Table of Contents

1 Relativity Data Grid ........................................................................................................................................... 5
  1.1 Getting started ................................................................................................................................................... 5
  1.2 Hardware and system requirements .................................................................................................................. 7
  1.3 Supported and unsupported functionality ....................................................................................................... 7
    1.3.1 Storage (as compared to SQL storage) ........................................................................................................ 7

2 Elasticsearch system requirements ....................................................................................................................... 8
  2.1 Infrastructure considerations .......................................................................................................................... 8
  2.2 Other considerations ....................................................................................................................................... 8
  2.3 Elasticsearch cluster system requirements ..................................................................................................... 9
    2.3.1 Tier 2 (1.5-3TB) ......................................................................................................................................... 9
    2.3.2 Tier 3 (3-15 TB) ......................................................................................................................................... 10
  2.4 Java version compatibility ................................................................................................................................ 10

3 Installing Data Grid ............................................................................................................................................. 11
  3.1 Enabling your workspace and extracted text field for Data Grid ................................................................. 11
  3.2 Data Grid agents .............................................................................................................................................. 12

4 Installing Elasticsearch ....................................................................................................................................... 14
  4.1 Creating master, client, and data nodes ............................................................................................................ 14
    4.1.1 Adding additional masters .......................................................................................................................... 20
    4.1.2 Java Virtual Machine (JVM) settings ......................................................................................................... 21
  4.2 Creating a cluster .............................................................................................................................................. 21
  4.3 Installing a monitoring cluster ........................................................................................................................ 22
  4.4 Installing the license ....................................................................................................................................... 24
    4.4.1 Setting up the Kibana server ....................................................................................................................... 25
    4.4.2 Installing Kibana ....................................................................................................................................... 26
    4.4.3 Installing Marvel and Sense ...................................................................................................................... 27
    4.4.4 Configuring Shield .................................................................................................................................. 27
    4.4.5 Updating your Elasticsearch cluster ........................................................................................................ 27
  4.5 Linking Relativity to the Elasticsearch cluster .............................................................................................. 28
  4.6 Configuring Shield authentication ................................................................................................................ 32
7 Upgrading Data Grid ............................................................... 70
  7.1 Finding the Elasticsearch version .................................... 70
  7.2 Upgrading from Elasticsearch 2.1.2 to 2.3.3.x ...................... 71
  7.3 Preparing the environment for upgrade ............................ 71
  7.4 Running the upgrade script ............................................ 72
  7.5 Verifying the upgrade .................................................. 72
  7.6 Running a manual upgrade .......................................... 73
    7.6.1 Upgrading a node .................................................. 73
    7.6.2 Verifying the upgrade ............................................ 76
  7.7 Resetting the Shield or Head password ......................... 77

8 Audit ................................................................................. 79
  8.1 Supported functionality .............................................. 79
  8.2 Installing and configuring Audit .................................... 79
    8.2.1 Audit agents ....................................................... 80
  8.3 Audit tab ..................................................................... 80
    8.3.1 Searching on the Details column ............................. 81
    8.3.2 Exporting workspace audits .................................. 83
  8.4 Instance audits ............................................................. 84
    8.4.1 Filtering on multi-workspace audits ...................... 85
  8.5 Reverting audited actions ........................................... 86
    8.5.1 Mass reverting .................................................... 87
  8.6 Audit Migration Reports tab ....................................... 88
    8.6.1 Migration Error Report ....................................... 88
    8.6.2 Migration Error Retry Script ................................. 90
    8.6.3 Migration Status Report ...................................... 91
  8.7 Audit Workspace Settings tab ................................... 92
    8.7.1 DisplayAuditDataSource ...................................... 92
    8.7.2 HistoryTabVisibleOnUpgrade ................................. 93
  8.8 Audited actions .............................................................. 94
1 Relativity Data Grid

Relativity Data Grid allows you to store, search, and analyze extracted text and audit data at massive scale. Data Grid enables faster workflows through continuous indexing and distributed search to gain deeper insight to your extracted text and audit data. The benefits of using the Data Grid data store include:

- Scalability and more efficient review workflows on case sizes with 10 million or more documents and billions of audits.
- More performant database maintenance, backup, and upgrades.
- A reduction in SQL server database sizes, leading to better performing SQL databases.
- Increased visibility into reviewer productivity and system performance.

**Note:** Text or audit information stored in Relativity Data Grid is inaccessible for some third-party applications. It's recommended that you contact any vendors of third-party applications to confirm their compatibility with Relativity Data Grid.

Data Grid uses Elasticsearch to store and search audits. This page provides a brief description of Data Grid terminology, functionality, and important considerations to take into account before implementing Data Grid for new workspaces.

**Note:** Data Grid supports Windows servers only.

This page contains the following sections:

1.1 Getting started

Use the following workflow to get started with Data Grid.
What do you want to store?

Text Data

Storage

Configure your file share

Enable workspace for Data Grid

Enable extracted text fields for Data Grid

Import your extracted text

Audit Data

Audit

Setup and configure Elasticsearch

Install the Audit application

Start searching audits

Search

Build a dtSearch index

Search
1.2 Hardware and system requirements

The following table describes the hardware and system requirements per Data Grid feature.

<table>
<thead>
<tr>
<th>Storage</th>
<th>Audit</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hardware requirements</td>
<td>Fileshare</td>
</tr>
<tr>
<td>System requirements</td>
<td>Per Relativity specification</td>
</tr>
</tbody>
</table>

1.3 Supported and unsupported functionality

The following sections describe supported and unsupported functionality for Data Grid.

1.3.1 Storage (as compared to SQL storage)

To store text in Data Grid, you can use the same default file share as your natives and images, or you can designate a file share specifically Data Grid text. Once you enable an extracted text field’s access to Data Grid, you can’t disable it, so it's important to understand the benefits and limitations of storing text in Data Grid.

<table>
<thead>
<tr>
<th>Supported extracted text functionality</th>
<th>Currently unsupported functionality</th>
</tr>
</thead>
<tbody>
<tr>
<td>- Import/export through the Relativity Desktop Client</td>
<td></td>
</tr>
<tr>
<td>- Viewer</td>
<td></td>
</tr>
<tr>
<td>- Preview</td>
<td></td>
</tr>
<tr>
<td>- OCR</td>
<td></td>
</tr>
<tr>
<td>- dtSearch indexing and searching</td>
<td></td>
</tr>
<tr>
<td>- Persistent highlight sets</td>
<td></td>
</tr>
<tr>
<td>- Processing</td>
<td></td>
</tr>
<tr>
<td>- Relativity Integration Points</td>
<td></td>
</tr>
<tr>
<td>- Analytics</td>
<td></td>
</tr>
<tr>
<td>- ARM</td>
<td>- Keyword search</td>
</tr>
<tr>
<td></td>
<td>- SQL queries to long text fields stored in Data Grid</td>
</tr>
<tr>
<td></td>
<td>- Adding extracted long fields stored in Data Grid to layouts (including the Document panel)</td>
</tr>
<tr>
<td></td>
<td>- RSAPI query</td>
</tr>
<tr>
<td></td>
<td>- Pivot and Sort in the UI</td>
</tr>
<tr>
<td></td>
<td>- Filtering in the Document list on extracted text</td>
</tr>
<tr>
<td></td>
<td>- Mass operations:</td>
</tr>
<tr>
<td></td>
<td>- Edit</td>
</tr>
<tr>
<td></td>
<td>- Replace</td>
</tr>
<tr>
<td></td>
<td>- Tally/Sum/Average</td>
</tr>
<tr>
<td></td>
<td>- Export to File</td>
</tr>
</tbody>
</table>
2 Elasticsearch system requirements

Depending on your infrastructure tier, you have different server specifications and recommendations for the Elasticsearch cluster available to you. Elasticsearch is built on a distributed architecture made up of many servers or nodes. A node is a running instance of Elasticsearch (a single instance of Elasticsearch running in the JVM). Every node in an Elasticsearch cluster can serve one of three roles.

- Master nodes are responsible for managing the cluster.
- Data nodes are responsible for indexing and searching of the stored data.
- Client nodes are load balancers that redirect operations to the node that holds the relevant data, while offloading other tasks.

Set up an entirely separate cluster to monitor Elasticsearch with one node that serves all three roles: master, data, and client. While this setup doesn’t take advantage of the distributed architecture, it acts as an isolated logging system that won’t affect the main cluster.

2.1 Infrastructure considerations

Consider the following factors when determining the infrastructure requirements for creating an Elasticsearch environment:

- **Infrastructure tier** – When you build out your initial Relativity environment, we use these measures to determine a tier level of 1, 2, or 3. This tier level takes into consideration the number of users, SQL sizes, and the amount of data and activity in your system.

- **Virtual versus physical servers** – Although Elastic recommends physical servers, our implementation doesn’t require physical servers. Virtual servers can be implemented for all nodes.

- **Storage type** – Elasticsearch is a distributed system and you should run it on storage local to each server. SSDs are not required.

- **Network connectivity** – Because of the distributed architecture, network connectivity can impact performance, especially during peak activity. Consider 10 GB as you move up to the higher tiers.

- **Client nodes** – Larger clusters that do not perform heavy aggregations (search against your data), may perform better without client nodes. Simply use a master and data node configuration with a load balancer to handle data in your cluster.

**Note:** Elasticsearch won’t allocate new shards to nodes once they have more than 85% disk used.

2.2 Other considerations

- Shield is one of the many plugins that comes with Elasticsearch. Shield provides a user name and password for REST interaction and JWKS authentication to Relativity. JWKS is already running on your Relativity web server.

- The Elasticsearch cluster uses the certificate from a Relativity web server or a load balanced site for authentication to Relativity.
You can set up the nodes for TLS communication node to node. TLS communication requires a wild card for the nodes that contains a valid chain and SAN names. This is highly recommended for clusters that are in anyway exposed to the internet. You can request a script which can be used against an installation of OpenSSL to create the full chain that is not readily available. All of the certificates are contained within a Java keystore which is setup during installation by the script. To request this script, contact support@relativity.com.

If you have a chain of certificates with a wild card certificate and private key that contains SAN names of the servers, you can use those certificates to build the Java keystore for TLS.

### 2.3 Elasticsearch cluster system requirements

The number of nodes required and the specifications for the nodes change depending on both your infrastructure tier and the amount of data that you plan to store in Elasticsearch.

**Note:** These recommendations are for audit only.

#### 2.3.0.1 Test (425 GB)

<table>
<thead>
<tr>
<th>Node type</th>
<th># of nodes needed</th>
<th>CPU</th>
<th>RAM</th>
<th>DISK (GB)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Master/Client/Data</td>
<td>1</td>
<td>8</td>
<td>32</td>
<td>500</td>
</tr>
<tr>
<td>Monitoring cluster</td>
<td>0-1</td>
<td>4</td>
<td>8</td>
<td>200</td>
</tr>
</tbody>
</table>

#### 2.3.0.2 Tier 1 (850 GB)

<table>
<thead>
<tr>
<th>Node type</th>
<th># of nodes needed</th>
<th>CPU</th>
<th>RAM</th>
<th>DISK (GB)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Master/Data</td>
<td>1</td>
<td>8</td>
<td>32</td>
<td>1000</td>
</tr>
<tr>
<td>Client/Data</td>
<td>1</td>
<td>8</td>
<td>32</td>
<td>1000</td>
</tr>
<tr>
<td>Monitoring cluster</td>
<td>0-1</td>
<td>4</td>
<td>8</td>
<td>200</td>
</tr>
</tbody>
</table>

#### 2.3.1 Tier 2 (1.5-3TB)

<table>
<thead>
<tr>
<th>Node type</th>
<th># of nodes needed</th>
<th>CPU</th>
<th>RAM</th>
<th>DISK (GB)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Data</td>
<td>2-4</td>
<td>8</td>
<td>32</td>
<td>1800</td>
</tr>
<tr>
<td>Master*</td>
<td>3</td>
<td>4</td>
<td>8</td>
<td>200</td>
</tr>
<tr>
<td>Monitoring cluster</td>
<td>0-1</td>
<td>4</td>
<td>8</td>
<td>200</td>
</tr>
</tbody>
</table>

*Inactive master nodes are used as clients.
2.3.2 Tier 3 (3-15 TB)

<table>
<thead>
<tr>
<th>Node type</th>
<th># of nodes needed</th>
<th>CPU</th>
<th>RAM</th>
<th>DISK (GB)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Data</td>
<td>4-20</td>
<td>8</td>
<td>64</td>
<td>1800</td>
</tr>
<tr>
<td>Master*</td>
<td>3</td>
<td>4</td>
<td>8</td>
<td>200</td>
</tr>
<tr>
<td>Monitoring cluster</td>
<td>1</td>
<td>4</td>
<td>8</td>
<td>200</td>
</tr>
</tbody>
</table>

*Inactive master nodes are used as clients.

To assess the sizes of a workspace’s activity data and extracted text, contact support@relativity.com and request the AuditRecord and ExtractedText Size Gatherer script.

If you have further questions after running the script, our team can review the amount of activity and monitoring data you want to store in Elasticsearch and provide a personalized recommendation of monitoring nodes required.

2.4 Java version compatibility

See the Elastic website for compatible Java versions.
3 Installing Data Grid

This page includes steps for configuring Data Grid Text (Core) in your environment. Before configuring Data Grid, ensure you've completed the pre-installation steps. For more information, see .

3.1 Enabling your workspace and extracted text field for Data Grid

To enable your workspace for Data Grid, perform the following steps:

**Note:** We recommend you only enable Data Grid for fields storing extracted text, OCR text, or translated text.

1. Navigate to the **Workspace Details** tab, and then click **Edit**.
2. Set the **Enable Data Grid** field to **Yes**.
3. (Optional) Next to **Data Grid File Repository**, select the path for the physical location of the text files used by Data Grid. If no file repository is specified for this field, and Data Grid is enabled, Data Grid stores text in the default file repository.

   **Note:** If you run out of space in this repository, you can specify a new repository. Data Grid will continue to read from the old repository as well as the new repository.

4. Click **Save**.
To enable the extracted text field for Data Grid, perform the following steps:

1. Navigate to the **Fields** tab.
2. Locate the extracted text field and click the **Edit** link next to it.
3. Set the **Enable Data Grid** field to **Yes**.

![Field Information](image)

**Note:** If you are storing extracted text in Data Grid, the **Include in Text Index** field is set to No because there is no SQL text index. If you want to search using dtSearch, you must follow best practice of creating a saved search of fields you want to index.

4. Click **Save**.

**Note:** Enabling extracted text fields for Data Grid works for new workspaces only. You can't enable Data Grid for fields that already have text in SQL. If you want to migrate fields from SQL to Data Grid, you must use the Data Grid Text Migration application.

### 3.2 Data Grid agents

A number of agents are available to facilitate Data Grid operations in your environment. Ensure that the following agents are installed in your environment. For more information on installing agents, see the Agents Guide.
<table>
<thead>
<tr>
<th>Agent name</th>
<th>Requirement information</th>
<th>Function</th>
<th>Agent type</th>
</tr>
</thead>
<tbody>
<tr>
<td>Data Grid Manager</td>
<td>Only 1 per environment</td>
<td>A Data Grid Manager agent is an off-hours agent responsible for Data Grid enabled workspace management, including deleting outdated search results cache tables and monitoring Data Grid index conditions.</td>
<td>Single-installation</td>
</tr>
<tr>
<td>Data Grid Worker</td>
<td>At least 1 per environment</td>
<td>A Data Grid worker agent is part of building the Data Grid File Repository.</td>
<td>Multiple-installation</td>
</tr>
</tbody>
</table>
4 Installing Elasticsearch

In order to use Audit in your environment, you must install Elasticsearch.

The procedure for installing Elasticsearch to your Relativity environment is the following:

1. Complete the pre-installation steps. See the Pre-Installation Guide.

2. Install the software on which Elasticsearch runs. This includes:
   a. Master, client and data nodes. See Creating master, client, and data nodes below
   c. A monitoring cluster. See Installing a monitoring cluster on page 22

3. Install and configure the Relativity components. These include the following:
   a. The license and plugins. See Installing the license on page 24
   b. The Instance Settings that accompany the Data Grid Core application. See Linking Relativity to the Elasticsearch cluster on page 28.
   c. Elasticsearch plugins. See Setting up the Kibana server on page 25.
   d. Shield authentication. See Configuring Shield authentication on page 32

Once all installation steps are completed, you must configure the Relativity core application. See Installing Data Grid on page 11.

**Note:** Ensure that your antivirus software excludes the Data Grid data location (data node) from scanning. See the Environment Optimization Guide.

4.1 Creating master, client, and data nodes

Three different types of nodes are required to run Elasticsearch in your environment:

- Master - the node within a cluster that manages changes across an entire cluster.
- Client - the node within a cluster that either serves as the gateways through which data enters a cluster.
- Data - the node that stores data within a cluster.

The Tier 1 installation of Elasticsearch involves configuring one or more dedicated master nodes, one dedicated client node, one monitoring node, and two data nodes. The master node(s) manage the overall state of the cluster, delegates the structure by which data is stored and replicated across a cluster, and facilitates resources returned during searches. The client node serves as the endpoint through which you import data into the Elasticsearch cluster. The data nodes store primary shards and replica shards.

**Note:** Elasticsearch won’t allocate new shards to nodes once they have more than 85% disk used.

The account running the Elasticsearch service requires access to SQL Server, and specifically needs to have read, write, and bulk permissions for all workspace databases.
For more information on cluster specifications for each tier, see Elasticsearch system requirements on page 8.

To install a master, client, or data node, perform the following steps. Note that the settings within the Elasticsearch.yml determine which type of node you’re installing, but the remaining steps apply to all node types.

1. Install the Java Development Kit. See the Elastic website for compatible Java versions. For questions, email support@relativity.com.

2. Use the following steps to insert an environment variable (KCURA_JAVA_HOME).
   a. Click Start.
   b. Right-click on Computer and select Properties.
   c. Click Advanced system settings.
   d. Select the Advanced tab.
   e. Click Environment Variables…
   f. Click New under System Variables.
   g. Name the variable KCURA_JAVA_HOME.
   h. Copy the file path to C:\Program Files\Java\jdk(version_number)

3. Contact your Relativity Account Manager to download the Elasticsearch installer package.
4. Extract the Elasticsearch zip folder to a root directory (Example: C:\RelativityDataGrid).

5. Rename the default directory (\RelativityDataGrid\elasticsearch-2.x.x) to RelativityDataGrid\elasticsearch-main. This allows you to make upgrades to Elasticsearch without having to modify the folder to accommodate future version numbers.

6. Navigate to RelativityDataGrid\elasticsearch-main\config\elasticsearch.yml and update the settings in the following Elasticsearch.yml file in a text editor. This is where you'll specify what type of node you're creating:

   **Note:** When entering these values, you must enter a single space between the field name and the field value.

   - **cluster.name** - enter the name of the cluster. All nodes on the cluster must share a common name in order to communicate with one another.
     
     **Note:** You can't use hyphens to name a cluster. Use underscores if necessary.

   - **node.name** - enter the fully qualified domain name.

   - **node.master** - enter true because this is a master node. Enter false for client and data nodes.

   - **node.data** - enter false for a master and client node. Enter true for a data node.

   - **discovery.zen.minimum_master_nodes** - enter 1

   - **discovery.zen.ping.multicast.enabled** - enter false

   - **discovery.zen.ping.unicast.hosts** - enter the full computer name of all nodes on the cluster using the following format: ["DN_FULL_COMPNAME_1","DN_FULL_COMPNAME_2","DN_FULL_COMPNAME_3"]. Always include the master node in the list of hosts.
     
     **Note:** Don't put spaces between the commas when specifying multiple nodes. The default port number is 9300, but you can specify a port number after the computer name with the following format: "DN_FULL_COMPNAME_1:9500". The specified port must be free.

   - **marvel.agent.enabled** - enter false

   - **marvel.agent.exporter.es.hosts** - enter the computer name (and optionally port number) that each node should send stats to. ["host1","host2:port"]

   - **action.destructive_requires_name** - enter true. This prevents destructive actions w/ wildcards. For example, DELETE /*.

   - **action.auto_create_index** - enter false. This disables automatic index creation.

   - **format** - enter json. This is for global cluster state file to be readable.

   - **transport.tcp.compress** - enter true

   - **http.max_content_length** - enter 201mb

   - **http.cors.enabled** - enter false
- **gateway.expected_master_nodes** - enter 0. This delays cluster recovery, providing additional time for all nodes to first come online.
- **gateway.expected_data_nodes** - enter 0
- **gateway.recover_after_time** - enter 5m
- **script.disable_dynamic** - enter true. This puts all scripting functionality in a sandbox.
- **script.default_lang** - enter groovy
- **script.groovy.sandbox.enabled** - enter true
- **path.data** - enter the path of the location in which you want to store the allotted index data for this node. For example, `C:\RelativityDataGrid\data`.
- **path.repo** - enter the path of the location in which you want to store backups. For example, `\networklocation\backups`.

**Note:** You can specify multiple backup locations with the following format: `"/mount/backups", "/mount/longterm_backups"`. You can also specify a network share with the following format: `"\\my_server\snapshots"`.

- Configure the Shield settings as follows:

  **Note:** To disable Shield, remove the number sign (#) in front of `shield.enabled:false`.

  ```
  #shield.enabled: false
  shield.authc.realms:
    custom:
      type: kCuraBearerRealm
      order: 0
      publicJWKsUrl: https://<server>/Relativity/Identity/.well-known/jwks
  esusers1:
    type: esusers
    order: 1
  ```

  **Note:** The URL must point to the Relativity installation where Identity Server can be found. This should be the same URL used to log in to Relativity.

7. If using Shield, see Configuring Shield authentication on page 32 for steps on configuring users and roles.

8. Use the following steps to launch the command prompt:
   a. Click **Start**.
   b. Search for "**cmd**".
   c. Right-click on **cmd.exe** and select **Run as Administrator**.

9. Navigate to the bin directory in the RelativityDataGrid folder (`\RelativityDataGrid\elasticsearch-main\bin`) by running `cd \RelativityDataGrid\elasticsearch-main\bin` and enter the following commands:
a. Install the Windows service: `\kservice.bat install`

b. Run the GUI manager: `\kservice.bat manager`

c. Use the drop down menu for **Startup Type** to select **Automatic**.

d. Click the **Java** tab and configure the **Maximum memory pool** to be 30 GB or half of the total RAM available (whichever is less). You can enter identical values for the **Initial memory pool** and **Maximum memory pool** settings. It's recommended that you keep the **Thread stacks size** value at its default of 256.
e. Select the Log On tab. In the Log on as setting, select This account. Enter a valid Relativity service account domain name and password and confirm the password.

![Elasticsearch Properties](image)

f. Click OK.

- g. Return to the command prompt.

- h. Start the Windows service: `\kservice.bat start`

  **Note:** Don’t start the Windows service on a node until you complete step 5 above to configure the elasticsearch.yml file on the node.

10. Verify the installation is complete by navigating to the following address in a Chrome browser:

    - http://computername:9200/_plugin/head
    - http://localhost:9200/_plugin/head

  **Note:** This page might not properly display in Internet Explorer. Use Chrome for verification.

### 4.1.1 Adding additional masters

Elasticsearch 2.x allows for multiple masters with a minimum of 3 masters. With multiple masters, your cluster is highly available. If a master node goes down, one of your additional master nodes can be elected and your cluster will continue in an active green state.

Set up your additional two master nodes per the instruction for installing Elasticsearch nodes.

The following needs to be updated in the elasticsearch.yml file for all master nodes:

**discovery.zen.minimum_master_nodes:** 2
This will ensure that if any master goes offline, the third one joins the cluster forming a quorum.

For more information please see Elastic documentation [here](#).

**Note:** Enabling multiple masters is optional. If you would like to stay at 1 master, move on to the next step to install plugins.

### 4.1.2 Java Virtual Machine (JVM) settings

You must also configure the following settings on each JVM for JVM logging:

1. Run the Elasticsearch GUI manager: `.\kservice.bat manager`
2. Select default logging to "Error" instead of "Info".
3. Update JVM memory to half of total RAM of the machine or 30GB (whichever is less).
4. Start each of the nodes services: `.\kservice.bat start`

**Note:** Never restart a node if data already exists on the cluster. If there is no data in the cluster, you can restart the service if it's already started and configuration settings change. See [Restarting nodes and clusters](#) for more information.

Clusters automatically form between nodes on a network if the cluster names are the same across nodes.

### 4.2 Creating a cluster

Clusters automatically form between multiple nodes when the same cluster name is specified across multiple nodes in the `elasticsearch.yml` file. You can check to see if your various nodes have joined the cluster by checking Head. Navigate to [http://computername:9200/_plugin/head](http://computername:9200/_plugin/head) or [http://localhost:9200/_plugin/head](http://localhost:9200/_plugin/head) and see if all of your nodes are present.
4.3 Installing a monitoring cluster

A monitoring cluster allows you to store Marvel data from the production cluster for analysis. A monitoring cluster for Data Grid only needs one node, but you can set up a multi-node monitoring cluster if you prefer. We recommend using no more than three monitoring cluster nodes.

Use the following steps to install Elasticsearch 2.x on a machine that you want to use as a single node monitoring cluster:

1. Install the Java Development Kit. You can acquire the correct version by emailing support@relativity.com.

2. Use the following steps to insert an environment variable (KCURA_JAVA_HOME):
   1. Click Start.
   2. Right-click on Computer and select Properties.
   3. Click Advanced system settings.
   4. Select the Advanced tab.
   5. Click Environment Variables…
6. Click **New** under **System Variables**.
7. Name the variable ** KCURA_JAVA_HOME**.
8. Copy the file path to **C:\Program Files\Java\jdk(version_number)**

3. Contact your Relativity Account Manager to download and unzip the Elasticsearch installer package.

4. Extract the Elasticsearch zip folder to a root directory (Example: C:\RelativityDataGrid).

5. Rename the default directory (C:\RelativityDataGrid\elasticsearch-2.1.x) to **RelativityDataGrid\elasticsearch-main**. This allows you to make upgrades to Elasticsearch without having to modify the folder to accommodate future version numbers.

6. Extract and copy the Relativity Data Grid package to each node on your monitoring cluster.

7. Navigate to **RelativityDataGrid\elasticsearch-main\config\elasticsearch.yml** and update only the following settings in the elasticsearch.yml file in a text editor.

   **Note:** When entering these values, you must enter a single space between the field name and the field value.

   - **cluster.name** - enter the name of the monitoring cluster. All nodes on the cluster must share a common name in order to communicate with one another. Do not confuse this cluster name
with the name of the production cluster. This cluster name should be the name of the monitoring cluster.

- **node.name** - enter the name of the monitoring node.
- **node.master** - enter true because this node performs the same tasks as a master node.
- **node.data** - enter true. You can use the master node to store data for a single node monitoring cluster, but don’t store data on the master node for Tier 2 or Tier 3 setups on a production cluster.
- **action.auto_create_index** - enter true.

8. If you have Shield enabled on your cluster, create a user with a kibana4_server role and marvel_remote_agent role, both of which are defined in roles.yml. For more information on defining roles, see Configuring Shield authentication on page 32.

### 4.4 Installing the license

Install the Elasticsearch license from the search/bin folder. If you haven’t already, submit a support ticket to Relativity Client Services for a production grade license for Elasticsearch plugins. While you wait for a license, you can continue using the plugins on a trial period.

To install the license, complete the following in Sense or Head:
1. Copy the contents of the license and run `PUT/_license<<contents_of_license_file.json>>`

2. Verify that the expiration date is set to a year from now by running the command `GET/_license`.

![Elasticsearch interface with license field](image)

### 4.4.1 Setting up the Kibana server

Extract the Kibana 4.5.3 package on your monitoring cluster, and then use the following steps to set up the Kibana server on your monitoring cluster.

The following table breaks down the compatibility of different versions of Data Grid, Marvel, and Kibana.

<table>
<thead>
<tr>
<th>Data Grid version</th>
<th>Marvel version</th>
<th>Kibana version</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.3.3</td>
<td>2.3.3</td>
<td>4.5.3+</td>
</tr>
</tbody>
</table>
4.4.2 Installing Kibana

Kibana is a requirement for Marvel which allows you to manage your Elasticsearch indexes. To use Kibana with Elasticsearch, you need the following:

- Elasticsearch 2.1 or later.
- One of the following supported browsers: IE9, IE10+, Firefox, Chrome, Safari (Mac).
- The endpoint of the master node of the Data Grid instance you're connecting to.
- Which Data Grid indexes you want to search.

To install and start Kibana:

1. Download the [Kibana 4.5.x binary package](#) for your platform.
2. Extract the .zip or tar.gz archive file.
3. After installing, run Kibana from the install directory: bin/kibana (Linux/MacOSX) or bin\kibana.bat (Windows).
4. Navigate to the kibana.yml file, and update the following in a text editor:
   - `elasticsearch.url` - enter the name of your monitoring cluster.
5. If you have Shield enabled on your cluster, also update the following `kibana.yml` settings in a text editor:
   - `kibana_elasticsearch_username` - enter the name of the user with a kibana4_server role defined in roles.yml.
   - `kibana_elasticsearch_password` - enter the password for the user who has the kibana4_server role defined in roles.yml.
   - `elasticsearch.ssl.ca` - enter PEM file path.
   - `elasticsearch.ssl.verify` - enter false.
   - `path.data` - enter the path of the location in which you want to store the allotted data for this node. For example, C:/RelativityDataGrid/data.
   - `path.repo` - enter the path of the location in which you want to store backups. For example, \networklocation\backups. You can specify multiple backup locations with the following format: ["/mount/backups", "/mount/longterm_backups"]. You can also specify a network share with the following format: ["\\my_server\snapshots"].
   - `server.port` - enter the port where the Kibana server runs. Defaults to 5601.
   - `server.host` - enter the IP address where the Kibana server runs. Defaults to 0.0.0.0 (localhost).
6. Browse to the kibana/bin folder and enter the following:
   - kibana plugin -i elasticsearch/marvel/2.3.3
   - kibana plugin -i elastic/sense
7. Start the Kibana server by running the following:
   - bin/kibana.bat


### 4.4.3 Installing Marvel and Sense

Marvel is a management and monitoring tool for Elasticsearch. Marvel aggregates cluster wide statistics and events and offers a single interface to view and analyze them.

Sense is a Kibana plugin and visual console that provides auto-complete, auto-indentation, and syntax checking.

To install Marvel and Sense:

1. Refer to the [Marvel requirements](#) and [Sense requirements](#).
2. Open the command prompt and go to the `elasticsearch/bin` directory. Enter the following command: `plugin install marvel-agent`
3. Browse to the Kibana/bin folder and enter the following:
   - kibana plugin --install elasticsearch/marvel/2.3.3
   - kibana plugin --install elastic/sense

### 4.4.4 Configuring Shield

When you install Elasticsearch 2.3.3, Shield is enabled by default; however, you still need to configure Shield's authentication on every node. For more information, see [Configuring Shield authentication](#).

**Note:** With Shield on by default, other plugins like Marvel or Head are not supported. In order to use your other plugins, you need to provide the Kibana server with credentials so it can access the .kibana index and monitor the cluster. See [Configuring Shield authentication](#) for more information.

### 4.4.5 Updating your Elasticsearch cluster

To finish setting up your monitoring cluster, the following changes need to be made on all nodes in the Elasticsearch cluster.

Complete the following on all nodes in the Elasticsearch cluster:

1. Install the marvel-agent plugin by running the command below.
   - bin\plugin install marvel-agent
2. If your monitoring cluster has Shield enabled, navigate to `C:\relativity\elasticsearch-main\config\elasticsearch.yml` and update the elasticsearch.yml file in a text editor with the following. Set the `auth.username` and `auth.password` to the user who has the remote_marvel_agent role defined in roles.yml

   ```yaml
   marvel.agent.exporters:
     id1:
       type: http
   ```
3. Restart the Elasticsearch service for the changes to take effect.


### 4.5 Linking Relativity to the Elasticsearch cluster

After you’ve installed the Data Grid Core application and added the accompanying agents in the Agents tab, you'll need to configure all the entries in the Instance Settings tab that allow Relativity to communicate with the Elasticsearch service.

This will enable Relativity to save data into the cluster you created as part of the Elasticsearch component installation.

Use the following steps to link Relativity to your Elasticsearch service:

1. Specify the URL and port of the client node in the DataGridEndPoint instance setting value (i.e., http://CLIENT_NODE_COMPNAME:9200). This is the endpoint that Relativity uses to ingest data into the cluster. If you have multiple client nodes, you can specify all of them as endpoints. Separate your endpoints by commas.

2. Enter a unique prefix for this instance of Relativity in the DataGridIndexPrefix instance setting value. The prefix allows you to differentiate data from instance to instance within the cluster.

3. Set the **number_of_shards** and **number_of_replicas** fields in the DataGridIndexCreationSettings and DataGridIndexCreationSettings instance setting value to match your cluster setup. The **number_of_shards** field should be one shard per data node, and the **number_of_replicas** field should be two. The default instance setting value for number_of_shards is set to four, which assumes four data nodes. The default instance setting value for number_of_replicas value is two.

   Use the following steps to edit the DataGridIndexCreationSettings instance setting value to match your particular cluster setup. The instance setting value holds the template for creating a text index (DataGridIndexCreationSettings) in Elasticsearch.

   a. Enter the following for the DataGridIndexCreationSettings entry in Instance Settings, edit the **number_of_shards** and **number_of_replicas** fields to match your preferred settings, and paste the updated text into the text box.

```
{
  "template": "relativity_*",
  "aliases": {
    "{index}_read": {},
    "{index}_write": {},
    "{index}_verify": {}
  },
  "settings": {
    "index": {
      "number_of_shards": 12,
      "number_of_replicas": 2
    },
    "analysis": {
```
b. You must install the Data Grid for Audit application from the library in at least one workspace in order to use Data Grid for Audit and to create audit indexes. This installation creates your necessary Data Grid for Audit entries in the Instance Settings tab.

c. Enter a unique prefix for this instance of Relativity in the ESIndexPrefix instance setting value. The prefix allows you to differentiate data from instance to instance within the cluster.

d. Set the number of shards and number of replicas fields in the ESIndexCreationSettings instance setting value to match your cluster setup. The number of shards field should be twice the number of nodes on the cluster, and the number of replicas field should be two. Use the following steps to edit the ESIndexCreationSettings and instance setting value to match your particular cluster setup. The instance setting value holds the template for creating an audit index (ESIndexCreationSettings) in Elasticsearch.

e. The ESIndexCreationSettings value populates with the default template that includes settings for four shards and two replicas. Edit the ESIndexCreationSettings entry in Instance Settings to update the template if your setup requires shard or replication modifications. Always update the template through these instance settings to keep templates consistent across

```json
"mappings": {
  "document": {
    "_timestamp": {
      "enabled": true
    },
    "_size": {
      "enabled": true
    },
    "dynamic_templates": [
      {
        "raw": {
          "match_mapping_type": "string",
          "path_match": "*.Value",
          "mapping": {
            "type": "string",
            "fields": {
              "raw": {
                "type": "string",
                "analyzer": "lwhitespace"
              }
            }
          }
        }
      }
    ]
  }
}
```
Relativity and Elasticsearch.

**Note:** If you already migrated audit data into Elasticsearch through the Data Grid for Audit application, you must manually delete the template in Elasticsearch (DELETE/_template/audit and DELETE/_template/document) before updating the template with any new changes. Contact support@relativity.com if you require assistance.

f. Use the following command to edit the ESIndexCreationSettings instance setting value to match your particular cluster setup:

```json
{
    "template": "audit_*",
    "aliases": {
        "{index}_read": {
        },
        "{index}_write": {
        },
        "{index}_verify": {
        }
    },
    "settings": {
        "index": {
            "number_of_shards": 4,
            "number_of_replicas": 2
        },
        "analysis": {
            "analyzer": {
                "str_search_analyzer": {
                    "tokenizer": "keyword",
                    "filter": ["lowercase",
                                "substring"]
                },
                "str_index_analyzer": {
                    "tokenizer": "keyword",
                    "filter": ["lowercase",
                                "substring"]
                },
                "lwhitespace": {
                    "tokenizer": "whitespace",
                    "filter": ["lowercase"]
                }
            },
            "filter": {
                "substring": {
                    "type": "nGram",
                    "min_gram": 1,
                    "max_gram": 20
                }
            }
        },
        "mappings": {
            "audit": {
                "dynamic_templates": [{
                    "raw": {
                        "match_pattern": "regex",
                        "type": "nGram",
                        "min_gram": 1,
                        "max_gram": 20
                    }
                },
```

"path_match": "Details\.\auditElement\..\.*",
"mapping": {
  "type": "string",
  "fields": {
    "Raw": {
      "analyzer": "lwhitespace",
      "type": "string"
    }
  }
},
"match_mapping_type": "string"
},
{ "newvalue": {
  "match_pattern": "regex",
  "path_match": ".\.*\\.newValue$",
  "mapping": {
    "type": "string",
    "fields": {
      "Raw": {
        "analyzer": "lwhitespace",
        "type": "string"
      }
    }
  }
}
},
{ "oldvalue": {
  "match_pattern": "regex",
  "path_match": ".\.*\\.oldValue$",
  "mapping": {
    "type": "string",
    "fields": {
      "Raw": {
        "analyzer": "lwhitespace",
        "type": "string"
      }
    }
  }
}
},
{ "analytics_text": {
  "match_pattern": "regex",
  "path_match": ".\.*\#text$",
  "mapping": {
    "type": "string",
    "fields": {
      "Raw": {
        "analyzer": "lwhitespace",
        "type": "string"
      }
    }
  }
}
}
,"_timestamp": {
4. Change the **Enable Data Grid** setting on the workspace details tab from **No** to **Yes**.

5. Change the **Enable Data Grid** setting for extracted text from **No** to **Yes**. On import, all data mapped to this field saves to Data Grid.

6. For information on setting up Data Grid for Audit, see [Installing Data Grid on page 11](#).

**Note:** You can't migrate extracted text fields that already contain data from the SQL Server to Data Grid.

### 4.6 Configuring Shield authentication

Shield allows you to easily protect your data from unintentional modification or unauthorized access while simplifying your architecture. Without Shield, anyone who knows the correct endpoint or URL can query, update, or delete all data in Data Grid. With default Shield installation, only Relativity can access Data Grid data, and all traffic between servers is encrypted. In addition, system admins can specify custom authentication realms so that targeted users can interact with Data Grid data outside of Relativity (via plugins and endpoints).

When you install Elasticsearch 2.3.3, Shield is enabled by default; however, you still need to configure Shield's authentication on every node.

The basic procedure for configuring Shield authentication is the following:

1. Install the authenticate Shield plugin on every node. [Installing the Shield plugins on the next page](#).
2. Configure custom authentication. See [Configuring custom authentication on the next page](#).
3. Install an X.509 certificate. See [Installing an X.509 certificate on page 36](#).
4. Enable SSL in the node configuration. See [Enabling SSL in the node configuration on page 38](#).
5. Restart Elasticsearch.
Note: With Shield on by default, other plugins like Marvel or Head aren't operational until you configure Kibana to use your other plugins. For information on configuring Kibana, see Configuring Kibana to work with Shield.

4.6.1 Installing the Shield plugins

To install the authenticate and Shield plugins, perform the following steps on every node:

1. Copy the authenticate and Shield plugins from the elasticsearch-main\plugins folder in the zip file to the plugins folder.
2. Restart Elasticsearch.

Note: Relativity 9.3.389.9 includes a change to the Shield plugin. Data Grid no longer needs to send credentials back to Relativity for verification, which enhances security between Data Grid and Relativity. If you adopted Shield before Relativity 9.3.389.9, we recommend you update your elasticsearch.yml file with the new Shield configuration on every node to get the benefits of this change to Data Grid authentication.

4.6.2 Configuring custom authentication

With Shield authentication, you can also specify custom authentication that allows targeted users to interact with Data Grid data outside of Relativity. To specify custom authentication, you must first define roles in the roles.yml file, located in <installDirectory>\plugins\shield\config\roles.yml, and assign users to roles using your preferred authentication realm.

4.6.2.1 Defining roles

The roles.yml file contains a default role called admin, which Relativity uses to access the data stored in Relativity. The admin role has complete system admin rights to all data stored in Data Grid.

Note: The node manages the roles.yml file locally; the cluster doesn't manage the yml file globally. This means that with a typical multi-node cluster, you need to apply the exact same changes on each and every node in the cluster. A safer approach would be to apply the change on one of the nodes and have the roles.yml distributed/copied to all other nodes in the cluster (either manually or using a configuration management system such as Puppet or Chef).

If you want to create a new role with a different permission set, you must configure the roles.yml file to specify the unique role name(s) and the cluster and indices permissions associated with it. For example:

```
# All cluster rights
# All operations on all indices
admin:
  cluster: all
  indices:
    '*':
      privileges: all

# Monitoring cluster privileges
# All operations on all indices
power_user:
  cluster: monitor
  indices:
```
Only read operations on indices

```yaml
user:
  indices:
    '*':
      privileges: read
  events_*:
    privileges: read
```

See the complete list of cluster and indices privileges here.

Once you create your roles, you can create users using any of the following authentication realms and associate them to a role to determine the level of permissions they have to access data stored in Data Grid.

### 4.6.2.2 Native realm

The default Shield realm is the native realm. In the native realm, all user operations occur within the Elasticsearch cluster. The native realm stores and distributes users with the Elasticsearch cluster. Users authenticate with a username and password.

To add or update a user to the native realm, submit a PUT or POST request to the `/_shield/user/<username>` endpoint. The username must be at least 1 character and no longer than 30 characters. The first character must be a letter (a-z or A-Z) or an underscore (_). Subsequent characters can be letters, underscores (_), digits (0-9), or any of the following symbols: @, -, . or $.

```
POST /_shield/user/username
{
  "password" : "P@ssword1",
  "roles" : [ "admin", "other_role1" ],
  "full_name" : "Jane Smith",
  "email" : "jsmith@example.com",
  "metadata" : {
    "intelligence" : 7
  }
}
```

Note the following considerations:

- **password** - the password for the user. Passwords must be at least 6 characters long.
- **roles** - determines the user's access privileges. You must assign at least one role to the user.
- **full_name** - (Optional) the user's full name.
- **email** - (Optional) - the user's email address.
- **metadata** - (Optional) arbitrary metadata you want to associate with the user.

### 4.6.2.3 Elastic Shield realm

Using the Elastic Shield realm allows you to add a user to your cluster and specify a password. This realm is not recommended because it potentially undermines Shield authentication.
To configure the Elastic Shield realm:

1. Open the command prompt and navigate to `elasticsearch/bin/shield`.

2. Enter the following command to add a new user to your cluster and specify a password for that user:
   
   ```bash
   esusers useradd <username> -p <secret>
   ```

3. To add users to roles, enter the following command:

   ```bash
   esusers roles <username> -a <comma-separated list of roles> -r <comma-separated list of roles>
   ```

   The `-a` option adds a comma-separated list of roles to a user. The `-r` option removes a comma-separated list of roles from a user.

### 4.6.2.4 LDAP realm

To integrate with LDAP you must configure an LDAP realm and assign LDAP groups to Shield roles in the role mapping file.

To configure the LDAP realm with User search, add the following settings to the `elasticsearch.yml` file in a text editor:

```yaml
shield.authc.realms:
  ldap1:
    type: ldap
    order: 1
    url: "LDAP://Test-DC01.testing.corp:389"
    bind_dn: "CN=Administrator,CN=Users,DC=testing,DC=corp"
    bind_password: P@ssw0rd@1
    user_search:
      base_dn: "DC=testing,DC=corp"
      attribute: sAMAccountName
    group_search:
      base_dn: "DC=testing,DC=corp"
```

To configure the LDAP realm with User templates, add the following settings to the `elasticsearch.yml` file in a text editor:

```yaml
ldap2:
  type: ldap
  order: 2
  url: "LDAP://Test-DC01.testing.corp:389"
  user_dn_templates:
    - "cn={0},OU=Testing - Users,DC=testing,DC=corp"
  group_search:
    base_dn: "DC=testing,DC=corp"
```

To assign LDAP groups to Shield roles, add the following settings for your user to the `role_mapping.yml` file in a text editor:

```yaml
admin:
  - "OU=Testing - Admins,DC=testing,DC=corp"
  - "OU=Testing - Users,DC=testing,DC=corp"
  - "CN=Jane Smith,OU=Testing - Users,DC=testing,DC=corp"
```
4.6.2.5 Active Directory (AD) realm

To configure the Active Directory realm, add the following settings to the elasticsearch.yml file in a text editor:

```yaml
active_directory:
type: active_directory
order: 2
domain_name: testing corp
url: ldap://testing.corp:636
user_dn_templates:
  - "CN=Administrator,OU=Testing Users,DC=testing,DC=corp"
group_search:
  base_dn: "DC=testing,DC=corp"
```

4.6.3 Installing an X.509 certificate

Installing an X.509 certificate consists of the following tasks:

1. Create a keystore and generate a node certificate.
2. Create a certificate signing request (CSR).
3. Send the certificate to your certificate authority (CA) for signing OR set up your certificate authority to sign the certificate.
4. Add the signed certificate to the node's keystore.

4.6.4 Create a keystore and generate a keypair

1. Create a keystore and import your certificate authority's (CA) certificate or a trusted certificate using Java Keytool. This process configures the node to trust certificates signed by the CA. For example, the following command creates a keystore for node01 and imports the CA certificate cacert.pem.
   Local self-signed certificates are not recommended. The keytool can be found in the following:

   ```bash
   cd \Program Files\Java\jdk1.8.*\bin keytool -importcert -keystore node01.jks -file cacert.pem -alias my_ca
   ```

   The Java keystore file (.jks) securely stores certificates for the node. The CA cert must be a PEM encoded certificate. If you need to convert your certificate to PEM file, you can use a tool such as OpenSSL to convert a certificate.

   When you create a keystore, you are prompted to set a password. This password protects the integrity of the keystore. You need to provide it whenever you interact with the keystore.

   **Note:** When the CA certificate expires, you must update the node's keystore with the new CA certificate.

   You can also store the CA certificate in a separate truststore. For more information, see Configuring a truststore.

2. Generate a private key and certificate for the node with Java Keytool. For example, the following command creates a key and certificate for node01:

   ```bash
   ```
keytool -genkey -alias node01 -keystore node01.jks -keyalg RSA -keysize 2048 -validity 712 -ext san=dns:node01.example.com,ip:192.168.1.1

This command creates an RSA private key with a key size of 2048 bits and a public certificate that is valid for 712 days. The key and certificate are stored in the node01.jks keystore.

The `san` value specifies all alternative names for the node. The generated certificate is valid for the DNS names and IP addresses specified as alternative names. You can specify multiple DNS or IP address entries as a comma-separated list.

When you run `keytool -genkey`, Keytool prompts you for the information needed to populate the node’s distinguished name that’s stored the certificate. Use a trusted domain wildcard certificate that can be trusted among many nodes. For example:

```
What is your first and last name?
[Unknown]: *.domain.corp

What is the name of your organizational unit?
[Unknown]: test

What is the name of your organization?
[Unknown]: Relativity
What is the name of your City or Locality?
[Unknown]: Chicago

What is the name of your State or Province?
[Unknown]: Illinois

What is the two-letter country code for this unit?
[Unknown]: US

Is CN=*.domain.corp, O=test, O=Relativity, L=Chicago, ST=Illinois, C=US correct?
[no]: yes

Enter key password for <node01>
(RETURN if same as keystore password):
```

4.6.5 Create a certificate signing request (CSR)

A node’s certificate needs to be signed by a trusted CA for the certificate to be trusted. To get a certificate signed, you need to create a certificate signing request (CSR) and send it to your CA.

To create a CSR with Java Keytool, use the `keytool t-certreq` command. You specify the same alias, keystore, key algorithm, and DNS names and IP addresses that you used when you created the node certificate. Specify where you want to store the CSR with the `-file` option.

```
keytool -certreq -alias node01 -keystore node01.jks -file node01.csr -keyalg rsa -ext san=dns:*.domain.corp,ip:192.168.1.1
```

4.6.6 Send the certificate to your certificate authority (CA) for signing

To get a signed certificate, send the generated CSR file to your CA. The CA will sign it and send you the signed version of the certificate.
4.6.7 Install the signed certificate
To install the signed certificate, use `keytool -importcert` to add it to the node’s keystore. You specify the same alias and keystore that you used when you created the node certificate.

```
  cd CONFIG_DIR/shield
  keytool -importcert -keystore node01.jks -file node01-signed.crt -alias node01
```

4.7 Enabling SSL in the node configuration
Once you’ve added the signed certificate to the node’s keystore, you need to modify the node configuration to enable SSL then restart Elasticsearch.

To enable SSL, make the following changes in `elasticsearch.yml`:

1. Specify the location of the node’s keystore and the password(s) needed to access the node’s certificate. For example:

```
  shield.ssl.keystore.path: /home/es/config/shield/node01.jks
  shield.ssl.keystore.password: myPass
  shield.ssl.hostname_verification: false
```

   The first line indicates the full path to the node keystore file. This must be a location within the Elasticsearch configuration directory.

2. Enable SSL on the transport networking layer to ensure that communication between nodes is encrypted:

```
  shield.transport.ssl: true
```

3. Enable SSL on the HTTP layer to ensure that communication between HTTP clients and the cluster is encrypted:

```
  shield.http.ssl: true
```

4. Restart Elasticsearch so these configuration changes take effect.

5. Copy and install the SSL certificate to all agent and web servers.

4.8 Configuring Kibana to work with Shield
When you install or update to Relativity 9.6, Shield is enabled by default. With Shield on by default, other plugins like Marvel or Head are not supported. In order to use your other plugins, you need to provide the Kibana server with credentials so it can access the .kibana index and monitor the cluster.

To configure credentials for the Kibana server:

1. Assign the `kibana4_server` role to a user in Shield. For more information, see Configuring a Role for the Kibana 4 Server in the Shield documentation.

2. Set the `kibana_elasticsearch_username` and `kibana_elasticsearch_password` properties in `kibana.yml` to specify the credentials of the user you assigned to the `kibana4_server` role:
3. Update the following setting in kibana.yml to false:

<table>
<thead>
<tr>
<th>Setting</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>kibana_elasticsearch_username</td>
<td>kibana4-user</td>
</tr>
<tr>
<td>kibana_elasticsearch_password</td>
<td>kibana4-password</td>
</tr>
<tr>
<td>elasticsearch.ssl.verify</td>
<td>false</td>
</tr>
</tbody>
</table>

Kibana 4 users also need access to the .kibana index so they can save and load searches, visualizations, and dashboards. For more information, see Configuring Roles for Kibana 4 Users in the Shield documentation.
5 Monitoring Elasticsearch

There are several tools external to Relativity that you can use to monitor and manage a Data Grid cluster. This page lists several of the most useful tools available with brief overviews of their functionality, installation instructions, and links to further documentation.

5.1 Head

Head shows the overall health of the cluster, which nodes are connected, and the health of the indices. Using head, perform queries from the head console that retrieve JSON results. The head plugin is easily accessible and doesn't require extensive setup.

Head uses three color-coded statuses to report on shard allocation:

- **green** - all shards are allocated.
- **yellow** - the primary shards are allocated, but replicas are not.
- **red** - the primary shard isn't allocated in the cluster.

A red status indicates that the specific primary shard is not allocated in the cluster, yellow means that the primary shard is allocated but replicas are not, and green means that all shards are allocated. The index level status is controlled by the worst shard status. The cluster status is controlled by the worst index status, so what this means is that if a shard or index is red, the cluster will be red and so on. Grey means that the cluster is not connected.

5.2 Marvel

All customers of Relativity that implement Data Grid may receive Marvel licenses. Marvel allows you to view your cluster status in a simple, single pane overview; investigate highly detailed system metrics including Apache Lucene; visualize cluster events and metrics together to analyze how changes affect performance; and access the REST API.
Note: You must install Marvel on each node in the cluster in order for it to properly work. Marvel's default settings are configured to store data in the same cluster that it monitors. If you use Marvel to monitor a production cluster, you must send the data to a secondary monitoring cluster. See this site for more information on setting up a secondary monitoring cluster.

The Marvel plugin provides the following features:

- **Clusters** - The clusters page shows you the clusters you’re monitoring and allows you to select them.

- **Overview dashboard** - The overview dashboard displays a quick look at your cluster performance. The four counters are an aggregate of the entire cluster. If there are any performance issues, investigate the individual components of the cluster.

- **Nodes** - The nodes section lists all of the nodes that are in the cluster that are being monitored, as well as information on CPU usage, memory usage, disk free space, and the shard they contain. From the nodes section, you can select individual nodes for more detailed information about how those nodes are using the compute allocated to them as well as all indices and shards on the node.

- **Indices** - Indices show the same counters as the overview dashboard, but allow you to get more information on individual indices. Some indices on your cluster will be larger than others and require more monitoring. The indices page shows the size of indices from a data and document count perspective and the index and search rate for each individual index. From here, you can select individual indices to see the state of different shards (primary, replica, relocating, initializing and unassigned) and additional information which can be used to monitor performance.

Use the following steps to install Marvel and Sense on the monitoring cluster.

1. Open the command prompt and navigate to the kibana/bin folder.
2. Enter the following command to install Marvel:

   ```bash
   kibana plugin --install elasticsearch/marvel/2.1.2
   ```
3. Enter the following command to install Sense:

```bash
timeout 150 kibana plugin --install elastic/sense
```


### 5.2.0.1 Performing offline installations of Marvel

Use the following steps to download and install Marvel on a machine that doesn't have access to the internet.

1. Download the installer from this URL on a machine that has internet:

```bash
http://download.elasticsearch.org/elasticsearch/marvel/marvel-latest.zip
```

2. Save the zip file on the C: drive of your cluster node.

3. Open a command prompt and execute the following command:

```bash
C:\RelativityDataGrid\elasticsearch-main\bin\plugin -i marvel -u file:///C:/marvel-latest.zip
```

See [this site](#) for more information on using the Marvel plugin.

### 5.3 Curator

The Curator Python API allows you to manage indexes and snapshots with the following features:

- **Iterative methods** - allow you to retrieve data across the cluster within specified parameters.
- **Non-iterative methods** - allow you to retrieve data within a single index or snapshot.
- **Helper methods** - allow you to retrieve values required to complete iterative and non-iterative methods.

You can install Curator on any machine that has access to the cluster. Use the following steps to install Python and Curator:

1. Download and install the latest version of Python from [this site](#).

   **Note:** Verify that the version of Python you choose to install includes pip.

2. Enter the following command in the Python command line interface (python.exe):

   ```bash
   pip install elasticsearch-curator
   ```

### 5.3.1 Cluster metrics to monitor

There are certain metrics that you should always monitor such as availability and status, indexing rate, RAM usage, etc. Review the following sites for information on metrics to monitor:
See this site for an extensive list of critical metrics.

See this site for an Elasticsearch guide on monitoring individual nodes.

5.3.2 Monitoring bulk rejections

It is important that you monitor bulk rejections. You can access information on bulk rejections through Sense.

Use the following query to access information on bulk rejections for the cluster:

```
GET /_cat/thread_pool?v&h=id,host,bulk.active,bulk.rejected,bulk.completed
```

If you see a message that reads "BULK THREAD POOL REJECTED", or if BULK THREAD POOL QUEUE SIZE rises above your specified limit, increase the default of 50 to 300 or reduce the writes to Data Grid.

5.3.3 Monitoring backup and recover times

It's important to monitor backup and recovery times to properly set RTO and RPO expectations.

Use the following queries to access information on backup and restore operations:

```
GET /_cat/snapshots/{snapshotname}?v
```

```
GET _cat/recovery?v
```

5.3.4 Additional cluster monitoring information

Use the following queries to access additional monitoring information from the cluster that may be useful to your team:

```
#Cluster information
Get _cat/health?v
```

```
#Node information
GET _cat/nodes?v
```

```
GET _cat/nodes?v&h=host,name,node.role,version,build,jdk,uptime
```

```
GET _cat/nodes?v&h=name,disk.avail,heap.percent,ram.percent,files_percent,merges.total,segments.count,segments.memory
```

```
GET _cat/allocation?v
```

```
#Index information
Get _cat/indices?v
```

```
Get _cat/indices?v&h=index,tm
```
5.4 Querying with Sense

This page provides a list of the most commonly used queries.

This page contains the following sections:

- Creating and deleting indexes below
- Retrieving documents and cluster settings below
- Shutting down a cluster on page 46
- Backing up clusters on page 47

5.4.1 Creating and deleting indexes

Use the following query to create an index:

POST /index_name

Use the following query to delete an index:

DELETE /index_name*

5.4.2 Retrieving documents and cluster settings

Use the following query to retrieve the settings and fields for document mapping:

GET /relativity_relativity_edds1019984_10/_mapping/document

Use the following query to retrieve the settings for the specified index:

GET /relativity_relativity_edds1019984_10/_settings

Use the following query to retrieve the total count of documents in the specified index:

GET /relativity_relativity_edds1019984_10/document/_count
{
    "query": {
        "match_all": {}
    }
}

Use the following query to retrieve the top five documents in the specified index:
Use the following query to retrieve the documents containing a specified term:

```json
POST /relativity_relativeity_edds1019984_10/document/_search
{
    "query": {
        "query_string": {
            "query": "happy",
            "fields": ["Fields.ExtractedText.Value"
        ]
    }
}
```

Use the following queries to retrieve cluster health information:

```bash
GET cat/health?v
GET _cluster/settings
```

Use the following query to retrieve all nodes detected by the master node:

```bash
GET _cat/nodes?v
```

Use the following query to retrieve information on the allocation of shards:

```bash
GET _cat/allocation?v
```

Use the following queries to retrieve the state of all indexes in the cluster:

```bash
GET _cat/indices
GET _cat/indices?v
GET _cat/indices/logstash*
GET _cat/indices/index_name*
```

Use the following queries to retrieve all aliases:

```bash
GET _cat/aliases
GET _cat/aliases/relativity_integration_edds9999*
```

Use the following query to update an alias:

```bash
```
POST /aliases
{
    "actions": [
        {
            "add": {
                "index": "relativity_integration_f176bd64-7926-4d59-9838-a88f3dfe9ab7_edds9999_10",
                "alias": "relativity_integration_edds9999_10_read"
            }
        }
    ]
}

Use the following queries to retrieve and read templates:

GET /_template/

GET /_template/kcuratemplate

GET /_template/audit

Use the following commands to perform maintenance tasks on the cluster:

POST _flush/sync?wait_if_ongoing=true

POST _refresh

POST /relativity_integration_484a7590-3d90-4215-b5ab-e29aa91b99b8_edds9999_10/_refresh

POST _optimize

### 5.4.3 Shutting down a cluster

To shut down a cluster, copy and paste the following template into the Sense plugin of Kibana and follow the instructions:

```
#########################################
# Reboot procedure for data nodes
#########################################

#########################################
# Step 1
# Disable or delete all Audit Migration agents.
# Step 2
# Check the cluster health.
GET _cluster/health
# Note the cluster health should be green for "status".
# Take note of the number of Data Nodes in "number_of_data_nodes".

# Step 3
# Perform a flush synced
POST _flush/synced
# Ensure there are no failed shards for the sync operation
```
# Step 4
# Turn off shard allocation

PUT _cluster/settings
{
  "persistent": {
    "cluster.routing.allocation.enable": "none"
  }
}

# Step 5
# Perform maintenance on the node, for example Windows updates, and restart.

# Step 6
# Check the cluster health and wait for the "number_of_data_nodes" to be the same as step 2. You may need to run the command multiple times as the node comes online.

GET _cluster/health
# Note: After the data node comes online, the cluster will remain in "status" yellow

# Step 7
# Turn shard allocation back on.

PUT _cluster/settings
{
  "persistent": {
    "cluster.routing.allocation.enable": "all"
  }
}

# Step 8
# Check the cluster health and ensure it is green.

GET _cluster/health
# Note you may need to run the command multiple times until the "active_shards_percent_as_number" is 100

# Step 9
# Repeat these steps for the remaining nodes that need maintenance.

# Additional notes

# Single node clusters do not need to go through these steps. They can be rebooted once the migration agents are turned off.

# Clusters with single master nodes will go down completely when the master node is rebooted. When a single master node is used, make sure migration agents are turned off and reboot the master node.

# For clusters with multiple masters: Reboot each master in turn and allow them to come back up before rebooting the next master. This can be done with migration agents active.

# Client nodes can be rebooted without the above steps as long as migration is not active.

## 5.4.4 Backing up clusters

Use the following commands for backup and restore operations:

Use the following command to get all backup repositories:
GET /_snapshot/_all

Use the following command to create a backup repository:

```
PUT /_snapshot/my_backup
{
   "type": "fs",
   "settings": {
      "location": "/shares/backups/my_backup",
      "compress": true
   }
}
```

Use the following command to create a backup:

```
PUT /_snapshot/my_backup/snapshot_1
{
   "indices": "relativity_test_loadfile,trial_loadfile",
   "ignore_unavailable": "true",
   "include_global_state": false
}
```

Use the following command to check running status of a snapshot:

```
GET /_snapshot/_status
```

Use the following command to check the status of a specific snapshot even if not running:

```
GET /_snapshot/my_backup/snapshot_1/_status
```

Use the following command to restore a backup:

```
POST /_snapshot/my_backup/snapshot_1/_restore
{
   "indices": "relativity_test_loadfile,trial_loadfile",
   "ignore_unavailable": "true",
   "include_global_state": false,
   "rename_pattern": "relativity_test_(.+)",
   "rename Replacement": "restored_$1"
}
```
6 Backing up Relativity Data Grid

Data Grid now stores text data in your file share. There is no longer automatic redundant storage of document text. Because Data Grid text data is now stored in your file share, you may need to adjust your file system data backup frequency to meet the requirements of your Service Level Agreements (SLAs) and disaster recovery plans.

6.1 Backing up Elasticsearch

Click to expand instructions for backing up Elasticsearch. We recommend you routinely back up your data. Elasticsearch replicas provide high availability during run time, allowing toleration of sporadic node loss without interruption of service, but replicas don’t provide protection against catastrophic failures. Create a complete backup of the entire cluster to protect your data if something goes wrong.

You can use the snapshot API to create a backup of the cluster. The snapshot API saves the current state of all data in your cluster to a shared repository. The first snapshot you create is a complete copy of all data on the cluster. Each subsequent snapshot compares the current state of the data in the cluster to the data stored in the repository and only modifies the differences between the two. The snapshot API incrementally edits the repository each time you create a new snapshot, so subsequent backups are significantly faster since they require less data transmission. This page explains all steps necessary to back up and restore Elasticsearch.

**Note:** Data Grid supports Windows servers only.

6.2 Creating a repository

Before implementing this backup method, you must create a repository that can store snapshots. You can use any of the following four repository types:

- Shared file system, such as a NAS
- Amazon S3
- Hadoop Distributed File System (HDFS)
- Azure Cloud

Use the following steps to create and share a folder:

2. Right-click on the folder, and then click Properties.
3. Select the Sharing tab, and then click Share.
4. Enter the user that runs the Elasticsearch Windows service (domain\account), and then click **Add**.

5. Select the user on the share list and set the **Permission Level** to **Co-owner**.
6. Click **Share**.
7. When the share completes, click **Done**.
8. On the **Document Properties** dialog, select the **Security** tab.
9. Verify that the user that runs the Elasticsearch Windows service has **Full Control** security permissions to the folder.

Use the following steps to link Elasticsearch to the repository folder:

1. Launch **Marvel** from within a browser to connect to one of the nodes in your cluster.
2. Launch **Sense** from the Dashboards drop-down near the top right.

3. Edit the location value and run the following to set up a shared file system repository:

```bash
PUT /_snapshot/my_backup
{
    "type": "fs",
```
4. Verify your snapshot settings exist by performing the following call:

GET /_snapshot/

6.3 Creating snapshots

There are two ways to create snapshots:

- Creating snapshots manually from within Sense below
- Scheduling a Windows task using Curator on the next page

6.3.1 Creating snapshots manually from within Sense

Run the following to back up all open indexes into a snapshot named "snapshot_1".

PUT /_snapshot/ElasticBackup/snapshot_1

**Note:** Increment the name of the snapshot for best results (e.g., snapshot_1, snapshot_2, snapshot_3, etc.). All alphabetical characters in the snapshot name must be lowercase.

Verify that this process created a backup by navigating to the following location:

//COMPUTER_NAME.business.corp/Shared/ElasticBackup

Your backup should look similar to the following image:
For more information on snapshot commands, including the ability to snapshot specific indexes, see the Backing up your cluster article on the Elasticsearch website for your version of Elasticsearch.

### 6.3.2 Scheduling a Windows task using Curator

The best way to schedule automatic backups of your data is to use Curator, which you can combine with scheduled tasks to automatically invoke the desired behavior.

The Curator Python API can be used to manage indexes and snapshots with the following features:
- **Iterative methods** - allow you to retrieve data across the cluster within specified parameters.
- **Non-iterative methods** - allow you to retrieve data within a single index or snapshot.
- **Helper methods** - allow you to retrieve values required to complete iterative and non-iterative methods.

For more information on Curator and snapshot capabilities, please see the following website:

### 6.3.2.1 Installing Curator 4

Before setting up Curator, you must complete the following:

1. Download and run the [Curator installer from Elastic](https://www.elastic.co/downloads/curator).

### 6.3.2.2 Running Curator

Once you install Curator, you can use it to run “actions” which are created in the action.yml file. Use the following command to run an operation in the action.yml file:

**Note**: This command runs a dry-run where Curator simulates the action(s) in the file without making any changes. To actually run the operation, remove the `--dry-run` flag.

```powershell
curator [--config CONFIG.YML] [--dry-run] ACTION_FILE.YML
```

This command references both a configuration and action file. You can run PowerShell scripts to automatically create the files needed to run Curator. For more information, see Running Curator from PowerShell below. If you want to manually set up Curator, see Running Curator manually on page 63.

### Running Curator from PowerShell

The following PowerShell scripts automatically create the PS1 and YML files needed to run Curator. It also sets up three Windows tasks: backup, backup cleanup, and Marvel cleanup.

Click to expand the Curator setup without SSL

Run the following in Powershell:

```powershell
<##>
Param(
    $probHostName = "masternodename",
    $marvelHostName = "marvelnodename",
    $repositoryName = "datagridbackup",
    [string]$esUsername = "esadmin",
    [string]$esPassword = "esadmin",
    [string]$marvelUsername = "marvel",
    [string]$marvelPassword = "marvel",
    $curatorMsiPath = "C:\Curator\elasticsearch-curator-4.2.6-win32.msi",
    $curatorPath = "C:\Curator",
    $useSSL = "False"
)

$httpAuth = [system.Text.Encoding]::ASCII
$httpAuth = "$esUsername`:$esPassword"
$httpAuthMarvel = [system.Text.Encoding]::ASCII
$httpAuthMarvel = "$marvelUsername`:$marvelPassword"
$powerShellPath = (Resolve-Path /Windows/Systems32/WindowsPowerShell/v1.0/).Path
```
function IsCuratorInstalled {
    $isInstalled = ((Get-ChildItem "HKLM:\Software\Microsoft\Windows\CurrentVersion\Uninstall") | Where-Object { $_.GetValue( "DisplayName" ) -eq "elasticsearch-curator" })

    If (!$isInstalled) {
        Try{
            Write-Host "Installing Curator 4.2.6" -ForegroundColor Green
            Start-Process msiexec.exe -Wait -ArgumentList "/I $curatorMsiPath /quiet" -ErrorAction Stop
            Write-Host "Installation Complete." -ForegroundColor Green
        } Catch{
            $ErrorMessage = $_.Exception.Message; Break
        }
    } else{
        Write-Host "Curator is already installed." -ForegroundColor Green
    }
}

function MakeFiles {
    function MakeProdConfigYML {
        $prodConfigYml = @"---
        # Remember, leave a key empty if there is no value. None will be a string,
        # not a Python "NoneType"
        client:
            hosts:
                - $probHostName
            port: 9200
            url_prefix:
            use_ssl: $useSSL
            certificate:
            client_cert:
            client_key:
            ssl_no_validate: True
            http_auth: $httpAuth
            timeout: 30
            master_only: False
        logging:
            loglevel: INFO
            logfile:
            logformat: default
            blacklist: ['elasticsearch', 'urllib3']
        "@

        $prodConfigYml | Out-File .\prodConfig.yml -Encoding ascii
    }
    MakeProdConfigYML

    function MakeBackupActionYML {
        $prodBackupYml = @"---
        # Remember, leave a key empty if there is no value. None will be a string,
        # not a Python "NoneType"
        # Also remember that all examples have 'disable_action' set to True. If you
        # want to use this action as a template, be sure to set this to False after
        "@"
# copying it.
# Leaving name blank will result in the default 'curator-%Y%m%d%H%M%S'
actions:
  1:
    action: snapshot
    description: "Created snapshots for all indexes."
    options:
      repository: $repositoryName
      name: ignore_unavailable: False
      include_global_state: True
      partial: False
      wait_for_completion: True
      skip_repo_fs_check: False
      timeout_override: continue_if_exception: False
      disable_action: False
      filters:
        - filtertype: none
      
"@

$prodBackupYml | Out-File ./prodBackup.yml -Encoding ascii
}

MakeBackupActionYML

function MakeCleanUpActionYML
{
$prodCleanUpYml = @"
# Remember, leave a key empty if there is no value. None will be a string,
# not a Python "NoneType"
#
# Also remember that all examples have 'disable_action' set to True. If you
# want to use this action as a template, be sure to set this to False after
# copying it.
# Leaving name blank will result in the default 'curator-%Y%m%d%H%M%S'
actions:
  1:
    action: delete_snapshots
    description: "Deletes snapshots older than 14 days."
    options:
      repository: $repositoryName
      retry_interval:
      retry_count:
      timeout_override:
      continue_if_exception: False
      disable_action: False
      filters:
        - filtertype: pattern
          kind: prefix
          value: curator-
          exclude:
        - filtertype: age
          source: creation_date
          direction: older
          unit: days
          unit_count: 14
          exclude:
        
"@

$prodCleanUpYml | Out-File ./prodCleanUp.yml -Encoding ascii
function MakeMarvelConfigYML
{
$marvelConfigYml = @"
---
# Remember, leave a key empty if there is no value. None will be a string,
# not a Python "NoneType"
client:
  hosts: - $marvelHostName
  port: 9200
  url_prefix:
    use_ssl: False
    certificate:
      client_cert:
      client_key:
      ssl_no_validate: False
  http_auth: $httpAuthMarvel
  timeout: 30
  master_only: False

logging:
  loglevel: INFO
  logfile:
  logformat: default
  blacklist: ['elasticsearch', 'urllib3']
"@

$marvelConfigYml | Out-File .\marvelConfig.yml -Encoding ascii
}

function MakeMarvelCleanUpYML
{
$marvelCleanUpYml = @"
# Remember, leave a key empty if there is no value. None will be a string,
# not a Python "NoneType"
# Also remember that all examples have 'disable_action' set to True. If you
# want to use this action as a template, be sure to set this to False after
# copying it.
# Leaving name blank will result in the default 'curator-%Y%m%d%H%M%S'
actions:
  1:
    action: delete_indices
    description: "Deletes old marvel indexes"
    options:
      timeout_override:
      continue_if_exception: False
      disable_action: False
    filters:
      - filtertype: pattern
        kind: regex
        value: .marvel-
        exclude:
      - filtertype: age
        source: creation_date
        direction: older
unit: days
unit_count: 30
exclude:
"

$marvelCleanUpYml | Out-File .\marvelCleanUp.yml -Encoding ascii
} MakeMarvelCleanUpYML

function MakeBackupActions
{
$prodBackUp = @"cd\cd "."Program Files\elasticsearch-curator" .\curator --config $curatorPath\prodConfig.yml $curatorPath\prodBackUp.yml "@

$prodBackUp | Out-File .\prodBackup.ps1
} MakeBackupActions

function MakeCleanUpActionProd
{
$prodCleanUpPs = @"cd\cd "."Program Files\elasticsearch-curator" .\curator --config $curatorPath\prodConfig.yml $curatorPath\prodCleanUp.yml "@

$prodCleanUpPs | Out-File .\prodCleanup.ps1
} MakeCleanUpActionProd

function MakeCleanUpActionMarvel
{
$marvelCleanUp = @"cd\cd "."Program Files\elasticsearch-curator" .\curator --config $curatorPath\marvelConfig.yml $curatorPath\marvelCleanUp.yml "@

$marvelCleanUp | Out-File .\marvelCleanup.ps1
} MakeCleanUpActionMarvel

function MakeFiles
{
MakeFiles

<# Make scheduled tasks. #>

function MakeScheduledTaskDataGridBackup
{
$action = New-ScheduledTaskAction -Execute ($"$powerShellPath" + "powershell.exe") -WorkingDirectory $curatorPath -Argument "$prodBackUp.ps1"
$trigger = New-JobTrigger -Once -At (Get-Date).date -RepeatIndefinitely -RepetitionInterval (New-TimeSpan -Hours 6)
$settings = New-ScheduledTaskSettingsSet -AllowStartIfOnBatteries -DontStopIfGoingOnBatteries -WakeToRun Register-ScheduledTask -Action $action -Trigger $trigger -RunLevel Highest -TaskName "Data Grid Backup" -Description "Scheduled snapshot of the Production Data Grid cluster." -Settings $settings
Click to expand the Curator setup with SSL

Run the following in Powershell:

```powershell
<##>
Param(
    $probHostName = "dg-ramp-01",
    $marvelHostName = "dg-ramp-agt",
    $repositoryName = "datagridbackup",
    [string]$esUsername = "esadmin",
    [string]$esPassword = "esadmin",
    [string]$marvelUsername = "marvel",
    [string]$marvelPassword = "marvel",
    $curatorMsiPath = "C:\Curator\elasticsearch-curater-4.2.6-win32.msi",
    $curatorPath = "C:\Curator",
    $useSSL = "True"
)

$httpAuth = [system.Text.Encoding]::ASCII
$httpAuth = "$esUsername:$esPassword"
$httpAuthMarvel = [system.Text.Encoding]::ASCII
$httpAuthMarvel = "$marvelUsername:$marvelPassword"
$powerShellPath = (Resolve-Path /Windows/System32/WindowsPowerShell/v1.0).Path

function IsCuratorInstalled {
    $isInstalled = ((Get-ChildItem "HKLM:\Software\Microsoft\Windows\CurrentVersion\Uninstall") | Where-Object { $_.GetValue( "DisplayName" ) -eq "elasticsearch-curater" })

    If (!$isInstalled) { Try{
```
Write-Host "Installing Curator 4.2.6" -ForegroundColor Green
Start-Process msiexec.exe -Wait -ArgumentList "/I $curatorMsiPath /quiet" -ErrorAction Stop
Write-host "Installation Complete."

)Catch{
    $ErrorMessage = $_.Exception.Message; Break
}
)else{
    Write-Host "Curator is already installed." -ForegroundColor Green
}

IsCuratorInstalled

function MakeFiles 
{
    function MakeProdConfigYML 
    {
        $prodConfigYml = @"
        # Remember, leave a key empty if there is no value. None will be a string,
        # not a Python "NoneType"
        client:
            hosts:
                - $probHostName
            port: 9200
            url_prefix:
            use_ssl: $useSSL
            certificate:
            client_cert:
            client_key:
            ssl_no_validate: True
            http_auth: $httpAuth
            timeout: 30
            master_only: False

        logging:
            loglevel: INFO
            logfile:
                logformat: default
                blacklist: ['elasticsearch', 'urllib3']
        "@

        $prodConfigYml | Out-File .\prodConfig.yml -Encoding ascii
    }
    MakeProdConfigYML

    function MakeBackupActionYML 
    {
        $prodBackupYml = @"
        # Remember, leave a key empty if there is no value. None will be a string,
        # not a Python "NoneType"
        # Also remember that all examples have 'disable_action' set to True. If you
        # want to use this action as a template, be sure to set this to False after
        # copying it.
        # Leaving name blank will result in the default 'curator-%Y%m%d%H%M%S'
        actions:
            1:
                action: snapshot
                description: "Created snapshots for all indexes."
                options:
        "@
repository: $repositoryName
name:
ignore_unavailable: False
include_global_state: True
partial: False
wait_for_completion: True
skip_repo_fs_check: False
timeout_override:
continue_if_exception: False
disable_action: False
filters:
  - filtertype: none
"@

$prodBackupYml | Out-File \prodBackup.yml -Encoding ascii
}
MakeBackupActionYML

function MakeCleanUpActionYML
{
$prodCleanUpYml = ""
# Remember, leave a key empty if there is no value. None will be a string,
# not a Python "NoneType"
#
# Also remember that all examples have 'disable_action' set to True. If you
# want to use this action as a template, be sure to set this to False after
# copying it.
# Leaving name blank will result in the default 'curator-%Y%m%d%H%M%S'
actions:
  1:
    action: delete_snapshots
    description: "Deletes snapshots older than 14 days."
    options:
      repository: $repositoryName
    retry_interval:
    retry_count:
    timeout_override:
    continue_if_exception: False
    disable_action: False
    filters:
      - filtertype: pattern
        kind: prefix
        value: curator-
        exclude:
      - filtertype: age
        source: creation_date
        direction: older
        unit: days
        unit_count: 14
        exclude:
"@

$prodCleanUpYml | Out-File \prodCleanUp.yml -Encoding ascii
}
MakeCleanUpActionYML

function MakeMarvelConfigYML
{
$marvelConfigYml = ""
---
Remember, leave a key empty if there is no value. None will be a string, not a Python "NoneType"

client:
  hosts:
    - $marvelHostName
  port: 9200
  url_prefix:
  use_ssl: False
  certificate:
  client_cert:
  client_key:
  ssl_no_validate: False
  http_auth: $httpAuthMarvel
  timeout: 30
  master_only: False

logging:
  loglevel: INFO
  logfile:
  logformat: default
  blacklist: ['elasticsearch', 'urllib3']

"@

$marvelConfigYml | Out-File .\marvelConfig.yml -Encoding ascii
}
MakeMarvelConfigYML

function MakeMarvelCleanUpYML
{
    $marvelCleanUpYml = @"
    # Remember, leave a key empty if there is no value. None will be a string, not a Python "NoneType"
    # Also remember that all examples have 'disable_action' set to True. If you want to use this action as a template, be sure to set this to False after copying it.
    # Leaving name blank will result in the default 'curator-%Y%m%d%H%M%S'
    actions:
        1:
          action: delete_indices
          description: "Deletes old marvel indexes"
          options:
            timeout_override:
            continue_if_exception: False
            disable_action: False
          filters:
            - filtertype: pattern
              kind: regex
              value: .marvel-.*
              exclude:
            - filtertype: age
              source: creation_date
              direction: older
              unit: days
              unit_count: 30
              exclude:

    "@

$marvelCleanUpYml | Out-File .\marvelCleanUp.yml -Encoding ascii
}
function MakeBackupActions
{
$prodBackUp = @
cd "\Program Files\elasticsearch-curateur"
.curator --config $curatorPath\prodConfig.yml $curatorPath\prodBackUp.yml "@

$prodBackUp | Out-File \prodBackup.ps1
}
MakeBackupActions

function MakeCleanUpActionProd
{
$prodCleanUpPs = @
cd "\Program Files\elasticsearch-curateur"
.curator --config $curatorPath\prodConfig.yml $curatorPath\prodCleanUp.yml "@

$prodCleanUpPs | Out-File \prodCleanup.ps1
}
MakeCleanUpActionProd

function MakeCleanUpActionMarvel
{
$marvelCleanUp = @
cd "\Program Files\elasticsearch-curateur"
.curator --config $curatorPath\marvelConfig.yml $curatorPath\marvelCleanUp.yml "@

$marvelCleanUp | Out-File \marvelCleanup.ps1
}
MakeCleanUpActionMarvel

function MakeScheduledTaskDataGridBackup
{
$action = New-ScheduledTaskAction -Execute ("$powershellPath" + "powershell.exe") -WorkingDirectory $curatorPath -Argument ".\prodBackUp.ps1"
$trigger = New-JobTrigger -Once -At (Get-Date).date -RepeatIndefinitely -RepetitionInterval (New-TimeSpan -Hours 6)
$settings = New-ScheduledTaskSettingsSet -AllowStartIfOnBatteries -DontStopIfGoingOnBatteries -WakeToRun
Register-ScheduledTask -Action $action -Trigger $trigger -RunLevel Highest -TaskName "Data Grid Backup" -Description "Scheduled snapshot of the Production Data Grid cluster." -Settings $settings
}
MakeScheduledTaskDataGridBackup

function MakeScheduledTaskDataGridBackupCleanup
{
$action = New-ScheduledTaskAction -Execute ($"powerShellPath" + "powershell.exe") -WorkingDirectory $curatorPath -Argument ".\prodCleanUp.ps1"
$trigger = New-JobTrigger -Once -At (Get-Date).date -RepeatIndefinitely -RepetitionInterval (New-TimeSpan -Hours 24)
$settings = New-ScheduledTaskSettingsSet -AllowStartIfOnBatteries -DontStopIfGoingOnBatteries -WakeToRun Register-ScheduledTask -Action $action -Trigger $trigger -RunLevel Highest -TaskName "Data Grid Backup Cleanup" -Description "Scheduled clean up job for the Production Data Grid cluster."
)

MakeScheduledTaskDataGridBackupCleanup

function MakeScheduledTaskMarvelCleanup
{

$action = New-ScheduledTaskAction -Execute ($"powerShellPath" + "powershell.exe") -WorkingDirectory $curatorPath -Argument ".\marvelCleanUp.ps1"
$trigger = New-JobTrigger -Once -At (Get-Date).date -RepeatIndefinitely -RepetitionInterval (New-TimeSpan -Hours 24)
$settings = New-ScheduledTaskSettingsSet -AllowStartIfOnBatteries -DontStopIfGoingOnBatteries -WakeToRun Register-ScheduledTask -Action $action -Trigger $trigger -TaskName "Marvel CleanUp" -Description "Deletes indexes on the marvel cluster older than 30 days."
}

MakeScheduledTaskMarvelCleanup

Running Curator manually
The following sections contain examples of the different action files you can create manually to run in Curator. You can save these YML files anywhere. Ensure you use a full path to the files when executing the command.

Sample configuration file
Click to expand a sample config.yml file containing one master node.

---

# Remember, leave a key empty if there is no value. None will be a string,
# not a Python "NoneType"
client:
  hosts:
  - cs-mv1-agtfs
  port: 9200
  url_prefix:
  use_ssl: False
  certificate:
  client_cert:
  client_key:
  ssl_no_validate: False
  http_auth:
  timeout: 30
  master_only: False

logging:
  loglevel: INFO
  logfile:
  logformat: default
  blacklist: ['elasticsearch', 'urllib3']

Sample backup action file
Click to expand a sample action_file.yml file that runs a backup action.
Remember, leave a key empty if there is no value. None will be a string, not a Python "NoneType"

Also remember that all examples have 'disable_action' set to True. If you want to use this action as a template, be sure to set this to False after copying it. Leaving name blank will result in the default 'curator-%Y%m%d%H%M%S' actions:

```
1:
  action: snapshot
  description: "description"
  options:
    repository: datagridbackup
    name: 
    ignore_unavailable: False
    include_global_state: True
    partial: False
    wait_for_completion: True
    skip_repo_fs_check: False
    timeout_override: 
    continue_if_exception: False
    disable_action: False
  filters:
    - filtertype: none
```

**Sample restore action file**

Click to expand a sample action_file.yml file that runs a restore action.

```
# Remember, leave a key empty if there is no value. None will be a string, not a Python "NoneType"
#
# Also remember that all examples have 'disable_action' set to True. If you want to use this action as a template, be sure to set this to False after copying it.
# Leaving name blank will result in restoring the most recent snapshot by age
# Leaving indices blank will result in restoring all indices in the snapshot
actions:
  1:
    action: close
    description: "Close selected indices"
    options:
      delete_aliases: False
      timeout_override: 
      continue_if_exception: False
      disable_action: False
    filters:
      - filtertype: none
  2:
    action: restore
    description: "restore"
    options:
      repository: datagridbackup
      # Leaving name blank will result in restoring the most recent snapshot by age
      name: 
      # Leaving indices blank will result in restoring all indices in the snapshot
      indices:
        include_aliases: False
        ignore_unavailable: False
        include_global_state: True
        partial: False
        rename_pattern:
```
6.3.2.3 Backup script with email service

You can also use Curator to send an email if an action failed or succeeded.

The script in the example uses a global variable to run and will not work if credentials are needed and it is run outside of the ISE, which contains that global variable. To create a global variable for your ISE session, run the following:

```
$global:cred = get-credential (Run this to create a global variable for your ISE session)
```

Click to expand a sample backup script with email service

**Note:** Ensure you use the format email@domain.com or the email server will not recognize you.

```
cd "C:\Program Files\elasticsearch-curator"

 CD \Program Files\elasticsearch-curator

.task.execute backup:

 .\curator.exe --config "C:\Users\csadmin\Desktop\Curator\Configuration.yml" "C:\Users\csadmin\Desktop\Curator\Backup.yml" | Out-File C:\Logs.txt

 $Job = (Get-Content C:\Logs.txt | Select-String "successfully completed")
 $complete = $Job.Contains("Job completed.")

 $Job = (Get-Content C:\Logs.txt | Select-String "Snapshot FAILED")
 $failed = $Job.Contains("FAILED")

 if($failed) {
     $body = "A Data Grid Backup Job has failed check the backup job or email support@relativity.com to create a ticket"
     Send-MailMessage -From "Jane Smith <jsmith@example.com>" -To "Jane Smith <jsmith@example.com>" -Subject "A Data Grid backup job has failed." -Body $body -BodyAsHTML -SmtpServer smtp.office365.com -credential $cred -UseSsl -Port 587
 } else {
     $body = "The Data Grid backup job has completed."
     Send-MailMessage -From "Jane Smith <jsmith@example.com>" -To "Jane Smith <jsmith@example.com>" -Subject "A Data Grid backup job has is successful." -Body $body -BodyAsHTML -SmtpServer smtp.office365.com -credential $cred -UseSsl -Port 587
 }

 Write-Output("Done")
```
6.3.2.4 Setting the script as a scheduled task

Use the following steps to set the script as a scheduled task:

1. Click Start > Administrative Tools > Windows Task Scheduler on the system that runs scheduled tasks.

2. In the Task Scheduler, click Create Task under Actions on the right.

3. Enter a name and description for the task. (Entering a description is optional.)

4. Navigate to the General tab, and then select Security Options.

5. Specify the user account that runs scheduled tasks. The account can be the same one that runs the Elasticsearch Windows service.

6. Edit the settings to run tasks regardless of whether or not the user is logged in.

7. Navigate to the Triggers tab, and then click New to add a new trigger for the scheduled task.

8. Verify that the Begin the Task field is set to On a schedule, and then set the start date to your preferred time.

9. Set the frequency to be every one hour if you’re unsure what your recovery point objective goals are.

   **Note:** Relativity stores the last 90 days of audits for each workspace in SQL Server. Long text fields, like extracted text, are usually never edited post import.

10. Set the duration of the task to run indefinitely.
11. Click OK.
12. The following example has the task running every hour indefinitely:

13. Navigate to the Actions tab, and then click New.

14. Set the Action to Start a program.

15. In the Program/script field, enter "Powershell."

16. In the Add arguments (optional) field, enter the following value:

   ```
   .\[Your PowerShell Script Name]
   ```

   For example, if your PowerShell script is named "Migration1.ps1" then you would enter ".\Migration1.ps1" as the value.

17. In the Start in (optional) field, add the location of the folder that contains your PowerShell script. In this example, the script directory is C:\Script. The location entered in the Start in box also stores the scheduled task run times, the job history for the copies, and any additional logging that may occur.
18. Click **OK** after configuring your preferred settings.

19. Set any other preferred settings in the **Conditions and Settings** tabs. You can also set up an additional action to email an system admin each time the script runs.

20. Click **OK**.

When you complete these steps, the task runs according to your settings.

### 6.4 Restoring a snapshot

There are multiple methods for restoring snapshots. Restoring snapshots from the Elasticsearch head console is the recommended procedure, but you can also use cURL to restore snapshots. The Elasticsearch website's documentation relies heavily on cURL commands for snapshot restoration. Brief descriptions of both methods are provided here.

#### 6.4.1 Restoring snapshots from the Elasticsearch head console

You can restore a snapshot from the Elasticsearch head console. Use the following steps to restore a snapshot with this method:

2. Expand the **Query** tab.
3. Enter the following URL in the first field: **http://localhost:9200/**
4. Enter **_search** in the second field, and use the drop-down menu to select **GET**.
5. Enter the following code to retrieve your snapshot:

   ```json
   {
   "type": "fs",
   "settings": {
   "location": "/mount/backups/my_backup",
   "compress": true
   }
   }
   ```

**Note**: You can restore a snapshot on a functioning cluster, but all indexes residing on the cluster must be closed. The restore only updates closed indexes and creates a new index for any index that doesn’t already exist on the cluster.

#### 6.4.2 Restoring snapshots with cURL

Most of the documentation on the Elasticsearch website relies on cURL commands to restore snapshots. You must install cURL for Windows in order to have access to cURL commands in a Windows environment. You can download cURL for Windows . the following site:

**Note:** You can paste cURL commands into Marvel Sense (excluding the $ character), and Marvel automatically converts the cURL command into JSON. The cURL command doesn't convert if typed manually, you must paste it from your clipboard.

Once you install cURL for Windows, you can restore the cluster state and all indexes in a snapshot with the following cURL command:

```
$ curl -XPOST "localhost:9200/_snapshot/my_backup/snapshot_1/_restore"
```

**Note:** You can restore a snapshot on a functioning cluster, but all indexes residing on the cluster must be closed. The restore only updates closed indexes and creates a new index for any index that doesn't already exist on the cluster.

For more information on creating or restoring snapshots of Elasticsearch, the following site:

7 Upgrading Data Grid

The Data Grid Core and Audit applications are updated automatically with Relativity. If you are upgrading to Relativity 9.6 and your environment uses Elasticsearch 2.3.58 or below to store audits, you must upgrade to Elasticsearch 2.3.79 or above when upgrading to Relativity 9.6. Once you upgrade to Elasticsearch 2.3.79, you're unable to use your Elasticsearch clusters with 2.1.2 or have a partially upgraded cluster.

- If you are upgrading to Relativity 9.6 from a version prior to Relativity 9.5.219.30, you must move your long text data out of Elasticsearch and import to SQL. After upgrade, reimport the data to Data Grid. Only then will you be able to access your long text in Data Grid. For more information contact Relativity Support.

7.1 Finding the Elasticsearch version

To find the version of Elasticsearch installed, you must connect to your client node endpoint. The client node is a member of a Relativity Data Grid Elastic cluster. The client nodes stores no data itself; rather, it communicates to and from Relativity agent and web servers and the Data Grid Elastic cluster.

The client node is the value in the AuditDataGridEndPoint and/or DataGridEndPoint instance setting. To connect to the client endpoint, enter the following URL into Chrome or Firefox, substituting the name of your client node. Internet Explorer won't properly display the page.

**Note:** You may be prompted for a username and password. If you are not sure of the username and password, see Resetting the Shield or Head password on page 77.

http://nameofclientnode:9200/

Once you connect to the endpoint, the following page appears which displays, among other things, your Elasticsearch version number.

```json
{
  "name": "emtTest",
  "cluster_name": "Prod01",
  "version": {
    "number": "2.3.3",
    "build_hash": "218bdf10790eef486ff2c41a3df5cfa326adcfde",
    "build_timestamp": "2016-05-17T15:40:04Z",
    "build_snapshot": false,
    "lucene_version": "5.5.0"
  },
  "tagline": "You Know, for Search"
}
```
7.2 Upgrading from Elasticsearch 2.1.2 to 2.3.3.x

In order to upgrade Elasticsearch from 2.1.2 to 2.3.3.x, complete the following workflow:

Click to expand instructions for upgrading Elasticsearch.
In order to upgrade Elasticsearch from 2.1.2 to 2.3.3.x, complete the following workflow:

- Prepare the environment for upgrade. For more information, see Preparing the environment for upgrade below.
- Run the upgrade script on data, master, and client nodes in that order. For more information, see Running the upgrade script on the next page.
- Verify your upgrade completed successfully. For more information, see Verifying the upgrade on the next page.

If you want to manually upgrade Elasticsearch, see Running a manual upgrade on page 73.

7.3 Preparing the environment for upgrade

Before upgrading Elasticsearch, perform the following:

- Ensure that the Elasticsearch service is running under a user account that has access to SQL Server, and specifically has read, write, and bulk permissions for all workspace databases.
- Verify that no reads or writes to Elasticsearch occur during the upgrade process.
- Disable all Audit migration and deletion agents.
- Disable all Text migration and deletion agents.
- Verify that all imports or publishing from Processing have stopped.
- Save a backup of the current lib and bin folders from any node and the data folder from the master node to mitigate the risk in possible restoration. Don’t save the backup files to the installer folder.
- If you are also upgrading Java versions, open the command prompt and run the following command. This example assumes you are upgrading to Java 8 Update 45 (64-bit). Edit the version number appropriately.

```
SETX /M KCURA_JAVA_HOME "C:\Program Files\java\jdk1.8.0_45"
```

- Disable shard allocation:
  - Run the following command in Sense to turn off re-balancing and set the cluster to persistent. The persistent state ensures that re-balancing stays off when the cluster restarts.

```
PUT _cluster/settings
{
  "persistent":{"cluster.routing.allocation.enable": "none"}
}
```
Run the following command to perform a synced flush:

```
POST /_flush/synced
```

### 7.4 Running the upgrade script

**Note:** If you run the script on Powershell versions earlier than 5.1, the script displays an error during back up but will continue with the upgrade.

Extract the contents of the upgrade package, and make sure the following files are in the folder:

- datagrid-2.3.3.58-install.zip
- elasticupgrade.ps1
- upgrade.psd1

You must run the upgrade script on each node. We recommend the following order when upgrading your nodes:

1. Data nodes
2. Master nodes
3. Client nodes

To upgrade a node, complete the following:

1. Open the `upgrade.psd1` file in a text editor. Update the following configurations:
   - **UpgradeFile** - enter the file path to the upgrade package.
   - **CurrentPath** - enter the current location of the installed Elasticsearch service.
   - **Url** - enter the URL of the local machine's Elasticsearch endpoint.
   - **UserName** - (optional) enter the service user name that has access to SQL.
   - **Password** - (optional) enter the password for the server user.

2. Run `elasticupgrade.ps1`.

### 7.5 Verifying the upgrade

After you upgrade all of the nodes on your cluster, complete the following on the cluster to complete the upgrade:

1. Run the following command in Sense:

   ```
   GET /_nodes/jvm?filter_path=**.jvm.gc_collectors
   ```

   Ensure the result shows "ParNew", "ConcurrentMarkSweep".
2. Enable shard allocation to rebalance the cluster:

```
PUT _cluster/settings
{
  "persistent": {"cluster.routing.allocation.enable": "all"}
}
```

You can monitor the indexes by running the following in Sense:

```
GET _cat/recovery?v
```

3. Verify the cluster status by running the following command in Sense.

```
GET _cat/health
```

Once the cluster is GREEN, your node restart is complete.

If the cluster status remains RED for an extended period, run the following in Sense to identify which indexes are RED:

```
GET _cat/recovery?v
```

Note: If you are using Kibana, ensure your version of Kibana is compatible with your version of Elasticsearch.

Note: With Shield on by default, other plugins like Marvel or Head are not supported. In order to use your other plugins, you need to provide the Kibana server with credentials so it can access the .kibana index and monitor the cluster. See the Relativity Data Grid guide for more information.

### 7.6 Running a manual upgrade

Click to expand instructions on how to run a manual Elasticsearch upgrade.

In order to upgrade Elasticsearch from 2.1.2 to 2.3.3.x, complete the following workflow:

- Prepare the environment for upgrade. For more information, see [Preparing the environment for upgrade on page 71](#).
- Upgrade your data, master, and client nodes in that order. For more information, see [Upgrading a node below](#).
- Verify your upgrade completed successfully. For more information, see [Verifying the upgrade on page 76](#).

#### 7.6.1 Upgrading a node

Perform the following steps on each node in the cluster. We recommend the following order when upgrading your nodes:

1. Data nodes
2. Master nodes
3. Client nodes
To upgrade a node, complete the following:

1. Shut down the node.
   a. Open a Windows command prompt as an administrator, and then navigate to the bin directory in the RelativityDataGrid folder.
   
   ```
   \C:\RelativityDataGrid\elasticsearch-main\bin
   ```
   b. Stop the Elasticsearch service by running the following command:
   
   ```
   .\kservice.bat stop
   ```
   
   **Note:** If the service doesn't shut down after being stopped, end the process using Process Explorer.

2. Save your current Java settings.
   a. Run the following:
   
   ```
   .\kservice.bat manager
   ```
   b. On the Java tab, take note of the values for the following settings:
   
   - Initial memory pool
   - Maximum memory pool
   - Thread stack size

   ![Elasticsearch (elasticsearch-service-x64) Properties](image)
3. Remove the service:

```bash
\kservice.bat remove
```

4. Delete the old lib, bin, sqlauth, modules, and plugin folders from `\RelativityDataGrid\elasticsearch-main`.

   - Copy the lib, bin, sqlauth, modules, and plugin folders from the Elastic 2.3.3.x extracted zip file to `\RelativityDataGrid\elasticsearch-main`.

5. Configure the elasticsearch.yml file (`\RelativityDataGrid\elasticsearch-main\config\elasticsearch.yml`) with the following:

   a. Add `.security` to the `action.auto_create_index` values. This is required when Shield is enabled and auto create is set to false.

      ```yaml
      # This disables automatic index creation
      action.auto_create_index: false,.security
      ```

   b. Configure the Shield settings as follows:

      **Note:** To disable Shield, remove the number sign (#) in front of `shield.enabled:false`.

      ```yaml
      #shield.enabled: false
      shield.authc.realms:
        custom:
          type: kCuraBearerRealm
          order: 0
          publicJwksUrl: https://<RELATIVITY.IDENTITY_SERVER>/Relativity/Identity/.well-known/jwks
          esusers1:
            type: esusers
            order: 1
      ```

      **Note:** The URL must point to the Relativity installation where Identity Server can be found. This should be the same URL used to log in to Relativity.

6. Install the service:

   ```bash
   .\kservice.bat install
   ```

7. Verify the Java settings:

   ```bash
   .\kservice.bat manager
   ```

   a. On the **Java** tab, make sure the values for the following settings for each particular node match the settings you took note of above:

      - Initial memory pool
      - Maximum memory pool
      - Thread stack size
b. Select the Log On tab. In the Log on as setting, select This account. Enter a valid Relativity service account domain name and password and confirm the password.

8. Restart the service:

```
\kservice.bat start
```

If the service fails to restart, navigate to C:\RelativityDataGrid\elasticsearch-main\logs and troubleshoot any errors in the logs.

9. Run the following command in Sense to monitor the progress of your node. Wait for the node to go to YELLOW before upgrading the next node.

```
GET _cat/health
```

### 7.6.2 Verifying the upgrade

After you upgrade all of the nodes on your cluster, complete the following on the cluster to complete the upgrade:

1. Run the following command in Sense:

```
GET /nodes/jvm?filter_path=**.jvm.gc_collectors
```

Ensure the result shows "ParNew", "ConcurrentMarkSweep".
2. Enable shard allocation to rebalance the cluster:

```
PUT _cluster/settings
{
  "persistent": {"cluster.routing.allocation.enable": "all"}
}
```

You can monitor the indexes by running the following in Sense:

```
GET _cat/recovery?v
```

3. Verify the cluster status by running the following command in Sense.

```
GET _cat/health
```

Once the cluster is GREEN, your node restart is complete.

If the cluster status remains RED for an extended period, run the following in Sense to identify which indexes are RED:

```
GET _cat/recovery?v
```

**Note:** If you are using Kibana, ensure your version of Kibana is compatible with your version of Elasticsearch.

**Note:** With Shield on by default, other plugins like Marvel or Head are not supported. In order to use your other plugins, you need to provide the Kibana server with credentials so it can access the .kibana index and monitor the cluster. See the Relativity Data Grid guide for more information.

### 7.7 Resetting the Shield or Head password

When navigating to Head or Sense on any node, you may be prompted for a username and password. If you are unsure of the password, or if the password is not set, you can use the esusers tool in the shield folder. This tool can list users, change passwords, and create REST users on the node. The tool is node-specific, meaning you create a user and password only on that node.

1. Run PowerShell as an administrator, and then navigate to `C:\RelativityDataGrid\elasticsearch-main\bin\shield`.

2. Enter the following command to list the users on the node.

   ```
   .\esusers list
   ```

3. If no user exists, create a user using the following command.

   ```
   .\esusers passwd esadmin -p password123 -r admin
   ```

**Note:** This example creates a new user, `esadmin`, with the password `password123` and the role of admin. Substitute the user name and password for the user you want to create.
4. To change the password on the node, use the following command:

```bash
./esusers passwd esadmin -p newpasswordhere
```
8 Audit

Audit is an application that you can use to monitor and run reports on audited user activity. With Audit, you can use pivots and visualizations to quickly identify audit activity in your Relativity instance.

Some of the benefits of using Audit include:

- Building multiple widgets and lists to create custom dashboards for reporting.
- Viewing audit records that occur at the instance level through the Relativity UI.
- Easily searching through millions of audits to identify actions ranging from reviewer coding decisions to users logging in and out of Relativity for monitoring activity.
- Quickly exporting audits to meet stakeholder requests for reporting on user activity in Relativity.

For information about common Audit workflows, see Recipes.

You must have Elasticsearch configured to use the Audit application. See Data Grid for more information on setting up the preliminary requirements.

8.1 Supported functionality

The following lists supported and unsupported Audit functionality as compared to the history tab.

<table>
<thead>
<tr>
<th>Supported functionality</th>
<th>Currently unsupported functionality</th>
</tr>
</thead>
<tbody>
<tr>
<td>ARM – Archive &amp; Restore</td>
<td>Custom scripts</td>
</tr>
<tr>
<td>Dashboards</td>
<td>Export of dashboards</td>
</tr>
<tr>
<td>Group By/Pivot On functionality (limited number of fields)</td>
<td></td>
</tr>
<tr>
<td>Item list</td>
<td></td>
</tr>
<tr>
<td>Filtering</td>
<td></td>
</tr>
<tr>
<td>Views</td>
<td></td>
</tr>
<tr>
<td>Widgets</td>
<td></td>
</tr>
<tr>
<td>Reverting/mass revert document edits</td>
<td></td>
</tr>
<tr>
<td>Error reporting</td>
<td></td>
</tr>
<tr>
<td>Exporting data from lists</td>
<td></td>
</tr>
<tr>
<td>Exporting dashboard widgets as image</td>
<td></td>
</tr>
</tbody>
</table>

8.2 Installing and configuring Audit

In order to use Audit, you must complete the following:
1. Install and configure Elasticsearch. For more information, see Installing Elasticsearch.
2. Install the Audit application to your workspace(s). For more information on installing applications, see the Admin Guide.
3. Install the Audit agents. See Audit agents below.

### 8.2.1 Audit agents

The Audit application includes the agents described below:

<table>
<thead>
<tr>
<th>Agent name</th>
<th>Requirement information</th>
<th>Function</th>
<th>Agent type</th>
</tr>
</thead>
<tbody>
<tr>
<td>Data Grid Audit Migrator</td>
<td>Only 1 core on the agent server</td>
<td>A Data Grid Audit Migrator agent migrates audit data from SQL to Elasticsearch for any workspace that has Audit installed. The frequency with which this agent checks for migrations and runs the migrations is controlled by the agent run interval value.</td>
<td>Multiple-installation</td>
</tr>
<tr>
<td>Data Grid Audit Deleter</td>
<td>Only 1 per environment</td>
<td>The Data Grid Audit Deleter agent is an off-hour agent that deletes all audits from SQL that have been successfully migrated to Elasticsearch. Along with unmigrated agents, it will leave other existing audits in SQL for a configurable number of days for billing purposes.</td>
<td>Single-installation</td>
</tr>
<tr>
<td>Data Grid Audit Reporter</td>
<td>Only 1 per environment</td>
<td>The Data Grid Audit Reporter agent reviews the audit queue for errors that occurred during migration from SQL to Elasticsearch. This agent triggers a Relativity error based on the agent's run interval. This agent is set to an hourly run interval by default.</td>
<td>Single-installation</td>
</tr>
<tr>
<td>Data Grid Audit Manager</td>
<td>Only 1 per environment</td>
<td>Populates filters in the Audit application.</td>
<td>Single-installation</td>
</tr>
<tr>
<td>Data Grid Manager</td>
<td>Only 1 per environment</td>
<td>A Data Grid Manager agent is an off-hours agent responsible for Data Grid enabled workspace management, including deleting outdated search results cache tables and monitoring Data Grid index conditions.</td>
<td>Single-installation</td>
</tr>
</tbody>
</table>

**Note:** If you’ve already installed this agent with Data Grid installation, you don’t need to add another.

For more information on installing agents, see the Agents Guide.

### 8.3 Audit tab

You can use the Audit tab to build custom widgets and design dashboards to meet workflow needs by utilizing the same Group By and Pivot On functionality available in Relativity. The default view on the Audit dashboard shows the last 7 days of Audit activity.
**Note:** You can’t add the Security, Edit, Name, System Last Modified On, System Last Modified By, System Created By, or System Created On fields to an Audit list view.

The following functionality is also available:

- Searching on Field, Old Value, and New Value columns in list view.
- Creating and customizing views on the Audit tab.
- Applying filters and conditions to sort through audit records.
- Exporting audit data from list views and widgets.

### 8.3.1 Searching on the Details column

To display the query text of an audit record, click ![in the Details column for that record. You can toggle between a table view and JSON view of audit details.](image)
## Audit Details

<table>
<thead>
<tr>
<th>Name</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>ID</td>
<td>442409</td>
</tr>
<tr>
<td>TimeStamp</td>
<td>2017-11-20T18:55:00.993</td>
</tr>
<tr>
<td>ArtifactName</td>
<td>Big Case 1</td>
</tr>
<tr>
<td>ActionName</td>
<td>Query</td>
</tr>
<tr>
<td>ActionID</td>
<td>29</td>
</tr>
<tr>
<td>ObjectType</td>
<td>Workspace</td>
</tr>
<tr>
<td>ObjectTypeID</td>
<td>8</td>
</tr>
<tr>
<td>ExecutionTime</td>
<td>16</td>
</tr>
<tr>
<td>ArtifactID</td>
<td>1003663</td>
</tr>
<tr>
<td>UserName</td>
<td>Service Account, Relativity</td>
</tr>
<tr>
<td>UserID</td>
<td>777</td>
</tr>
</tbody>
</table>

**SQL Query**

```sql
/* <Comments>
<ArtifactID />
<ArtifactTypeID1002039</ArtifactTypeID>
<UserID777</UserID>
<WorkspaceID1377826</WorkspaceID>
<QueryTypeIDFull</QueryTypeID>
<QueryID22012363-eb7b-40c2-8cf8-666ee4017ae2</QueryID>
<QuerySourceAndHocSearch</QuerySource>
</Comments> */

SET NOCOUNT ON
SELECT
    [ArtifactID] = [DataGridAuditFieldMapping].[ArtifactID],
    [ParentArtifactID] = [ExtendedArtifact].[ParentArtifactID],
    [Name] = [DataGridAuditFieldMapping].[Name],
    [AuditRelatedSN] = [DataGridAuditFieldMapping].AuditRelatedSN
FROM [DataGridAuditFieldMapping]
```
8.3.2 Exporting workspace audits

You can export audit data from list views and widgets. The export respects the data made on the filters in the list view, and the results are filterable.

**Note:** When you export audit data, the Field, Old Value, and New Value are not included in the output.

To export audit data, complete the following:
1. From the mass operations bar, click Export.

2. Select the format you want to export your audits in (either .csv or .xlsx).

3. Click Run.

**Note:** Exporting audits is not available at the instance level.

### 8.4 Instance audits

The Audit tab is available at the instance level upon upgrade. You must have Elasticsearch installed and configured to use the Audit tab. If you don't have Elasticsearch installed, you can hide this tab.

**Note:** You must be a system admin to view audits at the instance level.

The Audit tab functions the same at the instance level. You can build custom widgets and design dashboards by utilizing the same Group By and Pivot On functionality available in workspace audits. However, exporting audits is not available. For a list of admin-level audited actions, see Audited actions.
8.4.1 Filtering on multi-workspace audits

System admins can view workspace audits at the instance level.

You can filter on the **Workspace Name** column to view audits for specific workspaces. By default, the Audit application displays audit information for the first 10 workspaces, sorted by Artifact ID. You can configure this number using the MaxAggregatedWorkspaces instance setting.
8.5 Reverting audited actions

System admins can revert a coding decision made by a user on a document from the Audit list. Reverting the coding decision will edit the choice back to the previous selection.

You can only revert the Document Update, Mass Update, or Propagate action. You can only revert on the following field types:

- Currency
- Date
- Decimal
- Fixed length
- Single choice
- Single object
- User
- Whole number
- Yes/No

To revert a coding decision:
1. (optional) Filter on the Action column for Update.
2. Click the Revert button next to a coding decision. A modal appears asking if you’re sure you want to revert the audit.

![Image of Revert Action on Audit Grid]

3. Click Revert again.

After you revert the audit, a new audit record is created for the revert you made.

### 8.5.1 Mass reverting

System admins can also mass revert audits. Mass reverting is only available at the workspace level.

You can only mass revert the Document Update, Mass Update, or Propagate action, and it only applies to the latest action on the Document object. You can only mass revert on the following field types:

- Currency
- Decimal
- Single choice
- Single object
- Whole number

To mass revert:

1. From the mass operations bar, select Revert in the drop-down menu.

**Note:** By default, the maximum number of audits that can be present on the item list view for run the mass revert operation is 5000. You can configure this number using the RevertMaxAuditCount instance setting.
2. Audit scans the list of audits to determine how many are revertible. Once it finishes, click Run.

![Reverting Audits](image)

3. Click Close to close the pop-up window.

As when you revert a single audit, a new audit record is created for each reverted audit.

### 8.6 Audit Migration Reports tab

The Audit Migration Reports tab provides you with the option of generating a migration error and migration status report for you to monitor the status of audit data as you migrate it into Elasticsearch. You also have the option of running a migration error retry script.

#### 8.6.1 Migration Error Report

The Migration Error Report generates a report that provides information on all errors that have occurred while migrating audit data into Elasticsearch, including batch and document errors.

**Note:** You can also run this script on all environments in your workspace from the Relativity Script Library tab. Click the user drop-down menu in the upper-right corner of Relativity, and then click Home. Navigate to the Applications & Scripts > Relativity Script Library tab, and then search for the name of the script.

To preview the script, select the radio button for this script and click **Preview**. To run this report, click **Run**. The following results populate the bottom of the window.

**Note:** Audit automatically retries certain audit migration errors, like SQL timeouts, invalid XML, and inability to connect to Elasticsearch. With this improvement, the migration error report may display a total of 0 errors because the system automatically retried and resolved those common errors. In this situation, there is no need to run the Migration Error Retry Script.
Relativity Data Grid Guide

Audit Admin

Migration Error Report
Migration Error Retry Script
Migration Status Report

This Relativity script reports on all errors that have occurred while migrating audit data into the Data Grid repository.

Documentation workspace
Audit Migration Error Report

Total Error Counts

<table>
<thead>
<tr>
<th>Batch Errors</th>
<th>Document Errors</th>
</tr>
</thead>
<tbody>
<tr>
<td>4</td>
<td>0</td>
</tr>
</tbody>
</table>

Batch Errors

<table>
<thead>
<tr>
<th>Batch ID</th>
<th>Audit ID Range</th>
<th>Time Stamp</th>
<th>Error Message</th>
</tr>
</thead>
</table>
■ **Total Error Counts** - this section provides a summary of all errors that occurred during migration.
  - **Batch Errors** - lists the total number of batch errors that occurred during migration.
  - **Document Errors** - lists the total number of document errors that occurred during migration.

■ **Batch Errors** - this section provides a summary of all batch errors that occurred during migration.
  - **Batch ID** - lists the batch ID number of the batch that received an error.
  - **Audit ID Range** - lists the first and last Audit ID numbers contained in the batch that received an error.
  - **Time Stamp** - lists the time at which the error occurred.
  - **Error Message** - lists the text of the batch error.

■ **Document Errors** - this section provides a summary of all document errors that occurred during migration.
  - **Audit ID** - lists the audit ID number of the document that received an error.
  - **Batch ID** - lists the batch ID number of the batch in which the document that received an error was contained.
  - **Time Stamp** - lists the time at which the error occurred.
  - **Error Message** - lists the text of the document error.

**8.6.2 Migration Error Retry Script**

After identifying errors with the Migration Error Report, an admin can investigate and fix specific audit issues. Once the issues have been fixed, this script will resubmit audit records with errors to be migrated into Elasticsearch.

**Note:** You can also run this script on all environments in your workspace from the Relativity Script Library tab. Click the user drop-down menu in the upper-right corner of Relativity, and then click Home. Navigate to the Applications & Scripts > Relativity Script Library tab, and then search for the name of the script.
In the **Error type to retry** drop-down, select from the following options:

- **Batch Errors** - retries all batches with errors as reported in the Migration Error Report.
- **Document Errors** - retries all documents with errors as reported in the Migration Error Report
- **All Errors** - retries all batch errors and document errors as reported in the Migration Error Report.

Click **Preview** to see a sample of the report, or select the appropriate option and click **Run** to execute the script.

### 8.6.3 Migration Status Report

The Migration Status Report generates a report that shows the progress of the migration of audit data into Elasticsearch. This includes audits in SQL, audits pending migrations, and migration errors. To preview the script, select the radio button for this script and click **Preview**.
Click Run to generate this report and display the following information:

- **Audits in SQL** - lists the number of audits in the SQL database.
- **Audits Pending Migration** - lists the number of audits in queue for migration into Data Grid.
- **Migration Errors** - lists the number of errors that occurred during migration.

### 8.7 Audit Workspace Settings tab

Use the Audit Workspace Setting tab to configure settings specific to the Audit application. The Audit Workspace Setting tab contains two settings, DisplayAuditDataSource and HistoryTabVisibleOnUpgrade.

**Note:** You must have permission to view this tab in order to configure the setting.

#### 8.7.1 DisplayAuditDataSource

The DisplayAuditDataSource setting configures whether Relativity displays audits from SQL or Data Grid on the document history pane in the viewer and the View Audit button on object layout pages.

If you set audits to display from Data Grid, there may be a slight delay in viewing the most recent audits. A "Last Updated" value next to audits indicates the last time all audits were verified as migrated for that object. This let's you know that several of a document's most recent audits might not be migrated yet.
You can configure the DisplayAuditDataSource setting with one of three values:

- **0** - displays audits from SQL until the first point that the audit migration has no more audits to migrate. At this point, the system automatically updates this value to "1," and audits are displayed from Data Grid going forward.
- **1** - displays audits from Data Grid only.
- **2** - displays audits from SQL only.

By default, DisplayAuditDataSource is set to display audits from SQL only. To configure DisplayAuditDataSource, click the **Edit** link and modify the **Value** field in the subsequent layout.

### 8.7.2 HistoryTabVisibleOnUpgrade

The HistoryTabVisibleOnUpgrade tab configures whether the History tab is visible when upgrading the Audit application.

You can configure the HistoryTabVisibleOnUpgrade setting with one of two values:

- **True** - displays the History tab upon upgrade.
- **False** - hides the History tab upon upgrade.
By default, the HistoryTabVisibleOnUpgrade setting is set to hide the History tab when upgrading the Audit application. To configure HistoryTabVisibleOnUpgrade, click the Edit link and modify the Value field in the subsequent layout.

## 8.8 Audited actions

The following table lists audited actions in Relativity:

<table>
<thead>
<tr>
<th>Action name</th>
<th>Description of activity</th>
</tr>
</thead>
<tbody>
<tr>
<td>CaseMap - Add Document</td>
<td>A document was sent to CaseMap.</td>
</tr>
<tr>
<td>CaseMap - Add Fact</td>
<td>A selection of text from the viewer was sent to CaseMap as a fact.</td>
</tr>
<tr>
<td>Conversion Complete</td>
<td>A file was converted by way of a user clicking on a file link in the document list, running an imaging set, imaging on the fly, running a mass image operation, or switching to text or production mode in the viewer.</td>
</tr>
<tr>
<td>Create</td>
<td>An item was created.</td>
</tr>
<tr>
<td>Delete</td>
<td>An item was deleted.</td>
</tr>
<tr>
<td>Document Query</td>
<td>A query was run on a list of documents, or a document query was canceled. (A message indicating that a query was canceled is displayed in the details and on the Query Text pop-up.)</td>
</tr>
<tr>
<td>Export</td>
<td>The contents of a production set, saved search, folder, or subfolder were exported.</td>
</tr>
<tr>
<td>Images - Created</td>
<td>Images were created.</td>
</tr>
<tr>
<td>File Download</td>
<td>A file was downloaded through the Single File Upload application. For example, a user clicked Download File on the Error Actions console of an individual processing error layout.</td>
</tr>
<tr>
<td>File Upload</td>
<td>A file was uploaded through the Single File Upload application. For example, a user clicked Upload Replacement File on the Error Actions console of an individual processing error layout.</td>
</tr>
<tr>
<td>Images - Created for Production</td>
<td>Images corresponding to a production outside of Relativity were imported into the system.</td>
</tr>
<tr>
<td>Images - Deleted</td>
<td>Images were deleted.</td>
</tr>
<tr>
<td>Import</td>
<td>Content associated with a load, production, or image file was imported.</td>
</tr>
<tr>
<td>Markup - Image - Created</td>
<td>Redactions or highlights were added to an image.</td>
</tr>
<tr>
<td>Markup - Image - Deleted</td>
<td>Redactions or highlights were removed from an image.</td>
</tr>
<tr>
<td>Markup - Image - Modified</td>
<td>Redactions or highlights were moved, resized or edited on an image.</td>
</tr>
<tr>
<td>Action name</td>
<td>Description of activity</td>
</tr>
<tr>
<td>-----------------------------</td>
<td>--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Markup - Native - Created</td>
<td>Redactions or highlights were added. This audit entry applies to transcripts only.</td>
</tr>
<tr>
<td>Markup - Native - Deleted</td>
<td>Redactions or highlights were removed. This audit entry applies to transcripts only.</td>
</tr>
<tr>
<td>Markup - Native - Updated</td>
<td>Redactions or highlights were moved, resized or edited. This audit entry applies to transcripts only.</td>
</tr>
<tr>
<td>Move</td>
<td>A document was moved from one folder to another.</td>
</tr>
<tr>
<td>Native - Created</td>
<td>A native file was loaded into Relativity.</td>
</tr>
<tr>
<td>Native - Deleted</td>
<td>A native file was removed from Relativity.</td>
</tr>
<tr>
<td>Pivot Query</td>
<td>A Pivot report was run, or a Pivot report was canceled. (A message indicating that a query was canceled is displayed in the details and on the Query Text pop-up.)</td>
</tr>
<tr>
<td>Print</td>
<td>A document was printed. (ActiveX viewer only)</td>
</tr>
<tr>
<td>Production - Add Document</td>
<td>A document was added to a production.</td>
</tr>
<tr>
<td>Production - Remove Document</td>
<td>A document was removed from a production.</td>
</tr>
<tr>
<td>Query</td>
<td>A process ran a query (such as categorization), or a query was canceled. (A message indicating that a query was canceled is displayed in the details and on the Query Text pop-up.)</td>
</tr>
<tr>
<td>RelativityScriptExecution</td>
<td>A Relativity script was run.</td>
</tr>
<tr>
<td>ReportQuery</td>
<td>A summary report was run.</td>
</tr>
<tr>
<td>Run</td>
<td>A file was imaged, saved as a PDF, or otherwise converted for viewing.</td>
</tr>
<tr>
<td>Search Cache Table Creation</td>
<td>A search cache table was created. (Search cache tables are created the first time you search for a term or phrase using dtSearch or Relativity Analytics.)</td>
</tr>
<tr>
<td>Security</td>
<td>Security rights were assigned or changed</td>
</tr>
<tr>
<td>Tally/Sum/Average</td>
<td>The mass operation Tally/Sum/Average was run in the workspace.</td>
</tr>
<tr>
<td>Update</td>
<td>Document metadata was updated on a single-document basis. In addition, filters on information related to applications installed through the workspace or by an agent.</td>
</tr>
<tr>
<td>Update - Mass Edit</td>
<td>Document metadata was updated on a mass basis.</td>
</tr>
<tr>
<td>Update - Mass Replace</td>
<td>Document metadata was edited using a text mass replacement.</td>
</tr>
<tr>
<td>Update - Propagation</td>
<td>Document metadata was edited according to a propagation rule.</td>
</tr>
<tr>
<td>View</td>
<td>A document was viewed.</td>
</tr>
<tr>
<td>Workspace Upgrade</td>
<td>Details about scripts run on a workspace during an upgrade.</td>
</tr>
</tbody>
</table>
The following actions are audited at the admin-level

<table>
<thead>
<tr>
<th>Action name</th>
<th>Description of activity</th>
</tr>
</thead>
<tbody>
<tr>
<td>Create</td>
<td>An item was created.</td>
</tr>
<tr>
<td>Update</td>
<td>An item was updated</td>
</tr>
<tr>
<td>Delete</td>
<td>An item was deleted.</td>
</tr>
<tr>
<td>Security</td>
<td>An owner of a saved search was set during creation or edit.</td>
</tr>
<tr>
<td>Login</td>
<td>A user logged in to Relativity.</td>
</tr>
<tr>
<td>Logout</td>
<td>A user logged out of Relativity.</td>
</tr>
<tr>
<td>Export</td>
<td>Data was exported from a tab at the instance level. For example, a workspace list, list of users, list of groups, etc.</td>
</tr>
<tr>
<td>RelativityScriptExecution</td>
<td>A Relativity script was executed.</td>
</tr>
<tr>
<td>Password Reset Request - Submitted</td>
<td>A password reset request was submitted.</td>
</tr>
<tr>
<td>Password Reset Request - Opened</td>
<td>A password reset request was opened.</td>
</tr>
<tr>
<td>Password Reset Request - Successful</td>
<td>A password reset request was successful.</td>
</tr>
<tr>
<td>Security Preview Started</td>
<td>A security preview started on a user or group.</td>
</tr>
<tr>
<td>Security Preview Ended</td>
<td>A security preview ended on a user or group.</td>
</tr>
<tr>
<td>Workspace Upgrade</td>
<td>Upgrade script run on the workspace (details can be viewed in the JSON tab of the Details).</td>
</tr>
<tr>
<td>Login - Failed</td>
<td>A login attempt to Relativity failed.</td>
</tr>
</tbody>
</table>
Proprietary Rights

This documentation ("Documentation") and the software to which it relates ("Software") belongs to Relativity ODA LLC and/or Relativity’s third party software vendors. Relativity grants written license agreements which contain restrictions. All parties accessing the Documentation or Software must: respect proprietary rights of Relativity and third parties; comply with your organization’s license agreement, including but not limited to license restrictions on use, copying, modifications, reverse engineering, and derivative products; and refrain from any misuse or misappropriation of this Documentation or Software in whole or in part. The Software and Documentation is protected by the Copyright Act of 1976, as amended, and the Software code is protected by the Illinois Trade Secrets Act. Violations can involve substantial civil liabilities, exemplary damages, and criminal penalties, including fines and possible imprisonment.

©2020. Relativity ODA LLC. All rights reserved. Relativity® is a registered trademark of Relativity ODA LLC.