. **AWS EMR Cluster ID:** Paste the value for the EMR Cluster ID for the cluster to which the platform is connecting.
    2. **AWS Region:** Enter the region where your EMR cluster is located.
    3. **Resource Bucket:** Enter the name of the S3 bucket to use.
    4. **Resource Path:** you should use something like `EMRLOGS`.
8. Click **Save** underneath the External Service Settings section.

Set base storage layer

The platform requires that one backend datastore be configured as the base storage layer. This base storage layer is used for storing uploaded data and writing results and profiles.

> **NOTE:** By default, the base storage layer for Trifacta Wrangler Enterprise is set to HDFS. You must change this value for S3. After this base storage layer is defined, it cannot be changed again.

See *Set Base Storage Layer*.

Verify Operations

> **NOTE:** You can try to verify operations using the Trifacta Photon running environment at this time. While you can also try to run a job in the Spark running environment, additional configuration may be required to complete the integration. These steps are listed under Next Steps below.

Prepare Your Sample Dataset

To complete this test, you should locate or create a simple dataset. Your dataset should be created in the format that you wish to test.

Characteristics:

- Two or more columns.
- If there are specific data types that you would like to test, please be sure to include them in the dataset.
- A minimum of 25 rows is required for best results of type inference.
- Ideally, your dataset is a single file or sheet.

Store Your Dataset

If you are testing an integration, you should store your dataset in the datastore with which the product is integrated.

> **Tip:** Uploading datasets is always available as a means of importing datasets.

- You may need to create a connection between the platform and the datastore.
- Read and write permissions must be enabled for the connecting user to the datastore.
- For more information, see *Connections Page*.

Verification Steps

Steps:

1. Login to the application. See *Login*.
2. In the application menu bar, click **Library**.
3. Click **Import Data**. See *Import Data Page*.
    1. Select the connection where the dataset is stored. For datasets stored on your local desktop, click **Upload**.
    2. Select the dataset.
    3. In the right panel, click the Add Dataset to a Flow checkbox. Enter a name for the new flow.
    4. Click **Import and Add to Flow**.
    5. **Troubleshooting:** At this point, you have read access to your datastore from the platform. If not, please check the logs, permissions, and your Trifacta® configuration.
4. In the left menu bar, click the Flows icon. Flows page, open the flow you just created. See *Flows Page*.
5. In the Flows page, click the dataset you just imported. Click **Add new Recipe**.
6. Select the recipe. Click **Edit Recipe**.
7. The initial sample of the dataset is opened in the Transformer page, where you can edit your recipe to transform the dataset.
    1. In the Transformer page, some steps are automatically added to the recipe for you. So, you can run the job immediately.
    2. You can add additional steps if desired. See *Transformer Page*.
8. Click **Run Job**.
    1.
    2. If options are presented, select the defaults.
    3. To generate results in other formats or output locations, click **Add Publishing Destination.** Configure the output formats and locations.
    4. To test dataset profiling, click the Profile Results checkbox. Note that profiling runs as a separate job and may take considerably longer.
    5. See *Run Job Page*.

6.

Troubleshooting: Later, you can re-run this job on a different running environment. Some formats are not available across all running environments.

9. When the job completes, you should see a success message under the Jobs tab in the Flow View page.
    1. **Troubleshooting:** Either the Transform job or the Profiling job may break. To localize the problem, try re-running a job by deselecting the broken job type or running the job on a different running environment (if available). You can also download the log files to try to identify the problem. See *Job Details Page*.
10. Click **View Results** from the context menu for the job listing. In the Job Details page, you can see a visual profile of the generated results. See *Job Details Page*.
11. In the Output Destinations tab, click a link to download the results to your local desktop.
12. Load these results into a local application to verify that the content looks ok.

Checkpoint: You have verified importing from the selected datastore and transforming a dataset. If your job was successfully executed, you have verified that the product is connected to the job running environment and can write results to the defined output location. Optionally, you may have tested profiling of job results. If all of the above tasks completed, the product is operational end-to-end.

Documentation

Tip: You should access online documentation through the product. Online content may receive updates that are not present in PDF content.

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

Next Steps

After you have accessed the documentation, the following topics are relevant to AWS enterprise infrastructure deployments.

NOTE: These materials are located in the *Configuration Guide*.

Please review them in order.

Topic	Description
Required Platform Configuration	This section covers the following topics, some of which should already be completed: • *Set Base Storage Layer* - The base storage layer must be set once and never changed. Set this value to `s3`. • *Create Encryption Key File* - If you plan to integrate the platform with any relational sources, including Redshift, you must create an encryption key file and store it on the Trifacta node • *Running Environment Options* - Depending on your scenario, you may need to perform additional configuration for your available running environment(s) for executing jobs. • *Profiling Options* - In some environments, tweaks to the settings for visual profiling may be required. You can disable visual profiling if needed. • *Configure for Spark* - If you are enabling the Spark running environment, please review and verify the configuration for integrating the platform with the Spark running environment.
Configure for EMR	Set up for a new EMR cluster. Some content may apply to existing EMR clusters.

Enable Integration with Compressed Clusters	If the Hadoop cluster uses compression, additional configuration is required.
Enable Integration with Cluster High Availability	If you are integrating with high availability on the Hadoop cluster, please complete these steps. • If you are integrating with high availability on the Hadoop cluster, HttpFS must be enabled in the platform. HttpFS is required in other, less-common cases. See *Enable HttpFS*.
Enable Relational Connections	Enable integration with relational databases, including Redshift. • For more information on creating a connection to Redshift, see *Create Redshift Connections*.
Configure for KMS	Integration with the Hadoop cluster's key management system (KMS) for encrypted transport. Instructions are provided for distribution-specific versions of Hadoop.
Configure Security	A list of topics on applying additional security measures to the Trifacta platform and how integrates with Hadoop.
Configure SSO for AD-LDAP	Please complete these steps if you are integrating with your enterprise's AD/LDAP Single Sign-On (SSO) system.

Upgrade

For more information on upgrading your Trifacta Wrangler Enterprise on AWS, please contact *Trifacta Customer Success Services*.

Configure Server Access through Proxy

When you attempt to launch the application, you may receive an error message similar to the following:

```
No internet connection
Remote server timed out.
```

In some environments, your desktop machine may need to connect to the Internet through a proxy server. If you are using Wrangler Enterprise desktop application, it needs to know the proxy server to which to connect in order to access the Trifacta® node.

Please complete the following configuration steps to access the Trifacta servers.

Steps:

1. In the No internet connection dialog, click **Configure Proxy Settings**.
2. Please provide the following configuration information for your proxy server:
    1. **Proxy Host:** The URL of the proxy server. Please include the protocol identifier (e.g. `http://` or `https://`).
    2. **Proxy Port:** The port number to use to connect to the proxy server. In a URL, this value appears after a colon (e.g. `http://myproxy.example.com:8080`).
    3. **Username:** (optional) If your proxy requires a username to access, please enter it here.
    4. **Password:** (optional) Password associated with the user name.
3. Click **Save Proxy Settings and Restart**.

When the application restarts, you should be able to connect to the login screen.

NOTE: If you continue to have difficulties connecting to the Internet, please contact your network administrator or Internet provider.

Install Software

To install Trifacta® Wrangler Enterprise, please review and complete the following sections in the order listed below.

Topics:

Install Dependencies without Internet Access

Offline dependencies should be included in the URL location that Trifacta® provided to you. Please use the `\*deps\*` file.

NOTE: If your installation server is connected to the Internet, the required dependencies are automatically downloaded and installed for you. You may skip this section.

Use the steps below to acquire and install dependencies required by the Trifacta platform. If you need further assistance, please contact *Trifacta Support*.

Install software dependencies without Internet access for CentOS or RHEL:

1. In a CentOS or RHEL environment, the dependencies repository must be installed into the following directory:

   ```
   /var/local/trifacta
   ```

2. The following commands configure Yum to point to the repository in `/var/local/trifacta`, which yum knows as `local`. Repo permissions are set appropriately. Commands:

   ```
   tar xvzf <DEPENDENCIES_ARCHIVE>.tar.gz
   mv local.repo /etc/yum.repos.d
   mv trifacta /var/local
   chown -R root:root /var/local/trifacta
   chmod -R o-w+r /var/local/trifacta
   ```

3. The following command installs the RPM while disable all repos other than local, which prevents the installer from reaching out to the Internet for package updates:

   NOTE: The disabling of repositories only applies to this command.

```
sudo yum --disablerepo=* --enablerepo=local install <INSTALLER>.rpm
```

4. If the above command fails and complains about a missing repo, you can add the missing repo to the `enablerepo` list. For example, if the `centos-base` repo is reported as missing, then the command would be the following:

```
sudo yum --disablerepo=* --enablerepo=local,centos-base install
<INSTALLER>.rpm
```

5. If you do not have a supported version of a Java Developer Kit installed on the Trifacta node, you can use the following command to install OpenJDK, which is included in the offline dependencies:

```
sudo yum --disablerepo=* --enablerepo=local,centos-base install
java-1.8.0-openjdk-1.8.0 java-1.8.0-openjdk-devel
```

Install database dependencies without Internet access:

If you are installing the databases on a node without Internet access, you can install the dependencies using either of the following commands:

> **NOTE:** This step is only required if you are installing the databases on the same node where the software is installed.

For PostgreSQL:

```
sudo yum --disablerepo=* --enablerepo=local install postgresql96-server
```

For MySQL:

```
sudo yum --disablerepo=* --enablerepo=local install mysql-community-
server
```

> **NOTE:** You must also install the MySQL JARs on the Trifacta node. These instructions are provided later.

Database are installed after the software is installed. For more information, see *Install Databases*.

Install dependencies without Internet access in Ubuntu:

If you are trying to perform a manual installation of dependencies in Ubuntu, please contact *Trifacta Support*.

Install on CentOS and RHEL

Contents:

This guide takes you through the steps for installing Trifacta® Wrangler Enterprise software on CentOS or Red Hat.

For more information on supported operating system versions, see *System Requirements*.

Preparation

Before you begin, please complete the following.

> **NOTE:** Except for database installation and configuration, all install commands should be run as the root user or a user with similar privileges. For database installation, you will be asked to switch the database user account.

Steps:

1. Set the node where Trifacta Wrangler Enterprise is to be installed.
   1. Review the *System Requirements* and verify that all required components have been installed.
   2. Verify that all required system ports are opened on the node. See *System Ports*.
2. Review the *Desktop Requirements*.

   > **NOTE:** Trifacta Wrangler Enterprise requires the installation of Google Chrome on each desktop. For more information, see *Desktop Requirements*.

3. Review the *System Dependencies*.

   > **NOTE:** If you are installing on node without access to the Internet, you must download the offline dependencies before you begin. See *Install Dependencies without Internet Access*.

4. Acquire your *License Key*.
5. Install and verify operations of the datastore, if used.

   > **NOTE:** Access to the Spark cluster is required.

6. Verify access to the server where the Trifacta platform is to be installed.
7. **Cluster Configuration:** Additional steps are required to integrate the Trifacta platform with the cluster. See *Prepare Hadoop for Integration with the Platform*.

Installation

1. Install Dependencies

Without Internet access

If you have not done so already, you may download the dependency bundle with your release directly from Trifacta . For more information, see *Install Dependencies without Internet Access*.

With Internet access

Use the following to add the hosted package repository for CentOS/RHEL, which will automatically install the proper packages for your environment.

```
# If the client has curl installed ...
curl https://packagecloud.io/install/repositories/trifacta/dependencies
/script.rpm.sh | sudo bash

# Otherwise, you can also use wget ...
wget -qO- https://packagecloud.io/install/repositories/trifacta
/dependencies/script.rpm.sh | sudo bash
```

2. Install JDK

By default, the Trifacta node uses OpenJDK for accessing Java libraries and components. In some environments, basic setup of the node may include installation of a JDK. Please review your environment to verify that an appropriate JDK version has been installed on the node.

NOTE: Use of Java Development Kits other than OpenJDK is not currently supported. However, the platform may work with the Java Development Kit of your choice, as long as it is compatible with the supported version(s) of Java. See *System Requirements*.

NOTE: OpenJDK is included in the offline dependencies, which can be used to install the platform without Internet access. For more information, see *Install Dependencies without Internet Access*.

The following commands can be used to install OpenJDK. These commands can be modified to install a separate compatible version of the JDK.

```
sudo yum install java-1.8.0-openjdk-1.8.0 java-1.8.0-openjdk-devel
```

NOTE: If `java-1.8.0-openjdk-devel` is not included, the batch job runner service, which is required, fails to start.

JAVA_HOME:

By default, the `JAVA_HOME` environment variable is configured to point to a default install location for the OpenJDK package.

> **NOTE:** If you have installed a JDK other than the OpenJDK version provided with the software, you must set the `JAVA_HOME` environment variable on the Trifacta node to point to the correct install location.

The property value must be updated in the following locations:

1. Edit the following file: `/opt/trifacta/conf/env.sh`
2. Save changes.

3. Install Trifacta package

> **NOTE:** If you are installing without Internet access, you must reference the local repository. The command to execute the installer is slightly different. See *Install Dependencies without Internet Access*.

> **NOTE:** Installing the Trifacta platform in a directory other than the default one is not supported or recommended.

Install the package with yum, using root:

```
sudo yum install <rpm file>
```

4. Verify Install

The product is installed in the following directory:

```
/opt/trifacta
```

JAVA_HOME:

The platform must be made aware of the location of Java.

Steps:

1. Edit the following file: `/opt/trifacta/conf/trifacta-conf.json`
2. Update the following parameter value:

```
"env": {
  "JAVA_HOME": "/usr/lib/jvm/java-1.8.0-openjdk.x86_64"
},
```

3. Save changes.

5. Install License Key

Please install the license key provided to you by Trifacta. See *License Key*.

6. Store install packages

For safekeeping, you should retain all install packages that have been installed with this Trifacta deployment.

7. Install and configure Trifacta databases

The Trifacta platform requires installation of several databases. If you have not done so already, you must install and configure the databases used to store Trifacta metadata. See *Install Databases*.

Configuration

After installation is complete, additional configuration is required.

> **The Trifacta platform requires additional configuration for a successful integration with the datastore. Please review and complete the necessary configuration steps. For more information, see *Configure*.**

Install on Ubuntu

Contents:

This guide takes you through the steps for installing Trifacta® Wrangler Enterprise software on Ubuntu.

For more information on supported operating system versions, see *System Requirements*.

Preparation

Before you begin, please complete the following.

> **NOTE:** Except for database installation and configuration, all install commands should be run as the root user or a user with similar privileges. For database installation, you will be asked to switch the database user account.

Steps:

1. Set the node where Trifacta Wrangler Enterprise is to be installed.
    1. Review the *System Requirements* and verify that all required components have been installed.
    2. Verify that all required system ports are opened on the node. See *System Ports*.
2. Review the *Desktop Requirements*.

    > **NOTE:** Trifacta Wrangler Enterprise requires the installation of Google Chrome on each desktop.

3. Review the *System Dependencies*.

    > **NOTE:** If you are installing on node without access to the Internet, you must download the offline dependencies before you begin. See *Install Dependencies without Internet Access*.

4. Acquire your *License Key*.
5. Install and verify operations of the datastore, if used.

    > **NOTE:** Access to the cluster may be required.

6. Verify access to the server where the Trifacta platform is to be installed.
7. **Cluster configuration:** Additional steps are required to integrate the Trifacta platform with the cluster. See *Prepare Hadoop for Integration with the Platform*.

Installation

1. Install Dependencies

Without Internet access

If you have not done so already, you may download the dependency bundle with your release directly from Trifacta . For more information, see *Install Dependencies without Internet Access*.

With Internet access

Use the following to add the hosted package repository for Ubuntu, which will automatically install the proper packages for your environment.

> **NOTE:** Install curl if not present on your system.

Then, execute the following command:

> **NOTE:** Run the following command as the root user. In proxied environments, the script may encounter issues with detecting proxy settings.

```
curl https://packagecloud.io/install/repositories/trifacta/dependencies
/script.deb.sh | sudo bash
```

Special instructions for Ubuntu installs

These steps manually install the correct and supported version of the following:

- nodeJS
- nginX

Due to a known issue resolving package dependencies on Ubuntu, please complete the following steps prior to installation of other dependencies or software.

1. Login to the Trifacta node as an administrator.
2. Execute the following command to install the appropriate versions of nodeJS and nginX.
    1. Ubuntu 14.04:

```
sudo apt-get install nginx=1.12.2-1~trusty nodejs=10.13.0-
1nodesource1
```

    2. Ubuntu 16.04

```
sudo apt-get install nginx=1.12.2-1~xenial nodejs=10.13.0-
1nodesource1
```

3. Continue with the installation process.

2. Install JDK

By default, the Trifacta node uses OpenJDK for accessing Java libraries and components. In some environments, basic setup of the node may include installation of a JDK. Please review your environment to verify that an appropriate JDK version has been installed on the node.

> **NOTE:** Use of Java Development Kits other than OpenJDK is not currently supported. However, the platform may work with the Java Development Kit of your choice, as long as it is compatible with the supported version(s) of Java. See *System Requirements*.

> **NOTE:** OpenJDK is included in the offline dependencies, which can be used to install the platform without Internet access. For more information, see *Install Dependencies without Internet Access*.

The following commands can be used to install OpenJDK. These commands can be modified to install a separate compatible version of the JDK.

```
sudo apt-get install openjdk-8-jre-headless
```

JAVA_HOME:

By default, the `JAVA_HOME` environment variable is configured to point to a default install location for the OpenJDK package.

NOTE: If you have installed a JDK other than the OpenJDK version provided with the software, you must set the JAVA_HOME environment variable on the Trifacta node to point to the correct install location.

The property value must be updated in the following locations:

1. Edit the following file: `/opt/trifacta/conf/env.sh`
2. Save changes.

3. Install Trifacta package

NOTE: If you are installing without Internet access, you must reference the local repository. The command to execute the installer is slightly different. See *Install Dependencies without Internet Access*.

NOTE: Installing the Trifacta platform in a directory other than the default one is not supported or recommended.

Install the package with apt, using root:

```
sudo dpkg -i <deb file>
```

The previous line may return an error message, which you may ignore. Continue with the following command:

```
sudo apt-get -f -y install
```

4. Verify Install

The product is installed in the following directory:

```
/opt/trifacta
```

JAVA_HOME:

The platform must be made aware of the location of Java.

Steps:

1. Edit the following file: `/opt/trifacta/conf/trifacta-conf.json`
2. Update the following parameter value:

```
"env": {
  "JAVA_HOME": "/usr/lib/jvm/java-1.8.0-openjdk.x86_64"
},
```

3. Save changes.

5. Install License Key

Please install the license key provided to you by Trifacta. See *License Key*.

6. Store install packages

For safekeeping, you should retain all install packages that have been installed with this Trifacta deployment.

7. Install and configure Trifacta databases

The Trifacta platform requires installation of several databases. If you have not done so already, you must install and configure the databases used to store Trifacta metadata. See *Install Databases*.

Configuration

After installation is complete, additional configuration is required.

> **The Trifacta platform requires additional configuration for a successful integration with the datastore. Please review and complete the necessary configuration steps. For more information, see *Configure*.**

Install for Docker

Contents:

This guide steps through the process of acquiring and deploying a Docker image of the Trifacta® platform in your Docker environment. Optionally, you can build the Docker image locally, which enables further configuration options.

Deployment Scenario

- Trifacta Wrangler Enterprise deployed into a customer-managed environment: On-premises, AWS, or Azure.
- PostgreSQL 9.6 installed either:
  - Locally
  - Remote server

Limitations

- You cannot upgrade to a Docker image from a non-Docker deployment.
- You cannot switch an existing installation to a Docker image.
- Supported distributions of Cloudera or Hortonworks:
  - *Supported Deployment Scenarios for Cloudera*
  - *Supported Deployment Scenarios for Hortonworks*
- The base storage layer of the platform must be HDFS. Base storage of S3 is not supported.
- High availability for the Trifacta platform in Docker is not supported.
- SSO integration is not supported.

Requirements

Support for orchestration through Docker Compose only

- Docker version 17.12 or later
- Docker-Compose 1.11.2 or later. Version must be compatible with your version of Docker.

Docker Daemon

	Minimum	Recommended
CPU Cores	2 CPU	4 CPU
Available RAM	8 GB RAM	10+ GB RAM

Preparation

1. Review the *Desktop Requirements*.

   > **NOTE:** Trifacta Wrangler Enterprise requires the installation of Google Chrome on each desktop. For more information, see *Desktop Requirements*.

2. Acquire your *License Key*.

Acquire Image

You can acquire the latest Docker image using one of the following methods:

1. Acquire from FTP site.
2. Build your own Docker image.

Acquire from FTP site

Steps:

1. Download the following files from the FTP site:
    1. `trifacta-docker-setup-bundle-x.y.z.tar`
    2. `trifacta-docker-image-x.y.z.tar`

    > **NOTE:** `x.y.z` refers to the version number (e.g. `6.4.0`).

2. Untar the `setup-bundle` file:

    ```
    tar xvf trifacta-docker-setup-bundle-x.y.z.tar
    ```

3. Files are extracted into a `docker` folder. Key files:

File	Description
docker-compose-local-postgres.yaml	Runtime configuration file for the Docker image when PostgreSQL is to be running on the same machine. More information is provided below.
docker-compose-local-mysql.yaml	Runtime configuration file for the Docker image when MySQL is to be running on the same machine. More information is provided below.
docker-compose-remote-db.yaml	Runtime configuration file for the Docker image when the database is to be accessed from a remote server. **NOTE:** You must manage this instance of the database. More information is provided below.
README-running-trifacta-container.md	Instructions for running the Trifacta container **NOTE:** These instructions are referenced later in this workflow.
README-building-trifacta-container.md	Instructions for building the Trifacta container **NOTE:** This file does not apply if you are using the provided Docker image.

4. Load the Docker image into your local Docker environment:

    ```
    docker load < trifacta-docker-image-x.y.z.tar
    ```

5. Confirm that the image has been loaded. Execute the following command, which should list the Docker image:

```
docker images
```

6. You can now configure the Docker image. Please skip that section.

Build your own Docker image

As needed, you can build your own Docker image.

Requirements

- Docker version 17.12 or later
- Docker Compose 1.11.2 or newer. It should be compatible with above version of Docker.

Build steps

1. Acquire the RPM file from the FTP site:

   NOTE: You must acquire the el7 RPM file for this release.

2. In your Docker environment, copy the `trifacta-server\*.rpm` file to the same level as the `Dockerfile`.
3. Verify that the `docker-files` folder and its contents are present.
4. Use the following command to build the image:

```
docker build -t trifacta/server-enterprise:latest .
```

5. This process could take about 10 minutes. When it is completed, you should see the build image in the Docker list of local images.

   NOTE: To reduce the size of the Docker image, the Dockerfile installs the trifacta-server RPM file in one stage and then copies over the results to the final stage. The RPM is not actually installed in the final stage. All of the files are properly located.

6. You can now configure the Docker image.

Configure Docker Image

Before you start the Docker container, you should review the properties for the Docker image. In the provided image, please open the appropriate `docker-compose` file:

File	Description
docker-compose-local-postgres.yaml	Database properties in this file are pre-configured to work with the installed instance of PostgreSQL, although you may wish to change some of the properties for security reasons.

docker-compose-local-mysql.yaml	Database properties in this file are pre-configured to work with the installed instance of MySQL, although you may wish to change some of the properties for security reasons.
docker-compose-remote-db.yaml	The Trifacta databases are to be installed on a remote server that you manage. **NOTE:** Additional configuration is required.

NOTE: You may want to create a backup of this file first.

Key general properties:

NOTE: Avoid modifying properties that are not listed below.

Property	Description
image	This reference must match the name of the image that you have acquired.
container_name	Name of container in your Docker environment.
ports	Defines the listening port for the Trifacta application. Default is `3005`. **NOTE:** If you must change the listening port, additional configuration is required after the image is deployed. See *Change Listening Port* For more information, see *System Ports*.

Database properties:

These properties pertain to the installation of the database to which the Trifacta application connects.

Property	Description
DB_INIT	If set to `true`, database initialization steps are performed at startup. **NOTE:** This step applies only if you are starting the container for the first time, and PostgreSQL databases will be installed locally.
DB_TYPE	Set this value to `postgresql` or `mysql`.
DB_HOST_NAME	Hostname of the machine hosting the databases. Leave value as `localhost` for local installation.

DB_HOST_PORT	(Remote only) Port number to use to connect to the databases. Default is `5432`. **NOTE:** If you are modifying, additional configuration is required after installation is complete. See *Change Database Port*.
DB_ADMIN_USERNAME	Admin username to be used to create DB roles/databases. Modify this value for remote installation. **NOTE:** If you are modifying this value, additional configuration is required. Please see the documentation for your database version.
DB_ADMIN_PASSWORD	Admin password to be used to create DB roles/databases. Modify this value for remote installation.

Kerberos properties:

If your Hadoop cluster is protected by Kerberos, please review the following properties.

Property	Description
KERBEROS_KEYTAB_FILE	Full path inside of the container where the Kerberos keytab file is located. Default value: `/opt/trifacta/conf/trifacta.keytab` **NOTE:** The keytab file must be imported and mounted to this location. Configuration details are provided later.
KERBEROS_KRB5_CONF	Full path inside of the container where the Kerberos `krb5.conf file` is located. Default: `/opt/krb-config/krb5.conf`

Hadoop distribution client JARs:

Please enable the appropriate path to the client JAR files for your Hadoop distribution. In the following example, the Cloudera path has been enabled, and the Hortonworks path has been disabled:

```
# Mount folder from outside for necessary hadoop client jars
# For CDH
- /opt/cloudera:/opt/cloudera
# For HDP
#- /usr/hdp:/usr/hdp
```

Please modify these lines if you are using Hortonworks.

Volume properties:

These properties govern where volumes are mounted in the container.

> **NOTE:** These values should not be modified unless necessary.

Property	Description
volumes.conf	Full path in container to the Trifacta configuration directory. Default: `/opt/trifacta/conf`
volumes.logs	Full path in container to the Trifacta logs directory. Default: `/opt/trifacta/logs`
volumes.license	Full path in container to the Trifacta license directory. Default: `/trifacta-license`

Start Server Container

After you have performed the above configuration, execute the following to initialize the Docker container:

```
docker-compose -f <docker-compose-filename>.yaml run trifacta initfiles
```

When the above is started for the first time, the following directories are created on the localhost:

Directory	Description
./trifacta-data	Used by the Trifacta container to expose the `conf` and `logs` directories.

Import Additional Configuration Files

After you have started the new container, additional configuration files must be imported.

Import license key file

The Trifacta license file must be staged for use by the platform. Stage the file in the following location in the container:

NOTE: If you are using a non-default path or filename, you must update the `<docker-compose-filename> .yaml` file.

```
trifacta-license/license.json
```

Import Hadoop distribution libraries

If the container you are creating is on the edge node of your Hadoop cluster, you must provide the Hadoop libraries.

1. You must mount the Hadoop distribution libraries into the container. For more information on the libraries, see the documentation for your Hadoop distribution.
2. The Docker Compose file must be made aware of these libraries. Details are below.

Import Hadoop cluster configuration files

Some core cluster configuration files from your Hadoop distribution must be provided to the container. These files must be copied into the following directory within the container:

```
./trifacta-data/conf/hadoop-site
```

For more information, see *Configure for Hadoop* in the Configuration Guide.

Install Kerberos client

If Kerberos is enabled, you must install the Kerberos client and keytab on the node container. Copy the keytab file to the following stage location:

```
/trifacta-data/conf/trifacta.keytab
```

See *Configure for Kerberos Integration* in the Configuration Guide.

Perform configuration changes as necessary

The primary configuration file for the platform is in the following location in the launched container:

```
/opt/trifacta/conf/trifacta-conf.json
```

NOTE: Unless you are comfortable working with this file, you should avoid direct edits to it. All subsequent configuration can be applied from within the application, which supports some forms of data validation. It is possible to corrupt the file using direct edits.

Configuration topics are covered later.

Start and Stop the Container

Stop container

Stops the container but does not destroy it.

> **NOTE:** Application and local database data is not destroyed. As long as the `<docker-compose-filename> .yaml` properties point to the correct location of the `*-data` files, data should be preserved. You can start new containers to use this data, too. Do not change ownership on these directories.

```
docker-compose -f <docker-compose-filename>.yaml stop
```

Restart container

Restarts an existing container.

```
docker-compose -f <docker-compose-filename>.yaml start
```

Recreate container

Recreates a container using existing local data.

```
docker-compose -f <docker-compose-filename>.yaml up --force-recreate -d
```

Stop and destroy the container

Stops the container and destroys it.

> **The following also destroys all application configuration, logs, and database data. You may want to back up these directories first.**

```
docker-compose -f <docker-compose-filename>.yaml down
```

Local PostgreSQL:

```
sudo rm -rf trifacta-data/ postgres-data/
```

Local MySQL or remote database:

```
sudo rm -rf trifacta-data/
```

Verify Deployment

1. Verify access to the server where the Trifacta platform is to be installed.
2. **Cluster Configuration:** Additional steps are required to integrate the Trifacta platform with the cluster. See *Prepare Hadoop for Integration with the Platform*.
3. Start the platform within the container. See *Start and Stop the Platform*.

Configuration

After installation is complete, additional configuration is required. You can complete this configuration from within the application.

Steps:

1. Login to the application. See *Login*.
2. The primary configuration interface is the Admin Settings page. From the left menu, select **Settings menu > Settings > Admin Settings**. For more information, see *Admin Settings Page* in the Admin Guide.
3. Workspace-level configuration can also be applied. From the left menu, select **Settings menu > Settings > Workspace Admin**. For more information, see *Workspace Admin Page* in the Admin Guide.

The Trifacta platform requires additional configuration for a successful integration with the datastore. Please review and complete the necessary configuration steps. For more information, see *Configure* in the Configuration Guide.

License Key

Contents:

Download license key file

If you have not done so already, the license key file is available where you have acquired the installation package. Please download `license.json`.

Acquire license key

A valid license key (`license.json`) is provided to each customer prior to installation. Your license key file is a JSON file that contains important information on your license.

NOTE: If your license key has expired, please contact *Trifacta Support*.

Install your license key

If you are updating your license, you may want to save your previous license key to a new location before overwriting.

NOTE: Do not maintain multiple license key files in this directory.

To apply your license key, copy the key file to the following location in the Trifacta® deployment:

```
/opt/trifacta/license
```

Update your license key

After you have installed your license key, you can update your license with a new one through the Admin Settings page. See *Admin Settings Page*.

Changing the license key location

By default, the license key file in use must be named: `license.json`.

If needed, you can change the path and filename of the license key. The property is the following:

```
"license.location"
```

See *Admin Settings Page*.

Expired license

NOTE: If your license expires, you cannot use the product until a new and valid license key file has been applied. When administrators attempt to login to the application, they are automatically redirected to a location from which they can upload a new license key file.

Invalid license key file

When you start the Trifacta platform, you may see the following:

Your license key is missing or has expired. Please contact *Trifacta Support*.

Install Desktop Application

Contents:

If your environment does not support the use of Chrome, you can install the Wrangler Enterprise desktop application to provide the same access and functionality as the Trifacta® application. This desktop application connects to the enterprise Trifacta instance and provides the same capabilities without requiring a locally installed version of Chrome browser.

Wrangler Enterprise desktop application is a hybrid desktop application. Your local application instance accesses registered data files located in the datastore to which the Trifacta node is connected.

Install Process

> **NOTE:** The Wrangler Enterprise desktop application is a 64-bit Microsoft Windows application. It requires a 64-bit version of Windows to execute. The application also supports Single Sign On (SSO), if it is enabled.

Download

To begin, you must download the following Windows MSI file (`TrifactaEnterpriseSetup.msi`) from the location where your software was provided.

If you are planning to automate installation to desktops in your environment, please also download `setTrifactaServer.ps1`.

Setup

Before you begin, you should perform any necessary configuration of the Trifacta node before deploying the instances of the application. See *Configure for Desktop Application*.

Install for Windows

Steps:

1. On your Windows desktop, double-click the MSI file.
2. Follow the on-screen instructions to install the software.

Windows Command Line Installation and Configuration

As an alternative, you can perform installation and initial configuration from the command line. Download the MSI and the PS1 files to a local directory that is accessible.

> **NOTE:** For command line install, you must download from the `setTrifactaServer.ps1` from the download location.

Install software:

```
msiexec /i <path_to_TrifactaEnterpriseSetup.msi> /passive
```

Configure URL of Trifacta node:

```
setTrifactaServer.ps1 -trifactaServer <server_url> -installDir
<local_dir>
```

Parameter	Description
`trifactaServer`	(Required) URL of the server hosting the Trifacta platform. Format: `<http|https>://<host>:<port>`
`installDir`	(Optional) Specifies the installation directory in the local environment. If not specified, installation directory defaults to use the same path as the installer.

common installer parameters	This command supports the following Windows installer parameters: Verbose, Debug, ErrorAction, ErrorVariable, WarningAction, WarningVariable, OutBuffer, PipelineVariable, and OutVariable. For more information, see *about_CommonParameters* here: *http://go.microsoft.com/fwlink/?LinkID=113216*.

After this install is completed, desktop users should be able to use the application normally.

Launch the Application

Steps:

1. When installation is complete, double-click the application icon.
2. For the server, please enter the full URL including port number of the Trifacta instance to which you are connecting.

    1. By default, the server is available over port `3005`. For more information, please contact your IT administrator.
    2. If you connect to the Internet through a proxy, additional configuration is required. See *Configure Server Access through Proxy*.

    > **NOTE:** If you make a mistake in specifying the URL to the server, please uninstall and reinstall the MSI. This step clears the local application cache, and you can enter the appropriate path through the application. See *Uninstall* below.

3. When the proper URL and port number are provided, you may launch the application.
4. If your environment contains multiple server deployments, you can select the one to which to connect:

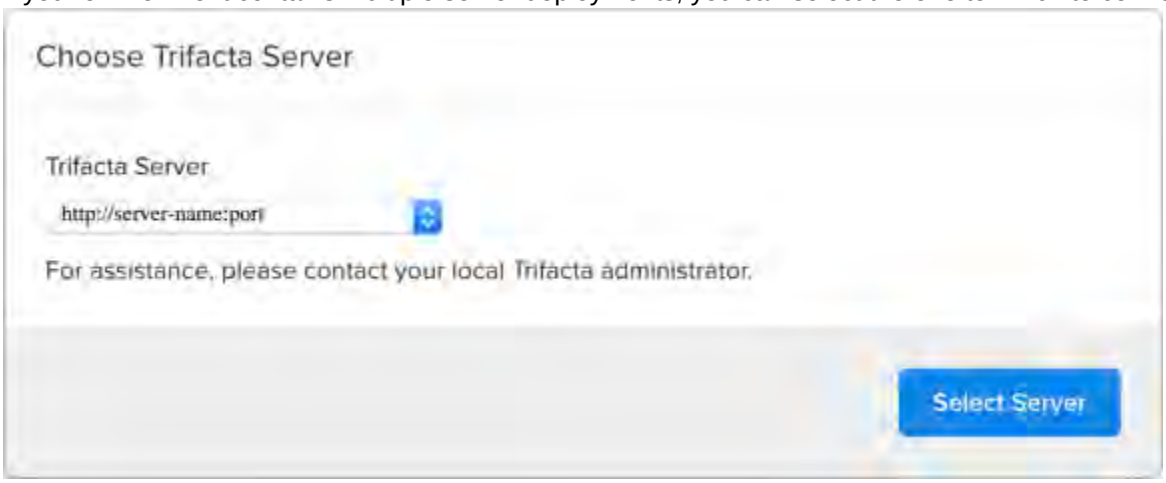


Figure: Choose Server

5. Login with your Trifacta account. See *Login*.

Documentation Note

Unless specifically noted, all features described for Trifacta Wrangler Enterprise or the Trifacta application apply to the Wrangler Enterprise desktop application.

Uninstall

To uninstall from your Windows machine, use the Add or Remove Programs control panel.

Troubleshooting

Cannot connect to server

If you are unable to connect to the server, please do the following:

1. Verify that you are connecting to the appropriate URL.
    1. If you are connecting to the incorrect URL, please uninstall the application and re-install using the MSI file. See *Uninstall* above.
2. Verify if you need to connect to the server through a proxy server. If so, additional configuration is required. See *Configure Server Access through Proxy*.
3. Check your firewall settings.

"Does Not Support Your Browser" error

This error message indicates that you are trying to connect to an instance of the server that does not support the Wrangler Enterprise desktop application. Please verify that your connection URL is pointed to a supported instance of the server.

Start and Stop the Platform

Contents:

Tip: The Restart Trifacta button in the Admin Settings page is the preferred method for restarting the platform.

NOTE: The restart button is not available when high availability is enabled for the Trifacta® node.

See *Admin Settings Page*.

Start

NOTE: These operations must be executed under the root user.

Command:

```
service trifacta start
```

Verify operations

Steps:

1. Check logs for errors:

   ```
   /opt/trifacta/logs/*.log
   ```

   1. You can also access logs through the Trifacta® application for each service. See *System Services and Logs*.
2. Login to the Trifacta application. If available, perform a simple transformation operation. See *Login*.
3. Run a simple job. See *Verify Operations*.

Restart

Command:

```
service trifacta restart
```

When the login page is available, the system has been restarted. See *Login*.

Tip: If you have made any configuration changes, you should verify operations. See *Verify Operations*.

Stop

Command:

```
service trifacta stop
```

Debugging

You can verify operations of WebHDFS. Command:

```
curl -i  "http://<hadoop_node>:<port_number>/webhdfs/v1/?
op=LISTSTATUS&user.name=trifacta"
```

Troubleshooting

Error - SequelizeConnectionRefusedError: connect ECONNREFUSED

If you have attempted to start the platform after an operating system reboot, you may receive the following error message, and the platform start fails to complete:

```
2016-10-04T14:03:17.883Z - error: [ENVIRONMENT] Environment Sanity Test
Failed
2016-10-04T14:03:17.883Z - error: [ENVIRONMENT] Exception Type: Error
2016-10-04T14:03:17.883Z - error: [ENVIRONMENT] Exception Message:
SequelizeConnectionRefusedError: connect ECONNREFUSED
```

Solution:

> **NOTE:** This solution applies to PostgreSQL 9.6 only. Please modify for your installed database version.

This error can occur when the operating system is restarted. Please execute the following commands to check the PostgreSQL configuration and restart the databases.

```
chkconfig postgresql-9.6 on
```

Then, restart the platform as normal.

```
service trifacta restart
```

Login

> **NOTE:** Administrators of the platform should change the default password for the admin account. See *Change Admin Password*.

To login to the Trifacta® application, navigate to the following in your browser:

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

where:

- `<host_name>` is the host of the Trifacta application.
- `<port_number>` is the port number to use. Default is `3005`.

If you do not have an account, click **Register**.

- If self-registration is enabled, you you may be able to immediately login after registering.
- If Kerberos or secure impersonation is enabled, an administrator must apply a Hadoop principal value to the account before you can login. Please contact your Trifacta administrator.
- System administrators can enable self-registration. See *Configure User Self-Registration*.

After you login, you are placed in the Flows page, where you can create and manage your datasets and flows. See *Flows Page*.

- If you are using S3 as your base storage layer and per-user authentication has been enabled, you must provide the AWS credentials to connect to your storage. From the left navigation bar, select **Settings > Storage** and then select the AWS option. See *Configure Your Access to S3*.
- For a basic walkthrough of the Trifacta application, see *Workflow Basics*.

To logout:

From the Settings menu, select **Logout**.

Install Reference

These appendices provide additional information during installation of Trifacta® Wrangler Enterprise.

Topics:

- *Install SSL Certificate*
- *Change Listening Port*
- *Supported Deployment Scenarios for Cloudera*
- *Supported Deployment Scenarios for Hortonworks*
- *Supported Deployment Scenarios for AWS*
- *Supported Deployment Scenarios for Azure*
- *Uninstall*

Install SSL Certificate

Contents:

You may optionally configure an SSL certificate to secure connections to the web application of the Trifacta® platform.

Pre-requisites

1. A valid SSL certificate for the FQDN where the Trifacta application is hosted
2. Root access to the Trifacta server
3. Trifacta platform is up and running

Configure nginx

There are two separate Nginx services on the server: one service for internal application use, and one service that functions as a proxy between users and the Trifacta application. To install the SSL certificate, all configuration are applied to the proxy process only.

Steps:

1. Log into the Trifacta server as the **centos** user. Switch to the **root** user:

```
sudo su
```

2. Enable the proxy nginx service so that it starts on boot:

```
systemctl enable nginx
```

3. Create a folder for the private key and limit access to it:

```
sudo mkdir /etc/ssl/private/ && sudo chmod 700 /etc/ssl/private
```

4. Copy the following files to the server. If you copy and paste the content, please ensure that you do not miss characters or insert unwanted characters.

    1. The `.key` file should go into the `/etc/ssl/private/` directory.
    2. The .crt file and the CA bundle/intermediate certificate bundle should go into the `/etc/ssl/certs/` directory.

    > **NOTE:** The delivery name and format of these files varies by provider. Please verify with your provider's documentation if this is unclear.

    3. Your certificate and the intermediate/authority certificate must be combined into one file for nginx. Here is an example of how to combine them together:

```
cat example_com.crt bundle.crt >> ssl-bundle.crt
```

5. Update the permissions on these files. Modify the following filenames as necessary:

```
sudo chmod 600 /etc/ssl/certs/ssl-bundle.crt
sudo chmod 600 /etc/ssl/private/your-private-cert.key
```

6. Use the following commands to deploy the example SSL configuration file provided on the server:

> **NOTE:** Below, some values are too long for a single line. Single lines that overflow to additional lines are marked with a `\`. The backslash should not be included if the line is used as input.

```
cp /opt/trifacta/conf/ssl-nginx.conf.sample /etc/nginx/conf.d
/trifacta.conf && \
rm /etc/nginx/conf.d/default.conf
```

7. Edit the following file:

```
/etc/nginx/conf.d/trifacta.conf
```

8. Please modify the following key directives at least:

Directive	Description
`server_name`	FQDN of the host, which must match the SSL certificate's Common Name
`ssl_certificate`	Path to the file of the certificate bundle that you created on the server. This value may not require modification.
`ssl_certificate_key`	Path to the .key file on the server.

Example file:

```
server {
  listen          443;
  ssl             on;
  server_name     EXAMPLE.CUSTOMER.COM;
  # Don't limit the size of client uploads.
  client_max_body_size 0;
  access_log      /var/log/nginx/ssl-access.log;
  error_log       /var/log/nginx/ssl-error.log;
  ssl_certificate      /etc/ssl/certs/ssl-bundle.crt;
  ssl_certificate_key  /etc/ssl/certs/EXAMPLE-NAME.key;
  ssl_protocols        SSLv3 TLSv1 TLSv1.1 TLSv1.2;
  ssl_ciphers RC4:HIGH:!aNULL:!MD5;
  ssl_prefer_server_ciphers on;
  keepalive_timeout    60;
  ssl_session_cache    shared:SSL:10m;
  ssl_session_timeout  10m;
  location / {
    proxy_pass  http://localhost:3005;
    proxy_next_upstream error timeout invalid_header http_500
http_502 http_503 http_504;
    proxy_set_header        Accept-Encoding   "";
    proxy_set_header        Host            $host;
    proxy_set_header        X-Real-IP       $remote_addr;
    proxy_set_header        X-Forwarded-For
$proxy_add_x_forwarded_for;
    proxy_set_header        X-Forwarded-Proto $scheme;
    add_header              Front-End-Https   on;
    proxy_http_version 1.1;
    proxy_set_header Upgrade $http_upgrade;
    proxy_set_header Connection "upgrade";
    proxy_set_header Host $host;
    proxy_redirect     off;
  }
  proxy_connect_timeout       6000;
  proxy_send_timeout          6000;
  proxy_read_timeout          6000;
  send_timeout                6000;
}
server {
  listen          80;
  return 301 https://$host$request_uri;
}
```

9. Save the file.
10. To apply the new configuration, start or restart the nginx service:

```
service nginx restart
```

Modify listening port for Trifacta platform

If you have changed the listening port as part of the above configuration change, then the `proxy.port` setting in Trifacta platform configuration must be updated. See *Change Listening Port*.

Add secure HTTP headers

If you have enabled SSL on the platform, you can optionally insert the following additional headers to all requests to the Trifacta node:

Header	Protocol	Required Parameters
X-XSS-Protection	HTTP and HTTPS	`proxy.securityHeaders.enabled=true`
X-Frame-Options	HTTP and HTTPS	`proxy.securityHeaders.enabled=true`
Strict-Transport-Security	HTTPS	`proxy.securityHeaders.enabled=true` and `proxy.securityHeaders.httpsHeaders=true`

NOTE: SSL must be enabled to apply these security headers.

Steps:

To add these headers to all requests, please apply the following change:

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 setting and change its value to `true`:

```
"proxy.securityHeaders.httpsHeaders": false,
```

3. Save your changes and restart the platform.

Enable secure cookies

If you have enabled SSL on the platform, you can optionally enable the use of secure cookies.

NOTE: SSL must be enabled.

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 setting and change its value to `true`:

```
"webapp.session.cookieSecureFlag": false,
```

3. Save your changes and restart the platform.

Troubleshooting

Problem - SELinux blocks proxy service from communicating with internal app service

If the Trifacta platform is installed on SELinux, the operating system blocks communications between the service that manages the proxy between users and the application and the service that manages internal application communications.

To determine if this problem is present, execute the following command:

```
sudo cat /var/log/audit/audit.log | grep nginx | grep denied
```

The problem is present if an error similar to the following is returned:

```
type=AVC msg=audit(1555533990.045:1826142): avc:  denied  { name_connect
} for  pid=25516 comm="nginx" dest=3005 scontext=system_u:system_r:
httpd_t:s0
```

For more information on this issue, see *https://www.nginx.com/blog/using-nginx-plus-with-selinux*.

Solution:

The solution is to enable the following network connection through the operating system:

```
sudo setsebool -P httpd_can_network_connect 1
```

Restart the platform.

Change Listening Port

If you need to change the listening port for the Trifacta® platform, please complete the following instructions.

> **Tip:** This change most typically applies if you are enabling use of SSL. For more information, see *Install SSL Certificate*.

> **NOTE:** By default, the platform listens on port `3005`. All client browsing devices must be configured to enable use of this port or any port number that you choose to use.

Steps:

1. Login to the Trifacta node as an admin.
2. Edit the following file:

   ```
   /opt/trifacta/conf/nginx.conf
   ```

3. Edit the following setting:

```
server {
  listen 3005;
  ...
```

4. Save the file.
5. You can apply this change through the *Admin Settings Page* (recommended) or `trifacta-conf.json`. For more information, see *Platform Configuration Methods*.
6. Locate the following setting:

```
"proxy.port": 3005,
```

7. Set this value to the same value you applied in `nginx.conf`.
8. Save your changes and restart the platform.

Supported Deployment Scenarios for AWS

Contents:

AWS Deployment Scenarios

The following are the basic AWS deployment scenarios.

Trifacta platform deployed through AWS Marketplace:

Deployment Scenario	Trifacta node installation	Base Storage Layer	Storage - S3	Storage - Redshift	Cluster	Notes

Trifacta Data Preparation for Amazon Redshift and S3 AWS install through AWS Marketplace CloudFormation template	EC2	S3	read/write	read/write	None	Trifacta Data Preparation for Amazon Redshift and S3 does not support integration with any running environment clusters. All job execution occurs on the Trifacta node in the Trifacta Photon running environment. This scenario is suitable for smaller user groups and data volumes.
Trifacta Wrangler Enterprise AWS install through AWS Marketplace CloudFormation template - with integration to EMR cluster	EC2	S3	read/write	read/write	EMR	This deployment scenario integrates by default with an EMR cluster, which is created as part of the process. It does not support integration with a Hadoop cluster.
Trifacta Wrangler Enterprise AWS install through AWS Marketplace - without integration to EMR cluster	EC2	S3	read/write	read/write	EMR	This deployment scenario assumes that the platform is to be integrated at a later time with a pre-existing EMR cluster.

Trifacta platform installed on AWS:

Deployment Scenario	Trifacta node installation	**Base Storage Layer**	**Storage - S3**	**Storage - Redshift**	**Cluster**	**Notes**
Trifacta Wrangler Enterprise AWS install with S3 read access	EC2	HDFS	read only	Not supported	EMR	When HDFS is the base storage layer, the only accessible AWS resources is read-only access to S3.
Trifacta Wrangler Enterprise AWS install with S3 read/write access	EC2	S3	read/write	read/write	EMR	

Trifacta platform installed on-premises and integrated with AWS resources:

Deployment Scenario	Trifacta node installation	**Base Storage Layer**	**Storage - S3**	**Storage - Redshift**	**Cluster**	**Notes**

Trifacta Wrangler Enterprise on-premises install with S3 read access	On-premises	HDFS	read only	Not supported	Hadoop	When HDFS is the base storage layer, the only accessible AWS resources is read-only access to S3. For more information, see *Install Software*.
Trifacta Wrangler Enterprise on-premises install with S3 read /write access	On-premises	S3	read/write	read/write	Hadoop or EMR	For more information, see *Install Software*.
Microsoft Azure						Integration with AWS-based resources is not supported. See *Install for Azure*.

Legend and Notes:

Column	Notes
Deployment Scenario	Description of the AWS-connected deployment
Trifacta node installation	Location where the Trifacta node is installed in this scenario. All AWS installations are installed on EC2 instances.
Base Storage Layer	When the Trifacta platform is first installed, the base storage layer must be set. **NOTE:** After you have begun using the product, you cannot change the base storage layer. **NOTE:** Read/write access to AWS-based resources requires that S3 be set as the base storage layer.
Storage - S3	Trifacta Wrangler Enterprise supports read access to S3 when the base storage layer is set to HDFS. For read/write access to S3, the base storage layer must be set to S3.
Storage - Redshift	For access to Redshift, the base storage layer must be set to S3.
Cluster	List of cluster types that are supported for integration and job execution at scale. • The Trifacta platform can integrate with at most one cluster. It cannot integrate with two different clusters at the same time. • Access to an EMR cluster requires S3 to be the base storage layer. • Smaller jobs can be executed on the Trifacta Photon running environment, which is hosted on the Trifacta node itself. • For more information, see *Running Environment Options*.
Notes	Any additional notes

AWS Installations

Trifacta Data Preparation for Amazon Redshift and S3 on AWS Marketplace (AMI)

Through the Amazon Marketplace, you can license and deploy an AMI of Trifacta Data Preparation for Amazon Redshift and S3, which does not require integration with a clustered running environment. All job execution happens within the AMI on the EC2 instance that you deploy. For more information, see the Trifacta Data Preparation for Amazon Redshift and S3 listing for AWS Marketplace.

- For install and configuration instructions, see *Install from AWS Marketplace*.

Trifacta Wrangler Enterprise on AWS Marketplace with EMR

You can deploy an AMI of the Trifacta platform onto an EC2 instance. For more information, see the Trifacta Wrangler Enterprise listing for AWS Marketplace.

You can deploy it in either of the following ways:

1. Auto-create a 3-node EMR cluster. For more information on installation, see *Install from AWS Marketplace with EMR*.
2. Integrate it later with your pre-existing EMR cluster.
    1. For more information on base AWS configuration, see *Configure for AWS*.
    2. For more information on configuring integration with EMR, see *Configure for EMR*.

Trifacta Wrangler Enterprise on EC2 Instance

When the Trifacta platform is installed on AWS, it is deployed on an EC2 instance. Through the EC2 console, there are a few key parameters that must be specified.

> **NOTE:** After you have created the instance, you should retain the instanceId from the console, which must be applied to the configuration in the Trifacta platform.

For more information, see *Install*.

For more information on base AWS configuration, see *Configure for AWS*.

For more information on configuring EC2, see *Configure for EC2 Role-Based Authentication*.

AWS Integrations

The following table describes the different AWS components that can host or integrate with the Trifacta platform. Combinations of one or more of these items constitute one of the deployment scenarios listed in the following section.

AWS Service	Description	Base Storage Layer	Other Required AWS Services

EC2	Amazon Elastic Compute Cloud (EC2) can be used to host the Trifacta node in a scalable cloud-based environment. The following deployments are supported: • Trifacta Wrangler Enterprise with or without access to an EMR cluster • Trifacta Data Preparation for Amazon Redshift and S3 on an AMI	Base storage layer can be S3 or HDFS. If set to HDFS, only read access to S3 is permitted.	
S3	Amazon Simple Storage Service (S3) can be used for reading data sources, writing job results, and hosting the Trifacta databases.	Base storage layer can be S3 or HDFS. If set to HDFS, only read access to S3 is permitted.	
Redshift	Amazon Redshift provides a scalable data warehouse platform, designed for big data analytics applications. The Trifacta platform can be configured to read and write from Amazon Redshift database tables.	Base Storage Layer = S3	S3
EMR	For more information on supported versions of EMR, see *Configure for EMR*.	Base Storage Layer = S3	EC2 instance
Amazon RDS	Optionally, the Trifacta databases can be installed on Amazon RDS. For more information, see *Install Databases on Amazon RDS* .	Base Storage Layer = S3	

AWS Marketplace integrations:

AWS Service	Description	Base Storage Layer	Other Required AWS Services
AMI	Through the AWS Marketplace, you can license and install an Amazon Machine Image (AMI) instance of Trifacta Data Preparation for Amazon Redshift and S3. This product is intended for smaller user groups that do not need large-scale processing of Hadoop-based clusters.	Base Storage Layer = S3 **NOTE:** HDFS is not supported.	EC2 instance
EMR	Through the AWS Marketplace, you can license and install an AMI specifically configured to work with Amazon Elastic Map Reduce (EMR), a Hadoop-based data processing platform.	Base Storage Layer = S3	AMI

Uninstall

To remove Trifacta® Wrangler Enterprise, execute as root user one of the following commands on the Trifacta node.

NOTE: All platform and cluster configuration files are preserved. User metadata is preserved in the Trifact a database.

CentOS/RHEL:

```
sudo rpm -e trifacta
```

Ubuntu:

```
sudo apt-get remove trifacta
```

Configure for AWS

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.

The Trifacta® platform can be hosted within Amazon and supports integrations with multiple services from Amazon Web Services, including combinations of services for hybrid deployments. This section provides an overview of the integration options, as well as links to related configuration topics.

For an overview of AWS deployment scenarios, see *Supported Deployment Scenarios for AWS*.

Internet Access

From AWS, the Trifacta platform requires Internet access for the following services:

NOTE: Depending on your AWS deployment, some of these services may not be required.

- AWS S3
- Key Management System [KMS] (if sse-kms server side encryption is enabled)
- Secure Token Service [STS] (if temporary credential provider is used)
- EMR (if integration with EMR cluster is enabled)

NOTE: If the Trifacta platform is hosted in a VPC where Internet access is restricted, access to S3, KMS and STS services must be provided by creating a VPC endpoint. If the platform is accessing an EMR cluster, a proxy server can be configured to provide access to the AWS ElasticMapReduce regional endpoint.

Database Installation

The following database scenarios are supported.

Database Host	Description
Cluster node	By default, the Trifacta databases are installed on PostgreSQL instances in the Trifacta node or another accessible node in the enterprise environment. For more information, see *Install Databases*.
Amazon RDS	For Amazon-based installations, you can install the Trifacta databases on PostgreSQL instances on Amazon RDS. For more information, see *Install Databases on Amazon RDS*.

Base AWS Configuration

The following configuration topics apply to AWS in general.

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

Base Storage Layer

NOTE: The base storage layer must be set during initial configuration and cannot be modified after it is set.

S3: Most of these integrations require use of S3 as the base storage layer, which means that data uploads, default location of writing results, and sample generation all occur on S3. When base storage layer is set to S3, the Trifacta platform can:

- read and write to S3
- read and write to Redshift
- connect to an EMR cluster

HDFS: In on-premises installations, it is possible to use S3 as a read-only option for a Hadoop-based cluster when the base storage layer is HDFS. You can configure the platform to read from and write to S3 buckets during job execution and sampling. For more information, see *Enable S3 Access*.

For more information on setting the base storage layer, see *Set Base Storage Layer*.

For more information, see *Storage Deployment Options*.

Configure AWS Region

For Amazon integrations, you can configure the Trifacta node to connect to Amazon datastores located in different regions.

> **NOTE:** This configuration is required under any of the following deployment conditions:
>
> 1. The Trifacta node is installed on-premises, and you are integrating with Amazon resources.
> 2. The EC2 instance hosting the Trifacta node is located in a different AWS region than your Amazon datastores.
> 3. The Trifacta node or the EC2 instance does not have access to *s3.amazonaws.com*.

1. In the AWS console, please identify the location of your datastores in other regions. For more information, see the Amazon documentation.
2. Login to the Trifacta application.
3. You can apply this change through the *Admin Settings Page* (recommended) or `trifacta-conf.json`. For more information, see *Platform Configuration Methods*.
4. Set the value of the following property to the region where your S3 datastores are located:

   ```
   aws.s3.region
   ```

   If the above value is not set, then the Trifacta platform attempts to infer the region based on default S3 bucket location.
5. Save your changes.

AWS Authentication

The following table illustrates the various methods of managing authentication between the platform and AWS. The matrix of options is basically determined by the settings for two key parameters.

- credential provider - source of credentials: platform (default), instance (EC2 instance only), or temporary
- AWS mode - the method of authentication from platform to AWS: system-wide or by-user

AWS Mode	System	User
Credential Provider		
Default	One system-wide key/secret combo is inserted in the platform for use	Each user provides key/secret combo.

	Config: `"aws.credentialProvider": "default", "aws.mode": "system", "aws.s3.key": <key>, "aws.s3.secret": <secret>,`	Config: `"aws.credentialProvider": "default", "aws.mode": "user",` User: *Configure Your Access to S3*
Instance	Platform uses EC2 instance roles.	Users provide EC2 instance roles.
	Config: `"aws.credentialProvider": "instance", "aws.mode": "system",`	Config: `"aws.credentialProvider": "instance", "aws.mode": "user",`
Temporary	Temporary credentials are issued based on per-user IAM roles.	Per-user authentication when using IAM role.
	Config: `"aws.credentialProvider": "temporary", "aws.mode": "system",`	Config: `"aws.credentialProvider": "instance", "aws.mode": "user",`

AWS Auth Mode

When connecting to AWS, the platform supports the following basic authentication modes:

Mode	Configuration	Description

system	`"aws.mode": "system",`	Access to AWS resources is managed through a single, system account. The account that you specify is based on the credential provider selected below. • The instance credential provider ignores this setting. See below.
user	`"aws.mode": "user",`	Authentication must be specified for individual users. **Tip:** In AWS user mode, Trifacta administrators can manage S3 access for users through the Admin Settings page. See *Manage Users*.

AWS Credential Provider

The Trifacta platform supports the following methods of providing credentialed access to AWS and S3 resources.

Type	Configuration	Description
default	`"aws.credentialProvider":"default",`	This method uses the provided AWS Key and Secret values to access resources. See below.
instance	`"aws.credentialProvider":"instance",`	When you are running the Trifacta platform on an EC2 instance, you can leverage your enterprise IAM roles to manage permissions on the instance for the Trifacta platform. See below.
temporary	Details are below.	

Default credential provider

Whether the AWS access mode is set to system or user, the default credential provider for AWS and S3 resources is the Trifacta platform.

Mode	Description	Configuration

`"aws.mode": "system",`	A single AWS Key and Secret is inserted into platform configuration. This account is used to access all resources and must have the appropriate permissions to do so.	`"aws.s3.key": "<your_key_value>", "aws.s3.secret": "<your_key_value>",`
`"aws.mode": "user",`	Each user must specify an AWS Key and Secret into the account to access resources.	For more information on configuring individual user accounts, see *Configure Your Access to S3*.

Default credential provider with EMR:

If you are using this method and integrating with an EMR cluster:

- Copying the custom credential JAR file must be added as a bootstrap action to the EMR cluster definition. See *Configure for EMR*.
- As an alternative to copying the JAR file, you can use the EMR EC2 instance-based roles to govern access. In this case, you must set the following parameter:

```
"aws.emr.forceInstanceRole": true,
```

For more information, see *Configure for EC2 Role-Based Authentication*.

Instance credential provider

When the platform is running on an EC2 instance, you can manage permissions through pre-defined IAM roles.

> **NOTE:** If the Trifacta platform is connected to an EMR cluster, you can force authentication to the EMR cluster to use the specified IAM instance role. See *Configure for EMR*.

For more information, see *Configure for EC2 Role-Based Authentication*.

Temporary credential provider

For even better security, you can enable use temporary credentials provided from your AWS resources based on an IAM role specified per user.

> **Tip:** This method is recommended by AWS.

Set the following properties.

Property	Description

`"aws.credentialProvider"`	• If `aws.mode` = `system`, set this value to `temporary`. • If `aws.mode` = `user` and you are using per-user authentication, then this setting is ignored and should stay as `default`.

Per-user authentication

Individual users can be configured to provide temporary credentials for access to AWS resources, which is a more secure authentication solution. For more information, see *Configure AWS Per-User Authentication*.

AWS Storage

S3 Sources

To integrate with S3, additional configuration is required. See *Enable S3 Access*.

Redshift Connections

You can create connections to one or more Redshift databases, from which you can read database sources and to which you can write job results. Samples are still generated on S3.

> **NOTE:** Relational connections require installation of an encryption key file on the Trifacta node. For more information, see *Create Encryption Key File*.

For more information, see *Create Redshift Connections*.

AWS Clusters

Trifacta Wrangler Enterprise can integrate with one instance of either of the following.

> **NOTE:** If Trifacta Wrangler Enterprise is installed through the Amazon Marketplace, only the EMR integration is supported.

EMR

When Trifacta Wrangler Enterprise in installed through AWS, you can integrate with an EMR cluster for Spark-based job execution. For more information, see *Configure for EMR*.

Hadoop

If you have installed Trifacta Wrangler Enterprise on-premises or directly into an EC2 instance, you can integrate with a Hadoop cluster for Spark-based job execution. See *Configure for Hadoop*.

Configure for EC2 Role-Based Authentication

Contents:

- *Additional AWS Configuration*
- *Use of S3 Sources*

When you are running the Trifacta platform on an EC2 instance, you can leverage your enterprise IAM roles to manage permissions on the instance for the Trifacta platform. When this type of authentication is enabled, Trifacta administrators can apply a role to the EC2 instance where the platform is running. That role's permissions apply to all users of the platform.

IAM roles

Before you begin, your IAM roles should be defined and attached to the associated EC2 instance.

> **NOTE:** The IAM instance role used for S3 access should have access to resources at the bucket level.

For more information, see *http://docs.aws.amazon.com/AWSEC2/latest/UserGuide/iam-roles-for-amazon-ec2.html*.

AWS System Mode

To enable role-based instance authentication, the following parameter must be enabled.

```
"aws.mode": "system",
```

Additional AWS Configuration

The following additional parameters must be specified:

Parameter	Description
`aws.credentialProvider`	Set this value to `instance`. IAM instance role is used for providing access.
`aws.hadoopFsUseSharedInstanceProvider`	Set this value to `true` for CDH. The class information is provided below.

Shared instance provider class information

Hortonworks:

```
"com.amazonaws.auth.InstanceProfileCredentialsProvider",
```

Pre-Cloudera 6.0.0:

```
"org.apache.hadoop.fs.s3a.SharedInstanceProfileCredentialsProvider"
```

Cloudera 6.0.0 and later:

Set the above parameters as follows:

```
"aws.credentialProvider": "instance",
"aws.hadoopFSUseSharedInstanceProvider": false,
```

Use of S3 Sources

To access S3 for storage, additional configuration for S3 may be required.

> **NOTE:** Do not configure the properties that apply to `user` mode.

Output sizing recommendations:

- Single-file output: If you are generating a single file, you should try to keep its size under 1 GB.
- Multi-part output: For multiple-file outputs, each part file should be under 1 GB in size.
- For more information, see *https://docs.aws.amazon.com/redshift/latest/dg/c_best-practices-use-multiple-files.html*

For more information, see *Enable S3 Access*.

Configure for EMR

Contents:

You can configure your instance of the Trifacta platform to integrate with Amazon Elastic MapReduce (EMR), a highly scalable Hadoop-based execution environment.

> **NOTE:** This section applies only to installations of Trifacta Wrangler Enterprise where a license key file has been acquired from Trifacta and applied to the platform.

- **Amazon EMR** (Elastic MapReduce) provides a managed Hadoop framework that makes it easy, fast, and cost-effective to process vast amounts of data across dynamically scalable Amazon EC2 instances. For more information on EMR, see *http://docs.aws.amazon.com/cli/latest/reference/emr/*.

Supported Versions

This section outlines how to create a new EMR cluster and integrate the Trifacta platform with it. The platform can be integrated with existing EMR clusters.

Supported Versions: EMR 5.6 - EMR 5.21

> **NOTE:** EMR 5.16 - EMR 5.21 requires Spark 2.4. For more information, see *Configure for Spark*.

Limitations

> **NOTE:** Job cancellation is not supported on an EMR cluster.

Create EMR Cluster

Use the following section to set up your EMR cluster for use with the Trifacta platform.

- **Via AWS EMR UI:** This method is assumed in this documentation.
- **Via AWS command line interface:** For this method, it is assumed that you know the required steps to perform the basic configuration. For custom configuration steps, additional documentation is provided below.

> **NOTE:** It is recommended that you set up your cluster for exclusive use by the Trifacta platform.

Cluster options

In the Amazon EMR console, click **Create Cluster**. Click **Go to advanced options**. Complete the sections listed below.

> **NOTE:** Please be sure to read all of the cluster options before setting up your EMR cluster.

> **NOTE:** Please perform your configuration through the Advanced Options workflow.

For more information on setting up your EMR cluster, see *http://docs.aws.amazon.com/cli/latest/reference/emr/create-cluster.html*.

Advanced Options

In the Advanced Options screen, please select the following:

- Software Configuration:
  - Release: 5.15
  - Select:
    - Hadoop 2.8.3
    - Hue 3.12.0
    - Ganglia 3.7.2

      > **Tip:** Although it is optional, Ganglia is recommended for monitoring cluster performance.

    - Spark 2.3.0

      > **NOTE:** Additional configuration is required. You must apply the Spark version number in the `spark.version` property in Admin Settings. See *Configure for Spark*.

  - Deselect everything else.
- Edit the software settings:
  - Copy and paste the following into **Enter Configuration**:

```
[
 {
 "Classification": "capacity-scheduler",
 "Properties": {
 "yarn.scheduler.capacity.resource-calculator": "org.apache.
hadoop.yarn.util.resource.DominantResourceCalculator"
 }
 }
]
```

- Auto-terminate cluster after the last step is completed: **Leave this option disabled**.

Hardware configuration

> **NOTE:** Please apply the sizing information for your EMR cluster that was recommended for you. If you have not done so, please contact your Trifacta representative.

General Options

- Cluster name: Provide a descriptive name.
- Logging: Enable logging on the cluster.
  - S3 folder: Please specify the S3 bucket and path to the logging folder.

    NOTE: Please verify that this location is read accessible to all users of the platform. See below for details.

- Debugging: Enable.
- Termination protection: Enable.
- Tags:
  - No options required.
- Additional Options:
  - EMRFS consistent view: You should enable this setting. Doing so may incur additional costs. For more information, see *EMRFS consistent view is recommended* below.
  - Custom AMI ID: None.
  - Bootstrap Actions:
    - If you are using a custom credential provider JAR, you must create a bootstrap action.

      NOTE: This configuration must be completed before you create the EMR cluster. For more information, see *Authentication* below.

Security Options

- EC2 key pair: Please select a key/pair to use if you wish to access EMR nodes via SSH.
- Permissions: Set to Custom to reduce the scope of permissions. For more information, see *EMR cluster policies* below.

  NOTE: Default permissions give access to everything in the cluster.

- Encryption Options
  - No requirements.
- EC2 Security Groups:

  - The selected security group for the master node on the cluster must allow TCP traffic from the Trifacta instance on port 8088. For more information, see *System Ports*.

Create cluster and acquire cluster ID

If you performed all of the configuration, including the sections below, you can create the cluster.

NOTE: You must acquire your EMR cluster ID for use in configuration of the Trifacta platform.

Specify cluster roles

The following cluster roles and their permissions are required. For more information on the specifics of these policies, see *EMR cluster policies*.

- **EMR Role:**
  - Read/write access to log bucket
  - Read access to resource bucket

- **EC2 instance profile:**
  - If using instance mode:
    - EC2 profile should have read/write access for all users.
    - EC2 profile should have same permissions as EC2 Edge node role.
  - Read/write access to log bucket
  - Read access to resource bucket
- **Auto-scaling role:**
  - Read/write access to log bucket
  - Read access to resource bucket
  - Standard auto-scaling permissions

Authentication

You can use one of two methods for authenticating the EMR cluster:

- **Role-based IAM authentication (recommended):** This method leverages your IAM roles on the EC2 instance.
- **Custom credential provider JAR file:** This method utilizes a JAR file provided with the platform. This JAR file must be deployed to all nodes on the EMR cluster through a bootstrap action script.

Role-based IAM authentication

You can leverage your IAM roles to provide role-based authentication to the S3 buckets.

> **NOTE:** The IAM role that is assigned to the EMR cluster and to the EC2 instances on the cluster must have access to the data of all users on S3.

For more information, see *Configure for EC2 Role-Based Authentication*.

Specify the custom credential provider JAR file

If you are not using IAM roles for access, you can manage access using either of the following:

- AWS key and secret values specified in `trifacta-conf.json`
- AWS user mode

In either scenario, you must use the custom credential provider JAR provided in the installation. This JAR file must be available to all nodes of the EMR cluster.

After you have installed the platform and configured the S3 buckets, please complete the following steps to deploy this JAR file.

> **NOTE:** These steps must be completed before you create the EMR cluster.

> **NOTE:** This section applies if you are using the default credential provider mechanism for AWS and are not using the IAM instance-based role authentication mechanism.

Steps:

1. From the installation of the Trifacta platform, retrieve the following file:

```
[TRIFACTA_INSTALL_DIR]/aws/credential-provider/build/libs/trifacta-
aws-emr-credential-provider.jar
```

2. Upload this JAR file to an S3 bucket location where the EMR cluster can access it:
    1. **Via AWS Console S3 UI:** See *http://docs.aws.amazon.com/cli/latest/reference/s3/index.html*.
    2. **Via AWS command line:**

```
aws s3 cp trifacta-aws-emr-credential-provider.jar s3://<YOUR-
BUCKET>/
```

3. Create a bootstrap action script named `configure_emrfs_lib.sh`. The contents must be the following:

```
sudo aws s3 cp s3://<YOUR-BUCKET>/trifacta-aws-emr-credential-
provider.jar  /usr/share/aws/emr/emrfs/auxlib/
```

4. This script must be uploaded into S3 in a location that can be accessed from the EMR cluster. Retain the full path to this location.
5. Add bootstrap action to EMR cluster configuration.
    1. **Via AWS Console S3 UI:** Create the bootstrap action to point to the script you uploaded on S3.
    2. Via AWS command line:
        1. Upload the `configure_emrfs_lib.sh` file to the accessible S3 bucket.
        2. In the command line cluster creation script, add a custom bootstrap action, such as the following:

```
--bootstrap-actions '[
{"Path":"s3://<YOUR-BUCKET>/configure_emrfs_lib.sh","
Name":"Custom action"}
]'
```

When the EMR cluster is launched with the above custom bootstrap action, the cluster does one of the following:

- Interacts with S3 using the credentials specified in `trifacta-conf.json`
- if `aws.mode` = `user`, then the credentials registered by the user are used.

For more information about AWSCredentialsProvider for EMRFS please see:

- *http://docs.aws.amazon.com/emr/latest/ManagementGuide/emr-plan-credentialsprovider.html*
- *https://aws.amazon.com/blogs/big-data/securely-analyze-data-from-another-aws-account-with-emrfs/*

EMRFS consistent view is recommended

Although it is not required, you should enable the consistent view feature for EMRFS on your cluster.

During job execution, including profiling jobs, on EMR, the Trifacta platform writes files in rapid succession, and these files are quickly read back from storage for further processing. However, Amazon S3 does not provide a guarantee of a consistent file listing until a later time.

To ensure that the Trifacta platform does not begin reading back an incomplete set of files, you should enable EMRFS consistent view.

> **NOTE:** If EMRFS consistent view is enabled, additional policies must be added for users and the EMR cluster. Details are below.

> **NOTE:** If EMRFS consistent view is not enabled, profiling jobs may not get a consistent set of files at the time of execution. Jobs can fail or generate inconsistent results.

For more information on EMRFS consistent view, see *http://docs.aws.amazon.com/emr/latest/ManagementGuide/emr-plan-consistent-view.html*.

DynamoDB

Amazon's DynamoDB is automatically enabled to store metadata for EMRFS consistent view.

> **NOTE:** DynamoDB incurs costs while it is in use. For more information, see *https://aws.amazon.com/dynamodb/pricing/*.

> **NOTE:** DynamoDB does not automatically purge metadata after a job completes. You should configure periodic purges of the database during off-peak hours.

Enable output job manifest

When EMRFS consistent view is enabled on the cluster, the platform must be configured to use it. During job execution, the platform can use consistent view to create a manifest file of all files generated during job execution. When the job results are published to an external target, this manifest file ensures proper publication.

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 set it to `true`:

```
"feature.enableJobOutputManifest": true,
```

3. Save your changes and restart the platform.

Set up S3 Buckets

Bucket setup

You must set up S3 buckets for read and write access.

> **NOTE:** Within the Trifacta platform, you must enable use of S3 as the default storage layer. This configuration is described later.

For more information, see *Enable S3 Access*.

Set up EMR resources buckets

On the EMR cluster, all users of the platform must have access to the following locations:

Location	Description	Required Access
EMR Resources bucket and path	The S3 bucket and path where resources can be stored by the Trifacta platform for execution of Spark jobs on the cluster. **NOTE:** If server-side encryption is in use, only SSE-S3 encryption type is supported for the resources bucket. If you are using the same bucket for resources and data and SSE-KMS is in use, you may need to deploy a second bucket for EMR resources. For more information on server-side encryption, see *Enable S3 Access*. The locations are configured separately in the Trifacta platform.	Read/Write
EMR Logs bucket and path	The S3 bucket and path where logs are written for cluster job execution.	Read

These locations are configured on the Trifacta platform later.

Access Policies

EC2 instance profile

Trifacta users require the following policies to run jobs on the EMR cluster:

```
{
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "elasticmapreduce:AddJobFlowSteps",
                "elasticmapreduce:DescribeStep",
                "elasticmapreduce:DescribeCluster",
                "elasticmapreduce:ListInstanceGroups"
            ],
            "Resource": [
                "*"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "s3:*"
            ],
            "Resource": [
                "arn:aws:s3:::__EMR_LOG_BUCKET__",
                "arn:aws:s3:::__EMR_LOG_BUCKET__/*",
                "arn:aws:s3:::__EMR_RESOURCE_BUCKET__",
                "arn:aws:s3:::__EMR_RESOURCE_BUCKET__/*"
            ]
        }

    ]
}
```

EMR roles

The following policies should be assigned to the EMR roles listed below for read/write access:

```
{
            "Effect": "Allow",
            "Action": [
                "s3:*"
            ],
            "Resource": [
                "arn:aws:s3:::__EMR_LOG_BUCKET__",
                "arn:aws:s3:::__EMR_LOG_BUCKET__/*",
                "arn:aws:s3:::__EMR_RESOURCE_BUCKET__",
                "arn:aws:s3:::__EMR_RESOURCE_BUCKET__/*"
            ]
        }
}
```

EMRFS consistent view policies

If EMRFS consistent view is enabled, the following policy must be added for users and the EMR cluster permissions:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Action": [
        "dynamodb:*"
       ],
      "Effect": "Allow",
      "Resource": [
        "*"
      ]
    }
  ]
}
```

Configure Trifacta platform for EMR

Please complete the following sections to configure the Trifacta platform to communicate with the EMR cluster.

Change admin password

As soon as you have installed the software, you should login to the application and change the admin password. The initial admin password is the instanceId for the EC2 instance. For more information, see *Change Password*.

Verify S3 as base storage layer

EMR integrations requires use of S3 as the base storage layer.

> **NOTE:** The base storage layer must be set during initial installation and set up of the Trifacta node.

See *Set Base Storage Layer*.

Set up S3 integration

To integrate with S3, additional configuration is required. See *Enable S3 Access*.

Enable EMR integration

After you have configured S3 to be the base storage layer, you must enable EMR integration.

Steps:

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

1. Search for the following setting:

```
"webapp.runInEMR": false,
```

2. Set the above value to `true`.
3. Set the following value to `false`:

```
"webapp.runInHadoop": false,
```

4. Verify the following property values:

```
"webapp.runInTrifactaServer": true,
"webapp.runInEMR": true,
"webapp.runInHadoop": false,
"webapp.runInDataflow": false,
"photon.enabled": true,
```

Apply EMR cluster ID

The Trifacta platform must be aware of the EMR cluster to which to connection.

Steps:

1. Administrators can apply this configuration change through the *Admin Settings Page* in the application. If the application is not available, the settings are available in `trifacta-conf.json`. For more information, see *Platform Configuration Methods*.
2. Under External Service Settings, enter your AWS EMR Cluster ID. Click the Save button below the textbox.

For more information, see *Admin Settings Page*.

Extract IP address of master node in private sub-net

If you have deployed your EMR cluster on a private sub-net that is accessible outside of AWS, you must enable this property, which permits the extraction of the IP address of the master cluster node through DNS.

NOTE: This feature must be enabled if your EMR is accessible outside of AWS on a private network.

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. Set the following property to `true`:

```
"emr.extractIPFromDNS": false,
```

3. Save your changes and restart the platform.

EMR Authentication for the Trifacta platform

Depending on the authentication method you used, you must set 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*.

Authentication method	Properties and values
Use default credential provider for all Trifacta access including EMR. **NOTE:** This method requires the deployment of a custom credential provider JAR.	`"aws.credentialProvider":" default", "aws.emr.forceInstanceRole": false,`
Use default credential provider for all Trifacta access. However, EC 2 role-based IAM authentication is used for EMR.	`"aws.credentialProvider":" default", "aws.emr.forceInstanceRole": true,`
EC2 role-based IAM authentication for all Trifacta access	`"aws.credentialProvider":" instance",`

Configure Spark for EMR

For EMR, you can configure a set of Spark-related properties to manage the integration and its performance.

Configure Spark version

Depending on the version of EMR with which you are integrating, the Trifacta platform must be modified to use the appropriate version of Spark to connect to EMR. For more information, see *Configure for Spark*.

Specify YARN queue for Spark jobs

Through the Admin Settings page, you can specify the YARN queue to which to submit your Spark jobs. All Spark jobs from the Trifacta platform are submitted to this queue.

Steps:

1. In platform configuration, locate the following:

   ```
   "spark.props.spark.yarn.queue"
   ```

2. Specify the name of the queue.
3. Save your changes.

Allocation properties

The following properties must be passed from the Trifacta platform to Spark for proper execution on the EMR cluster.

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*.

> **NOTE:** Do not modify these properties through the Admin Settings page. These properties must be added as extra properties through the Spark configuration block. Ignore any references in `trifacta-conf.json` to these properties and their settings.

```
"spark": {
  ...
  "props": {
    "spark.dynamicAllocation.enabled": "true",
    "spark.shuffle.service.enabled": "true",
    "spark.executor.instances": "0",
    "spark.executor.memory": "2048M",
    "spark.executor.cores": "2",
    "spark.driver.maxResultSize": "0"
  }
  ...
}
```

Property	Description	Value
`spark.dynamicAllocation.enabled`	Enable dynamic allocation on the Spark cluster, which allows Spark to dynamically adjust the number of executors.	`true`
`spark.shuffle.service.enabled`	Enable Spark shuffle service, which manages the shuffle data for jobs, instead of the executors.	`true`
`spark.executor.instances`	Default count of executor instances.	See Sizing Guide.
`spark.executor.memory`	Default memory allocation of executor instances.	See Sizing Guide.
`spark.executor.cores`	Default count of executor cores.	See Sizing Guide.
`spark.driver.maxResultSize`	Enable serialized results of unlimited size by setting this parameter to zero (0).	`0`

Default Hadoop job results format

For smaller datasets, the platform recommends using the Trifacta Photon running environment.

For larger datasets, if the size information is unavailable, the platform recommends by default that you run the job on the Hadoop cluster. For these jobs, the default publishing action for the job is specified to run on the Hadoop cluster, generating the output format defined by this parameter. Publishing actions, including output format, can always be changed as part of the job specification.

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

```
"webapp.defaultHadoopFileFormat": "csv",
```

Accepted values: `csv`, `json`, `avro`, `pqt`

For more information, see *Run Job Page*.

Additional configuration for EMR

You can set the following parameters as needed:

Steps:

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

Property	Required	Description
aws.emr.resource.bucket	Y	S3 bucket name where Trifacta executables, libraries, and other resources can be stored that are required for Spark execution. **NOTE:** If server-side encryption is in use, only SSE-S3 encryption type is supported for the resources bucket. If you are using the same bucket for resources and data and SSE-KMS is in use, you may need to deploy a second bucket for EMR resources. For more information on server-side encryption, see *Enable S3 Access*.
aws.emr.resource.path	Y	S3 path within the bucket where resources can be stored for job execution on the EMR cluster. **NOTE:** Do not include leading or trailing slashes for the path value.
aws.emr.proxyUser	Y	This value defines the user for the Trifacta users to use for connecting to the cluster. **NOTE:** Do not modify this value.
aws.emr.maxLogPollingRetries	N	Configure maximum number of retries when polling for log files from EMR after job success or failure. Minimum value is 5.

aws.emr.tempfilesCleanupAge	N	Defines the number of days that temporary files in the `/trifacta/tempfiles` directory on EMR HDFS are permitted to age. By default, this value is set to `0`, which means that batch job files are cleaned up after every job run. If needed, you can set this to a positive integer value. During each job run, the platform scans this directory for temp files older than the specified number of days and removes any that are found. This cleanup provides an additional level of system hygiene. Before enabling this secondary cleanup process, please execute the following command to clear the `tempfiles` directory: `hdfs dfs -rm -r -skipTrash /trifacta/tempfiles`

Optional Configuration

Configure for Redshift

For more information on configuring the platform to integrate with Redshift, see *Create Redshift Connections*.

Switch EMR Cluster

If needed, you can switch to a different EMR cluster through the application. For example, if the original cluster suffers a prolonged outage, you can switch clusters by entering the cluster ID of a new cluster. For more information, see *Admin Settings Page*.

Configure Batch Job Runner

Batch Job Runner manages jobs executed on the EMR cluster. You can modify aspects of how jobs are executed and how logs are collected. For more information, see *Configure Batch Job Runner*.

Modify Job Tag Prefix

In environments where the EMR cluster is shared with other job-executing applications, you can review and specify the job tag prefix, which is prepended to job identifiers to avoid conflicts with other applications.

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 and modify if needed:

```
"aws.emr.jobTagPrefix": "TRIFACTA_JOB_",
```

3. Save your changes and restart the platform.

Testing

1. Load a dataset from the EMR cluster.
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.

Enable AWS Glue Access

Contents:

If you have integrated with an EMR cluster version 5.8.0 or later, you can configure your Hive instance to use AWS Glue Data Catalog for storage and access to Hive metadata.

> **Tip:** For metastores that are used across a set of services, accounts, and applications, AWS Glue is the recommended method of access.

For more information on AWS Glue, see *https://docs.aws.amazon.com/emr/latest/ReleaseGuide/emr-hive-metastore-glue.html*.

This section describes how to enable integration with your AWS Glue deployment.

Supported Deployment

AWS Glue tables can be read under the following conditions:

- The Trifacta platform uses S3 as the base storage layer.
- The Trifacta platform is integrated with an EMR cluster:
    - EMR version 5.8.0 or later
    - EMR cluster has been configured with HiveServer2
- The Hive deployment must be integrated with AWS Glue.

> **NOTE:** Hive connections are supported when S3 is the backend datastore.

- For HiveServer2 connectivity, the Trifacta node has direct access to the Master node of the EMR cluster.

EMR Settings

When you create the EMR cluster, please verify the following in the AWS Glue Data Catalog settings:

- **`Use for Hive table metadata`**
- **`Use for Spark table metadata`**

Deploy Credentials JAR to S3

To enable integration between the Trifacta platform and AWS Glue, a JAR file for managing the Trifacta credentials for AWS access must be deployed to S3 in a location that is accessible to the EMR cluster.

When the EMR cluster is launched with the followng custom bootstrap action, the cluster does one of the following:

- Interacts with AWS Glue using the credentials specified in `trifacta-conf.json`
- If `aws.mode` = `user`, then the credentials registered by the user are used to connect to AWS Glue.

Steps:

1. From the installation of the Trifacta platform, retrieve the following file:

```
[TRIFACTA_INSTALL_DIR]/aws/glue-credential-provider/build/libs
/trifacta-aws-glue-credential-provider.jar
```

2. Upload this JAR file to an S3 bucket location where the EMR cluster can access it:
    1. **Via AWS Console S3 UI:** See *http://docs.aws.amazon.com/cli/latest/reference/s3/index.html*.
    2. **Via AWS command line:**

```
aws s3 cp trifacta-aws-glue-credential-provider.jar s3://<YOUR-
BUCKET>/
```

3. Create a bootstrap action script named `configure_glue_lib.sh`. The contents must be the following:

```
sudo aws s3 cp s3://<YOUR-BUCKET>/trifacta-aws-glue-credential-
provider.jar  /usr/share/aws/emr/emrfs/auxlib/
sudo aws s3 cp s3://<YOUR-BUCKET>/trifacta-aws-glue-credential-
provider.jar  /usr/lib/hive/auxlib/
```

4. This script must be uploaded into S3 in a location that can be accessed from the EMR cluster. Retain the full path to this location.
5. Add a bootstrap action to EMR cluster configuration.
    1. **Via AWS Console S3 UI:** Create the bootstrap action to point to the script that you uploaded on S3.
    2. **Via AWS command line:**
        1. Upload the `configure_glue_lib.sh` file to the accessible S3 bucket.
        2. In the command line cluster creation script, add a custom bootstrap action. Example:

```
--bootstrap-actions '[
{"Path":"s3://<YOUR-BUCKET>/configure_glue_lib.sh","
Name":"Custom action"}
]'
```

Authentication

Authentication methods are required permissions are based on the AWS authentication mode:

```
"aws.mode": "system",
```

aws.mode value	Permissions	Doc
`system`	IAM role assigned to the cluster must provide access to AWS Glue.	See *Configure for AWS*.
`user`	The user role must provide access to AWS Glue.	Se below for an example fine-grained access control. See *Configure AWS Per-User Authentication*.

Example fine-grain access control for IAM policy:

If you are using IAM roles to provide access to AWS Glue, you can review the following fine-grained access control, which includes the permissions required to access AWS Glue tables. Please add this to the Permissions section of your AWS Glue Catalog Settings page.

NOTE: Please verify that access is granted in the IAM policy to the default database for AWS Glue, as noted below.

```
{
    "Sid" : "accessToAllTables",
    "Effect" : "Allow",
    "Principal" : {
      "AWS" : [  "arn:aws:iam::<accountId>:role/glue-read-all" ]
    },
    "Action" : [ "glue:GetDatabases", "glue:GetDatabase", "glue:
GetTables", "glue:GetTable", "glue:GetUserDefinedFunctions", "glue:
GetPartitions" ],
    "Resource" : [ "arn:aws:glue:us-west-2:<accountId>:catalog", "arn:
aws:glue:us-west-2:<accountId>:database/default", "arn:aws:glue:us-west-
2:<accountId>:database/global_temp", "arn:aws:glue:us-west-2:<accountId>:
database/mydb", "arn:aws:glue:us-west-2:<accountId>:table/mydb/*" ]
}
```

Limitations

- Access is read-only. Publishing to Glue hosted on EMR is not supported.

Enable

Please verify the following have been enabled and configured.

1. Your deployment has been configured to meet the Supported Deployment guidelines above.
2. You must integrate the platform with Hive.

   > **NOTE:** For the Hive hostname and port number, use the Master public DNS values. For more information, see *https://docs.aws.amazon.com/emr/latest/ReleaseGuide/emr-hive-metastore-glue.html*.

   For more information, see *Configure for Hive*.
3. If you are using it, the custom SQL query feature must be enabled. For more information, see *Enable Custom SQL Query*.

Create Connection

You can create one or more connections to databases in your AWS Glue deployment.

Key fields:

Field	Description
EMR Master Node DNS	This DNS value can be retrieved from the EMR console.
Port	The port number through which to connect to the DNS master node
Connection String Options	No values need to be provided here.

- See *Create Connection Window*.
- See *Connections Page*.

Use

After the integration has been made between the platform and AWS Glue, you can import datasets.

- Browse for datasets through AWS Glue. See *AWS Glue Browser*.
- Import using custom SQL queries. For more information, see *Create Dataset with SQL*.

Configure AWS Per-User Authentication

Contents:

For Trifacta® Wrangler Enterprise, you can configure AWS authentication on a per-user basis, using temporary credentials for superior security.

Enable

The following parameters must be set:

Property	Description
`"aws.readFromConfigurationService" : false,`	Set this value to `false` for Trifacta Wrangler Enterprise, which prevents the product from retrieving AWS-related configuration information from the incorrect source.
`"aws.mode": "user",`	Each user can specify credentials.

To authenticate to AWS services from the Trifacta platform using an IAM role:

Property	Description

`"aws.ec2InstanceRoleForAssumeRole" : true,`	• If `true`, then all users use the EC2 instance role for authenticating to the AWS STS service for their temporary credentials.
	
	NOTE: You must ensure that the role provides adequate access to STS. Details are below.
	
	Tip: This method is recommended.
	
	• If `false`, then a system-wide set of AWS key/secret credentials must be inserted into platform configuration in the Admin Settings page as the master set of credentials to access STS for temporary credentials:
	
	Properties to set:
	
	`"aws.s3.key"`
	`"aws.s3.secret"`
	
	NOTE: After specifying the above key/secret combination, you can skip to the User Access section below.

Configure Per-User Authentication using IAM Role

Please complete the following general steps.

Steps:

1. Instance role: Create an IAM role and link it to the EC2 instance where the Trifacta node is hosted. Include the following trust relationship:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Service": "ec2.amazonaws.com"
      },
      "Action": "sts:AssumeRole"
    }
  ]
}
```

2. User role: Create another IAM role and provides required access to the S3 buckets. Example:

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "VisualEditor0",
            "Effect": "Allow",
            "Action": [
                "s3:PutObject",
                "s3:GetObject",
                "s3:ListBucket"
            ],
            "Resource": [
                "arn:aws:s3:::<my_s3_bucket>/trifacta/*",
                "arn:aws:s3:::<my_s3_bucket>"
            ]
        }
    ]
}
```

   where:
   `<my_s3_bucket>` is the name of your bucket.
3. Under the user role definition, edit the Trust relationship. Add the instance role to Principal:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "AWS": [
        "arn:aws:iam::       {awsAccountId}:role/{instanceRole}"
      },
      "Action": "sts:AssumeRole"
    }
  ]
}
```

    1. For more information, see *Insert Trust Relationship in AWS IAM Role*.
    2. For more granular control over the Trust relationship, see *https://docs.aws.amazon.com/IAM/latest/UserGuide/reference_policies_elements_principal.html*.
4. AWS Glue: If you are integrating with AWS Glue, additional permissions must be set. For more information, see *Enable AWS Glue Access*.
5. Log in the Trifacta platform as a Trifacta admin.
6. Click the link to specify storage settings. Populate the values for:
    1. IAM role
    2. Role ARN
    3. S3 Bucket Name
7. Save your changes.

User Access

After per-user authentication has been enabled, each user must provide or be provided the credentials and S3 bucket to use. Users can insert a default S3 bucket and credentials to use in their profiles. See *Configure Your Access to S3*.

Configure for AWS SAML Passthrough Authentication

Contents:

Optionally for single sign-on, the Trifacta® platform can leverage the AWS user/role mappings that are managed by a SAML authentication provider. In this authentication scenario:

- The Trifacta platform uses its native SAML support for SSO authentication.
- Access to AWS resources is governed by the set of permissions and IAM roles that are managed by your AWS admins. The Trifacta platform does not allow editing of the list of available IAM roles for use.

- Authentication to AWS is governed by a third-party SAML provider, which has access to this set of IAM roles and underlying permissions.
- Users of the Trifacta platform are mapped to one or more IAM roles. These IAM roles can be selected at the workspace (admin) or individual user level.

Usage:

When this feature is enabled, a user's available IAM roles are automatically synched via SAML. When a user signs in to the Trifacta application, the user can select their default role to use.

Pre-requisites

- Per-user authentication to AWS has been enabled. For more information, see *Configure AWS Per-User Authentication*.
- This feature is supported only for the SAML authentication method of SSO authentication native to the Trifacta platform. It is not supported for any other SSO auth method. For more information, see *Configure SSO for SAML*.
- AWS permissions must be defined via IAM role and made available to an identity provider that adheres to SAML standards. The SAML identity provider must be configured with a list of SAML assertions containing the IAM roles that an external user may assume.

NOTE: When this feature is enabled and the platform is restarted, users of the Trifacta platform cannot authenticate to AWS resources until IAM roles have been assigned to their accounts. Where possible, you should enable this feature on an unused instance of the platform.

Enable

To enable, the following configuration change must be applied.

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 set it to `true`:

```
"feature.importAwsRoles.saml.enabled": true,
```

3. Save your changes and restart the platform.

Configure

After the feature has been assigned, roles must be assigned to users.

List of Roles

The list of available roles is passed from the SAML identity provider to the Trifacta platform. From this list of roles, each user can select the one to apply to the account.

Per-User Assignments

Individual users must select the IAM role ARN to assume from the list exposed by Trifacta administrator.

NOTE: Before a user is permitted to complete login to the application, the user must select a role from the provided list.

For more information, see *Configure Your Access to S3*.

Assignment per API

You can use the platform APIs to create platform AWS roles and assign them to users. For more information, see *API Workflow - Manage AWS Configurations*.