TRIFACTA

Quick Install for Amazon EMR

Version: 5.0
Doc Build Date: 09/24/2018

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

Quick Install - Amazon EMR

Contents:

Scenario Description

This scenario assumes the following about the Trifacta® platform deployment:

- The platform is to be installed via an Amazon AMI onto an EC2 instance.
- It is to be connected to an EMR cluster.
- No security features are applied to the platform and its use of the datastore.
- You have acquired a Trifacta license key. The license key must be deployed to the Trifacta node before you start the platform.

> **NOTE:** This scenario does not provide information on installing and configuring optional components, including security features. It is intended to get the Trifacta platform installed, operational, and connected to the EMR cluster.

Pre-requisites

If you are integrating the Trifacta platform with an EMR cluster, you must acquire a license first. Additional configuration is required. For more information, please contact aws-marketplace@trifacta.com.

Before you begin:

1. **Read:** Please read this entire document before you create the EMR cluster or install theTrifacta platform.
2. **Cluster sizing:** Before you begin, you should allocate sufficient resources for sizing the EMR cluster. For guidance, please contact your Trifacta representative.

Product Limitations

- The EC2 instance, S3 buckets, and any connected Redshift databases must be located in the same Amazon region. Cross-region integrations are not supported at this time.
- No support for Hive integration
- No support for secure impersonation or Kerberos
- 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.

Install

NOTE: Before you install, you should review the configuration content for specific instructions on setting up the Trifacta node. See below.

1. Create the EC2 instance for the Trifacta platform.
2. Download and deploy the AMI into the EC2 instance.

Desktop Requirements

- All desktop users of the platform must have the latest version of Google Chrome installed on their desktops.
  - Google Chrome must have the PNaCl client installed and enabled.
  - PNaCl Version: `0.50.x.y` or later
- All desktop users must be able to connect to the EC2 instance through the enterprise infrastructure.

Pre-requisites

Before you install the platform, please verify that the following steps have been completed.

1. **EULA.** Before you begin, please review the End-User License Agreement. See *https://docs.trifacta.com/display/PUB/End-User+License+Agreement+-+Trifacta+Wrangler+Enterprise*.
2. **S3 bucket.** Please create an S3 bucket to storeTrifacta assets. In the bucket, the platform stores metadata in the following location:

   ```
   <S3_bucket_name>/trifacta
   ```

   See *https://s3.console.aws.amazon.com/s3/home*.
3. **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*

4. **EC2 instance role.** Create an EC2 instance role for this policy. See*https://console.aws.amazon.com/iam/home#/roles*.

Install Steps

1. Launch the product.
2. In the EC2 Console:
    a. **Instance size:** Select the instance size.
    b. **Network:** Configure the VPC, subnet, firewall and other configuration settings necessary to communicate with the instance.
    c. **Auto-assigned Public IP:** You must create a public IP to access the Trifacta platform.
    d. **EC2 role:** Select the EC2 role that you created.
    e. **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.

    f. **Security group:** Use a security group that exposes access to port 3005, which is the default port for the platform.
    g. **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.
    h. Save your changes.
3. Apply license key:
    a. Acquire the `license.json` license key file that was provided to you by your Trifacta representative.
    b. Transfer the license key file to the EC2 node that is hosting the Trifacta platform. Navigate to the directory where you stored it.
    c. Make the Trifacta user the owner of the file:

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

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

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

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

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

4. Launch the configured platform.

    > **NOTE:** From the EC2 Console, please acquire the `instanceId`, which is needed in a later step.

5. 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
    ```

6. When the login screen appears, enter the following:
    a. Username: `admin@trifacta.local`
    b. Password: (the `instanceId` value)

    > **NOTE:** As soon as you login as an admin for the first time, you should immediately change the password. Select the User Profile menu item in the upper-right corner. Change the password and click **Save** to restart the platform.

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

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

b. Set the value of this setting to be the bucket that you created.

9. 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 to access the EMR cluster.

Details on the above configuration are described later.

10. Click **Save**.
11. When the platform restarts, you can begin using the product.

SSH Access

If you need to SSH to the Trifacta node, you can use the following command:

```
ssh -i <path_to_key_file> <userId>@<tri_node_DNS_or_IP>
```

Parameter	Description
`<path_to_key_file>`	Path to the key file stored on your local computer.
`<userId>`	The user ID is always `centos`.
`<tri_node_DNS_or_IP>`	DNS or IP address of the Trifacta node

If you are integrating with an EMR cluster, additional configuration is required.

> **NOTE:** Please review these steps with your Trifacta representative.

Set up 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: EMR 5.6 - 5.12
  - Select:
    - Hadoop 2.7.3
    - Hue 3.12.0
    - Ganglia 3.7.2
    - Spark:
      - For EMR 5.6 - EMR 5.7: Spark 2.1.1
      - For EMR 5.8 - EMR 5.12.1: Spark 2.2.x

      NOTE: You must apply the Spark version number in the `spark.version` property in Admin Settings. Additional configuration is required. 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: **Disable this option**.

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.
- Scale down behavior: Terminate at instance hour.
- Tags:
    - No optiions 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 the default credential provider, 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 traffic 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/emr/build/libs/trifacta-aws-emr-credential
-provider[TIMESTAMP].jar
```

> **NOTE:** Do not remove the timestamp value from the filename. This information is useful for support purposes.

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

```
aws s3 cp trifacta-aws-emr-credential-provider[TIMESTAMP].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[TIMESTAMP].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.
    a. **Via AWS Console S3 UI:** Create the bootstrap action to point to the script you uploaded on S3.

    b. Via AWS command line:
        i. Upload the`configure_emrfs_lib.sh` file to the accessible S3 bucket.
        ii. 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.

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/path	The S3 bucket and path where resources can be stored by the Trifacta platform for execution of Spark jobs on the cluster.	Read/Write
EMR Logs bucket/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, EC2 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. For more information on how Spark is implemented in the platform, 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 Server.

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 EMR configuration

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.path	Y	S3 path 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.resource.bucket	Y	S3 bucket where Trifacta executables, libraries, and other resources can be stored that are required for Spark execution.
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.maxJobTimeoutMillis	N	Defines the timeout for EMR jobs in milliseconds. By default, this value is set to `-1`, which allows jobs to run for an infinite length of time. **NOTE:** This setting should be modified only if you are experiencing problems with jobs hanging during execution on the EMR cluster.

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.

Configure for EC2 Role-Based Authentication

This configuration is optional.

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 applya 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. 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 5.11 and later. The class information is provided below.
	Hortonworks and CDH 5.11 and earlier:
	`"com.amazonaws.auth.InstanceProfileCredentialsPro`
	CDH 5.11 and later:
	`"org.apache.hadoop.fs.s3a.SharedInstanceProfileCr`
	In the future:
	CDH is moving back to using the `Instance` class in a future release. For details, see *https://issues.apache.org/jira/browse/HADOOP-14301*.

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.

See *Enable S3 Access*.

Start and Stop the Platform

Use the following command line commands to start, stop, and restart the platform.

Start:

```
sudo service trifacta start
```

Stop:

```
sudo service trifacta stop
```

Restart:

```
sudo service trifacta restart
```

For more information, see *Install Start Platform*.

Verify

After you have installed or made changes to the platform, you should verify operations with the Hadoop cluster.

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

> **NOTE:** These steps verify operations with data sourced from HDFS. If you are verifying operations for other datastores, see *Verify Operations*.

Steps:

1. Login to the application. See *Login*.
2. In the application toolbar, click **Datasets**. Click **Import Dataset**. Click **HDFS**.
3. Navigate your HDFS directory structure to locate a small CSV or JSON file.
4. Select the file. In the right panel, click **Create and Transform**.
    a. **Troubleshooting:** If the steps so far work, then you have read access to HDFS from the platform. If not, please check permissions for the Trifacta user on the Hadoop cluster and its access to the appropriate directories.
    b. See *Import Data Page*.
5. In the Transformer page, some steps have already been added to your recipe, so you can run the job right away. Click **Run Job.**
    a. See *Transformer Page*.
6. In the Run Job Page:
    a. For Running Environment, some of these options may not be available. Choose according to the running environment you wish to test.
        i. **Trifacta Server:** Runs job on the Trifacta node.
        ii. **Hadoop:** Runs the job on the Hadoop cluster with which the product is integrated.
        iii. **Hadoop on EMR:** If the platform is integrated with an Amazon EMR cluster, you can test EMR jobs by selecting this option.
    b. Select CSV and JSON output.
    c. Select the Profile Results checkbox.
    d. **Troubleshooting:** At this point, you are able to initiate a job on Hadoop Spark.Later, you can verify operations by running the same job on the Trifacta Server.
    e. See *Run Job Page*.

7. When the job completes, you should see a success message in the job card for both the Transform job and the Profiling job.
    a. **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 the Trifacta Server. You can also download the log files to try to identify the problem.

8. Click **View Results** in the job card.In the Job Results page, you can see a visual profile of the generated results.
    a. See *Job Results Page*.

9. Click **Export Results**. In the Export Results window, click the CSV and JSON links to download the results to your local desktop.

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

Documentation

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