# Table of Contents

1 **Relativity Data Grid** ........................................................................................................... 6
   1.1 Data Grid compatibility matrix .................................................................................. 6
   1.2 Data Grid components .............................................................................................. 6
   1.3 Working with Data Grid .......................................................................................... 8
   1.4 Support, unsupported, and restricted functionality .................................................. 8

2 **Data Grid system requirements** ..................................................................................... 10
   2.1 Infrastructure considerations ................................................................................... 10
   2.2 Data Grid cluster system requirements ................................................................... 10
   2.3 Data Grid compatibility matrix ................................................................................. 11
   2.4 Supported JVM ........................................................................................................ 11

3 **Installing and configuring Data Grid** .......................................................................... 12
   3.1 Creating master, client, and data nodes .................................................................... 12
      3.1.1 Adding additional masters ............................................................................. 18
      3.1.2 Java Virtual Machine (JVM) settings ............................................................... 19
   3.2 Creating a cluster ...................................................................................................... 19
   3.3 Installing a monitoring cluster .................................................................................. 20
   3.4 Installing the license .................................................................................................. 22
      3.4.1 Setting up the Kibana server ......................................................................... 23
      3.4.2 Installing Kibana ............................................................................................ 24
      3.4.3 Installing Marvel and Sense ............................................................................ 25
      3.4.4 Configuring Shield ......................................................................................... 25
      3.4.5 Updating your Data Grid cluster ................................................................... 25
   3.5 Linking Relativity to the Data Grid cluster ................................................................. 26
   3.6 Configuring Shield authentication ............................................................................ 30
      3.6.1 Installing the Shield plugins ........................................................................... 31
      3.6.2 Configuring custom authentication ................................................................ 31
      3.6.3 Installing an X.509 certificate ........................................................................ 34
      3.6.4 Create a keystore and generate a keypair ....................................................... 34
      3.6.5 Create a certificate signing request (CSR) ..................................................... 35
3.6.6 Send the certificate to your certificate authority (CA) for signing ........................................35
3.6.7 Install the signed certificate .................................................................35
3.7 Enabling SSL in the node configuration ..................................................36
3.8 Configuring Kibana to work with Shield ................................................36
3.9 Configuring Data Grid ...........................................................................37
  3.9.1 Installing the Data Grid Core and Data Grid for Audit applications ..........37
  3.9.2 Enabling your workspace and extracted text field for Data Grid ................38
  3.9.3 Enabling the Data Grid search index ..................................................39
4 Upgrading Data Grid ...................................................................................41
  4.1 Upgrading a node ..................................................................................41
5 Configuring Shield authentication ..............................................................46
  5.1 Installing the Shield plugins .................................................................46
  5.2 Configuring custom authentication .......................................................46
    5.2.1 Defining roles ................................................................................47
    5.2.2 Elastic Shield realm ......................................................................47
    5.2.3 LDAP realm ................................................................................48
    5.2.4 Active Directory (AD) realm ..........................................................49
  5.3 Installing an X.509 certificate ...............................................................49
    5.3.1 Create a keystore and generate a keypair ........................................49
    5.3.2 Create a certificate signing request (CSR) ......................................50
    5.3.3 Send the certificate to your certificate authority (CA) for signing .........50
    5.3.4 Install the signed certificate ..........................................................51
  5.4 Enabling SSL in the node configuration ...............................................51
  5.5 Configuring Kibana to work with Shield ..............................................51
6 Data Grid agents .........................................................................................53
  6.1 Data Grid Core agents ..........................................................................54
  6.2 Data Grid for Audit agents ....................................................................56
  6.3 Adding a Data Grid agent .....................................................................57
7 Data Grid instance settings ..........................................................................61
  7.1 Accessing Data Grid instance settings ..................................................61
  7.2 Data Grid instance settings descriptions ..............................................62
<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>7.3 Linking Relativity to the Data Grid service</td>
<td>68</td>
</tr>
<tr>
<td>7.4 Updating the Audit template</td>
<td>72</td>
</tr>
<tr>
<td>8 Integrating Relativity Analytics with Relativity Data Grid</td>
<td>74</td>
</tr>
<tr>
<td>8.1 Installing the Data Grid service</td>
<td>74</td>
</tr>
<tr>
<td>8.2 Uninstalling the Data Grid service</td>
<td>75</td>
</tr>
<tr>
<td>9 Cluster monitoring tools</td>
<td>76</td>
</tr>
<tr>
<td>9.1 Head</td>
<td>76</td>
</tr>
<tr>
<td>9.1.1 Performing offline installations of Head</td>
<td>76</td>
</tr>
<tr>
<td>9.2 Marvel</td>
<td>77</td>
</tr>
<tr>
<td>9.2.1 Performing offline installations of Marvel</td>
<td>78</td>
</tr>
<tr>
<td>9.3 Curator</td>
<td>78</td>
</tr>
<tr>
<td>9.3.1 Performing offline installations of Curator</td>
<td>78</td>
</tr>
<tr>
<td>10 Cluster metrics to monitor</td>
<td>80</td>
</tr>
<tr>
<td>10.1 Monitoring bulk rejections</td>
<td>80</td>
</tr>
<tr>
<td>11 Querying with Marvel Sense</td>
<td>81</td>
</tr>
<tr>
<td>11.1 Retrieving documents and cluster settings</td>
<td>81</td>
</tr>
<tr>
<td>11.2 Shutting down a cluster</td>
<td>83</td>
</tr>
<tr>
<td>11.3 Backing up clusters</td>
<td>84</td>
</tr>
<tr>
<td>12 Backing up Relativity Data Grid</td>
<td>86</td>
</tr>
<tr>
<td>12.1 Creating a repository</td>
<td>86</td>
</tr>
<tr>
<td>12.2 Creating snapshots</td>
<td>88</td>
</tr>
<tr>
<td>12.2.1 Creating snapshots manually from within Sense</td>
<td>88</td>
</tr>
<tr>
<td>12.2.2 Scheduling a Windows task using Curator</td>
<td>89</td>
</tr>
<tr>
<td>12.3 Restoring a snapshot</td>
<td>94</td>
</tr>
<tr>
<td>12.3.1 Restoring snapshots from the Data Grid head console</td>
<td>94</td>
</tr>
<tr>
<td>12.3.2 Restoring snapshots with cURL</td>
<td>94</td>
</tr>
<tr>
<td>13 Data Grid search</td>
<td>96</td>
</tr>
<tr>
<td>13.1 Using Data Grid search</td>
<td>96</td>
</tr>
<tr>
<td>13.2 Using Data Grid search in the new UI framework</td>
<td>97</td>
</tr>
<tr>
<td>13.3 Data Grid check syntax</td>
<td>99</td>
</tr>
<tr>
<td>13.3.1 Using check syntax for Data Grid search in the new UI framework</td>
<td>100</td>
</tr>
</tbody>
</table>
13.4 Data Grid search syntax considerations ................................................. 101
  13.4.1 Phrases .................................................................................. 102
  13.4.2 Question mark single wildcard .................................................. 102
  13.4.3 Asterisk wildcard ................................................................... 103
  13.4.4 Fuzziness .............................................................................. 103
  13.4.5 Proximity .............................................................................. 105
  13.4.6 AND operator ....................................................................... 105
  13.4.7 OR operator .......................................................................... 106
  13.4.8 NOT operator ......................................................................... 106
  13.4.9 Regular expressions ................................................................ 107
  13.4.10 Grouped queries ................................................................. 108
  13.4.11 Special characters .............................................................. 109
  13.4.12 Unsupported dtSearch syntax in Data Grid search .................. 109

14 Data Grid infrastructure optimization ..................................................... 110
  14.1 Virtual vs. physical servers .......................................................... 110
  14.2 Storage types ........................................................................... 110
  14.3 Network connectivity ................................................................. 110
  14.4 Shard settings .......................................................................... 111

15 Data Grid for Audit .............................................................................. 112
  15.1 Installing the Data Grid for Audit application ................................. 112
  15.2 Home tab .................................................................................. 114
    15.2.1 Details filter ...................................................................... 117
    15.2.2 ID filter ............................................................................ 118
    15.2.3 Timestamp filter ............................................................... 119
    15.2.4 Name filter ...................................................................... 120
    15.2.5 Action filter ..................................................................... 120
    15.2.6 Object Type filter ............................................................. 122
    15.2.7 Execution Time filter ......................................................... 124
    15.2.8 Artifact ID filter ............................................................... 124
    15.2.9 User Name filter ............................................................... 125
    15.2.10 Advanced filter .............................................................. 126
15.2.11 Saved Search filter ................................................................. 126
15.3 Saving and loading filter sets ......................................................... 127
15.4 Adding widgets ............................................................................. 129
  15.4.1 User Actions widget ................................................................. 130
  15.4.2 User Names widget ................................................................. 133
  15.4.3 Object Types widget ............................................................... 135
  15.4.4 Actions over Time widget ....................................................... 137
  15.4.5 Results widget ..................................................................... 139
15.5 Creating dashboards .................................................................. 143
15.6 Admin tab .................................................................................. 144
  15.6.1 Migration Error Report ........................................................... 144
  15.6.2 Migration Error Retry Script ............................................... 147
  15.6.3 Migration Status Report ......................................................... 148

16 Restarting nodes and clusters ....................................................... 150
  16.1 Preparing the cluster for node restart ......................................... 150
  16.2 Shutting down a node ............................................................... 150
  16.3 Restarting a node .................................................................... 151
  16.4 Restarting a cluster .................................................................. 151

17 Setting up a monitoring cluster ..................................................... 153
  17.1 Configuring a monitoring cluster in Data Grid 2.x ....................... 153
    17.1.1 Setting up the Kibana server ............................................... 155
    17.1.2 Updating your Data Grid cluster .......................................... 156
1 Relativity Data Grid

Relativity Data Grid is a NoSQL data store that allows you to horizontally distribute full text and audit data across any number of servers.

The benefits of using the Data Grid data store include a reduction in SQL Server database sizes; easier database maintenance, backup, and upgrades; reduced memory requirements; automatic workspace distribution across available servers, increased visibility into Relativity audit data; and an increase in the natural limit of case sizes.

**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.

The Data Grid data store relies on Elasticsearch as its underlying architecture. This page provides a brief description of terminology, functionality, and important considerations to take into account before implementing the Data Grid data store for new workspaces.

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

1.1 Data Grid compatibility matrix

The following table lists the version of Data Grid compatible with each version of Relativity.

<table>
<thead>
<tr>
<th></th>
<th>Relativity 9.4</th>
<th>Relativity 9.3</th>
</tr>
</thead>
<tbody>
<tr>
<td>Data Grid 2.3.3</td>
<td>X</td>
<td>X</td>
</tr>
</tbody>
</table>

1.2 Data Grid components

The architecture on which Data Grid relies is made up of the following components:

- **Node** - a running instance of Data Grid on a Java Virtual Machine (JVM). There are three node types:
  - **Master node** - the node within a cluster that manages changes across the entire cluster.
  - **Client node** - the node that serves as the gateway through which data enters a cluster. Relativity communicates with Data Grid through the client node. The value you enter for the DataGridEndPoint entry in the Instance Settings tab serves as your client node. This is the endpoint that Relativity refers to when requesting data from Data Grid for a Data Grid-enabled field or search.
  - **Data node** - a node that stores data within a cluster.

- **Cluster** - a group of nodes with a common cluster name that share data. Data Grid automatically reorganizes data across nodes within a cluster as you add or remove nodes.

- **Index** - a store of related data that point to one or more shards.
- Shard - a container of data (multiple documents) on a node. There are two shard types:
  - Primary shard - a container of data on a node to which each document in the index belongs. You can have multiple primary shards. The number of primary shards determines the maximum size of an index.
  - Replica shard - a copy of a primary shard.

The following diagram and accompanying steps depict how the different node types interact with each other within a cluster after Relativity sends data to Data Grid via import, processing, or audit migration. This diagram is meant for reference purposes only.

**Note:** Beginning in Relativity 9.3.297.13, you have the option of creating multiple master nodes.

### Elasticsearch cluster workflow

1. The master node tells the client node into which data node it should place the primary shard.
2. The client node places the primary shard into the data node specified by the master node. Data is written to this primary shard.
3. The master node tells the data node to replicate the primary shard.
4. The data node replicates the primary shard and sends the replica shard(s) to another data node based on a command from the master node.

5. Relativity continuously requests data from the endpoint of the client node.

6. The client node continuously supplies Relativity with data for Data Grid-enabled fields and searches.

1.3 Working with Data Grid

Data Grid is a distributed infrastructure integrated with a search engine and an analytics engine. Data Grid allows you to scale out, instead of scaling up, which reduces cost and minimizes the risk of lost data without sacrificing (and actually enhancing) performance.

Data Grid stores documents on data nodes in containers called shards. Each data node can store one or more shards, and each shard stores a collection of documents. There are two kinds of shards: primary shards and replica shards. A primary shard is a collection of original documents within an Data Grid cluster, and a replica shard is a duplicate of a primary shard. Data Grid automatically duplicates any primary shard into a replica shard and distributes the replica to another data node on the cluster. A primary shard and its corresponding replica shard never share a common data node, so redundancy occurs automatically. This ensures that if a data node on a cluster fails, the primary shards on that node can be restored from the replica shards on other nodes. In such cases, Data Grid automatically rebalances the distribution of data across the cluster. The number of replica shards is configurable, and queries on the cluster return faster results as replicas increase.

For more advanced technical information on Elasticsearch, the underlying architecture on which Data Grid runs, visit Elasticsearch.com.

1.4 Support, unsupported, and restricted functionality

Once you enable a field's access to the Data Grid data store, you can't disable it, so it's important to understand the benefits and limitations of allowing fields to access the Data Grid data store in new workspaces. You must have a valid endpoint URL configured to add, edit, or enable a field's access to the Data Grid data store. The maximum record size for a record stored in the Data Grid data store is 100MB. See Installing and configuring Data Grid on page 12 for more information on setting up a valid endpoint URL.
<table>
<thead>
<tr>
<th>Currently supported functionality</th>
<th>Currently unsupported functionality</th>
<th>Restricted functionality</th>
</tr>
</thead>
<tbody>
<tr>
<td>▪ Storage of long text fields on the document object</td>
<td>▪ Sorting or filtering the item list</td>
<td>Applications:</td>
</tr>
<tr>
<td>▪ Visibility of long text fields from within the viewer</td>
<td>▪ Editing fields stored in the Data Grid data store from within Relativity</td>
<td>▪ OCR Destination field</td>
</tr>
<tr>
<td>▪ Concept search</td>
<td>▪ Keyword searching</td>
<td>▪ RAR Text Excerpt field</td>
</tr>
<tr>
<td>▪ dtSearch</td>
<td>▪ Conditional searching</td>
<td>▪ Transform Set Source field</td>
</tr>
<tr>
<td>▪ Relativity Analytics email threading</td>
<td>▪ Analytics searches with data source selections of default training set, default searchable set, or all documents in workspace don't accurately incorporate fields with enabled access to the Data Grid data store. Create an optimized saved search that contains fields with enabled access to the Data Grid data store when building Analytics indexes.</td>
<td>▪ Transform Set Destination field</td>
</tr>
<tr>
<td>▪ Relativity Compare</td>
<td>▪ The migration of data from previously existing workspaces into the Data Grid data store</td>
<td>General Functionality:</td>
</tr>
<tr>
<td>▪ You can import and export data to and from the Data Grid data store with the Relativity Desktop Client.</td>
<td>▪ Changes to long text fields stored in the Data Grid data store don't include old and new values within the audit record.</td>
<td>▪ Export from Item List</td>
</tr>
</tbody>
</table>

**General Functionality:**
- Accessing the View Audit of objects in Relativity
- OCR into a Data Grid-enabled long text field
- The Set extracted text size field script in the Relativity Script Library is not compatible with Data Grid and won't display results for workspaces that use Data Grid.
2 Data Grid system requirements

Depending on your infrastructure tier, you have different server specifications and recommendations for the Data Grid configuration available to you. Data Grid is built on a distributed architecture made up of many servers or nodes. A node is a running instance of Data Grid (a single instance of Data Grid running in the JVM). Every node in a Data Grid 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 Data Grid 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 and is available in case the original cluster becomes unavailable.

2.1 Infrastructure considerations

Consider the following factors when determining the infrastructure requirements for creating a Data Grid 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** – We strongly recommend physical servers but do understand that they might not be a viable option for everyone.

- **Storage type** – Data Grid is a distributed system and you should run it on storage local to each server. For tier 1 systems, regular disks are acceptable, but as you move into tier 2 and 3 consider SSDs for their added performance benefits.

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

2.2 Data Grid 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 Data Grid. When building out your Data Grid infrastructure, use the following formula to calculate the total required number of data nodes:

\[
\text{Size of Data in TBs} \times 0.8
\]

This formula is a shortened version of the following formula

\[
\left(\text{Size of Data in TBs} \times 3.2 \times 1.2\right)/4.5
\]

Where:
3 refers to the total copies of the data
1.2 accounts for indexed data
4.5 refers to the maximum data allowed per node

Note the following considerations:

<table>
<thead>
<tr>
<th>Maximum shards per node</th>
<th>150</th>
</tr>
</thead>
<tbody>
<tr>
<td>Recommended GBs per shard</td>
<td>30</td>
</tr>
<tr>
<td>Total copies of the data</td>
<td>3</td>
</tr>
</tbody>
</table>

Contact support@relativity.com so that our infrastructure team can review the amount of activity and monitoring data you want to store in Data Grid and provide a personalized recommendation of monitoring nodes required.

### 2.3 Data Grid compatibility matrix

The following table lists the version of Data Grid compatible with each version of Relativity.

<table>
<thead>
<tr>
<th></th>
<th>Relativity 9.3</th>
<th>Relativity 9.2</th>
</tr>
</thead>
<tbody>
<tr>
<td>Data Grid 2.1.2</td>
<td>X</td>
<td>X*</td>
</tr>
<tr>
<td>Data Grid 1.7.x</td>
<td>X</td>
<td>X</td>
</tr>
</tbody>
</table>

* The Elastic Shield plugin is not supported for Relativity 9.2. To configure Data Grid 2.1.2 with Relativity 9.2, you must disable Shield in the elasticsearch.yml file.

### 2.4 Supported JVM

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

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

1. Complete the pre-installation steps. See the Pre-Installation Guide.
2. Install the software on which Data Grid runs. This includes:
   a. Master, client and data nodes. See [Creating master, client, and data nodes below](#).
   b. A cluster. See [Creating a cluster on page 19](#).
   c. A monitoring cluster. See [Installing a monitoring cluster on page 20](#).
3. Install and configure the Relativity components. These include the following:
   a. The license and plugins. See [Installing the license on page 22](#).
   b. The Instance Settings that accompany the Data Grid Core application. See [Linking Relativity to the Data Grid cluster on page 26](#).
   c. Data Grid plugins. See
   d. Shield authentication. See [Configuring Shield authentication on page 30](#).

Once all installation steps are completed, you must configure the Relativity core application. The procedure of configuring the Relativity core application is the following:

#### 3.1 Creating master, client, and data nodes

Three different types of nodes are required to run Data Grid 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 the Data Grid data store 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 Data Grid cluster. The data nodes store primary shards and replica shards.

The account running the Data Grid 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 [Data Grid system requirements on page 10](#).

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 Runtime Environment. 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 Data Grid 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 RelativityDataGridelasticsearch-main. This allows you to make upgrades to Data Grid without having to modify the folder to accommodate future version numbers.
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`

- **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.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, `C:\RelativityDataGrid\backups`.

**Note:** You can specify multiple backup locations with the following format: `[/mount/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 30](#) 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**.

![Elasticsearch (elasticsearch-service-x64) Properties](image)

- **Service Name**: elasticsearch-service-x64
- **Display name**: Elasticsearch (elasticsearch-service-x64)
- **Description**: Elasticsearch Windows Service - http://elasticsearch.org
- **Path to executable**: C:\RelativityDataGrid\bin\elasticsearch-service-x64.exe /RS://elasticsearch
- **Startup type**: Automatic

**Service Status**: Started

**Start**, **Stop**, **Pause**, **Restart**

**OK**, **Cancel**, **Apply**

- **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.

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://computename:9200/_plugin/head or http://localhost:9200/_plugin/head

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

3.1.1 Adding additional masters

Data Grid 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 Data Grid 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.

### 3.1.2 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 on page 150](#) for more information.

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

### 3.2 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://compute\name:9200/\_plugin/head](http://compute\name:9200/\_plugin/head) or [http://localhost:9200/\_plugin/head](http://localhost:9200/\_plugin/head) and see if all of your nodes are present.
3.3 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 Data Grid 2.x on a machine that you want to use as a single node monitoring cluster:

1. Install the Java Runtime Environment. 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 Data Grid 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 Data Grid 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 30.

### 3.4 Installing the license

Install the Elasticsearch license from the search/bin folder. If you haven’t already, submit a support ticket to Relativity 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`.

3.4.1 Setting up the Kibana server

Extract the Kibana 4.5.x 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>
3.4.2 Installing Kibana

Kibana is a requirement for Marvel which allows you to manage your Data Grid indexes.

To use Kibana with Data Grid, you need the following:

- Data Grid 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, `C:\RelativityDataGrid\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

3.4.3 Installing Marvel and Sense
Marvel is a management and monitoring tool for Data Grid. 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

3.4.4 Configuring Shield
When you install Data Grid 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.

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

Complete the following on all nodes in the Data Grid 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:\RelativityDataGrid\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
     auth:
   ```
3. Restart the Data Grid service for the changes to take effect.


### 3.5 Linking Relativity to the Data Grid 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 Data Grid service.

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

Use the following steps to link Relativity to your Data Grid 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 note, 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 Data Grid.

   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.

```json
{
   "template": "relativity_*",
   "aliases": {
      "{index}_read": {},
      "{index}_write": {},
      "{index}_verify": {}
   },
   "settings": {
      "index": {
         "number_of_shards": 12,
         "number_of_replicas": 2
      },
      "analysis": {
         "analyzer": {
            "lwhitespace": {
```
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 Data Grid.

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 Relativity and Data Grid.
**Note:** If you already migrated audit data into Data Grid through the Data Grid for Audit application, you must manually delete the template in Data Grid (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",
                            "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": {
  "enabled": true
4. Change the **Enable Data Grid** setting on the workspace details tab from **No** to **Yes**.

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

6. For information on setting up Data Grid for Audit, see **Installing the Data Grid Core and Data Grid for Audit applications on page 37**.

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

### 3.6 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 Data Grid 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 34**.
4. Enable SSL in the node configuration. See **Enabling SSL in the node configuration on page 36**.
5. Restart Data Grid.
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.

3.6.1 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 Data Grid.

3.6.2 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.

3.6.2.1 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:

```yaml
# 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: '*':
    privileges: all

# Only read operations on indices
user:
  indices: '*':
    privileges: read
```
# Only read operations on indices named events_*

```yaml
events_user:
  indices:
    '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.

### 3.6.2.2 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 $

```json
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.

### 3.6.2.3 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:
   ```
   esusers useradd <username> -p <secret>
   ```
3. To add users to roles, enter the following command.
   - `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.

### 3.6.2.4 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"
```

### 3.6.2.5 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
```
3.6.3 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.

3.6.4 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:

   ```
   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:

   ```
   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, L=Chicago, ST=Illinois, C=US correct?
[no]: yes
```

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

### 3.6.5 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
```

### 3.6.6 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.

### 3.6.7 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
```
3.7 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 Data Grid.

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:

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

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

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

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

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

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

4. Restart Data Grid so these configuration changes take effect.

3.8 Configuring Kibana to work with Shield

When you install or update to Relativity 9.3, 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:

   ```yaml
   kibana_elasticsearch_username: kibana4-user
   kibana_elasticsearch_password: kibana4-password
   ```

3. Update the following setting in kibana.yml to false:

   ```yaml
   elasticsearch.ssl.verify: false
   ```

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.
3.9 Configuring Data Grid

The Data Grid Core application contains all the required infrastructure components you need to use Data Grid in your environment. Install the Data Grid Core application a workspace in your environment.

3.9.1 Installing the Data Grid Core and Data Grid for Audit applications

To install the Data Grid Core application, perform the following steps:

1. Navigate to the Relativity Applications tab.
2. Click New Relativity Application in the upper left corner of the All Relativity Applications view.
3. In the Application Type field, select the radio button for Select from Application Library.
4. Click the ellipsis in the Choose from Application Library field.
5. Locate and select the Data Grid Core application in the list of available applications and click Ok.
6. Once Relativity loads the application, click **Import** to install it to your workspace.
7. Repeat steps 1-6 for the **Data Grid for Audit** application.

### 3.9.2 Enabling your workspace and extracted text field for Data Grid

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

1. Navigate to the **Workspace Details** tab.
2. Click **Edit** and set the **Enable Data Grid** field to **Yes**.

![Workspace Details](image)

3. 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**.

![Data Grid Settings](image)

4. Click **Save**.

**Note:** Enabling the extracted text field for Data Grid works for new workspaces only. You can't enable Data Grid for fields that already have text in SQL, as there is no option for migrating data from SQL to Data grid.

### 3.9.3 Enabling the Data Grid search index

To enable the Data Grid search index in your workspace, perform the following steps:

1. Navigate to the Search Indexes tab and open the **Data Grid** index.
2. Edit the value of the **Active** field from **No** to **Yes**.
1. Click **Save**.

You can now select Data Grid from the Search drop-down on your document list.
4 Upgrading Data Grid

There are two strategies for upgrading your Data Grid instance:

- Shut down the entire cluster (all nodes), upgrade, and then restart your nodes. Shut down the master node first to ensure Relativity doesn’t send reads or writes to the cluster, then shut down the client node, then the data nodes. When you bring the nodes back up, start with the data nodes, then the client node, then the master node.

- Take one node down, upgrade it, restart it, and then repeat those steps for each node. Upgrade the data nodes first, then upgrade the client node, then upgrade the master node.

Before upgrading Data Grid, perform the following:

- Verify that no reads or writes to Data Grid occur during the upgrade process.
- Disable all migration agents.
- Verify that all imports or publishing from processing have stopped.
- Save a backup of one set of your old lib and bin folders to mitigate the possibility of restoration. Don’t save the backup files to the installer folder.

Note that you must turn off recovery mode on the cluster to avoid automatic re-balancing of shards during cluster maintenance. The re-balancing process can be resource intensive. Since you’re upgrading with your cluster in a controlled state (maintenance), you can turn off re-balancing. See Restarting nodes and clusters on page 150 for instructions.

The Data Grid service requires access to SQL Server, and specifically needs to have read, write, and bulk permissions for all workspace databases.

4.1 Upgrading a node

Use the following steps to upgrade a node from an old version of Data Grid to the latest version:

1. Prepare your cluster for node restart before uninstalling the kService. See Preparing the cluster for node restart on page 150 for more information.
2. Extract the Elasticsearch upgrade folder to the desktop of each machine on the cluster.
3. Open the elasticsearch-main folder that contains the previous version of Data Grid.
4. Uninstall the previous version of the kService by opening a command prompt to the following path:

   ```
   C:\RelativityDataGrid\elasticsearch-main
   ```
5. Enter the following command:

   ```
   bin\kservice.bat stop
   ```
6. Press Enter.
7. Enter the following command:

   ```
   bin\kservice.bat remove
   ```
8. Press Enter.
9. Delete the old lib and bin folders from C:\RelativityDataGrid\elasticsearch-main.
10. Copy the lib and bin folders from the new version of Data Grid copied onto your desktop in step one, and paste them into C:\RelativityDataGrid\elasticsearch-main.
11. If you are also upgrading Java versions, open the command prompt and run the following command:

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

**Note:** The previous example assumes you are upgrading to Java 8 Update 45 (64-bit). Edit the version number appropriately.

12. Press Enter.
13. Close and reopen the command prompt, and navigate to the following path:

```
C:\RelativityDataGrid\elasticsearch-main
```

14. Enter the following command:

```
bin\kservice.bat install
```

15. Press Enter.
16. Run the GUI manager to reset the Java and Log On values, which were reset when you uninstalled. To do this, perform the following steps:
   a. Enter .\kservice.bat manager and press Enter.
   b. 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.
c. 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 (elasticsearch-service-x64) Properties]

- **Log on as:**
  - Local System account
  - This account: `domain\relativity.service`
  - Password: `********`
  - Confirm Password: `********`

  ![Password Input]

  ![OK, Cancel & Apply Buttons]

  ![Log On Tab]

  ![Login as a User]

  ![Select Local System account]

  ![Select This account]

  ![Enter valid Relativity service account domain name and password]

  ![Confirm the password]

  ![Click OK]

  ![Return to the command prompt]

  ![Enter the following command:]

```
bin\kservice.bat start
```

18. Press Enter.

19. Navigate to the endpoint (localhost:9200) and verify the version number field matches the latest version. It should look similar to the following image:
20. Once all nodes are upgraded, turn recovery mode back on in the cluster. See Restarting nodes and clusters on page 150 for instructions.
5 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 Data Grid 2.1.2, 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. See Installing the Shield plugins.
2. Configure custom authentication. See Configuring custom authentication.
4. Enable SSL in the node configuration. See Enabling SSL in the node configuration.
5. Restart Data Grid.

**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.

5.1 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 Data Grid.

**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. For more information, see Installing Data Grid 2.1.2.

5.2 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.
5.2.1 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:

```yaml
# 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: '*':
    privileges: all

# Only read operations on indices
user:
  indices: '*':
    privileges: read

# Only read operations on indices named events_*
events_user:
  indices: '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.

5.2.2 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:
   ```
   esusers useradd <username> -p <secret>
   ```
3. To add users to roles, enter the following command.
   ```
   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.

### 5.2.3 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"
```

**Settings with User template**

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"
```
5.2.4 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"
```

5.3 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.

5.3.1 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.

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

### 5.3.2 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 -keyalg rsa -keysize 2048 -validity 712 -ext san=dns:*domain.corp,ip:192.168.1.1 -file node01.csr
```

### 5.3.3 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.
5.3.4 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
```

5.4 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 Data Grid.

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.keystore.key_password: myKeyPass
shield.ssl.hostname_verification: false
```

The first line indicates the full path to the node keystore file. This must be a location within the Data Grid 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:

```
trueshield.http.ssl: true
```

4. Restart Data Grid so these configuration changes take effect.

5.5 Configuring Kibana to work with Shield

When you install or update to Relativity 9.3, 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:

```
kibana_elasticsearch_username: kibana4-user
kibana_elasticsearch_password: kibana4-password
```
3. Update the following setting in kibana.yml to false:

```yaml
elasticsearch.ssl.verify: false
```

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.
6 Data Grid agents

A number of agents are available to facilitate Data Grid™ operations in your environment. After you install the Data Grid Core application to your environment and configure the appropriate instance settings, you need to add the agents that accompany that application in order to fully use Data Grid. The following sections provide information on adding these agents.
6.1 Data Grid Core agents

The Data Grid Core application includes the agents shown and described below.

The following table provides a breakdown of what each Data Grid Core agent does:
<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 delete outdated search results cache tables and monitoring Data Grid index conditions.</td>
<td>Single-installation</td>
</tr>
<tr>
<td>Data Grid Kepler Host</td>
<td>Only 1 per agent server</td>
<td>A Data Grid Kepler Host agent is used to host the Kepler service that is responsible for processing requests to Data Grid. Specifically this is used in Processing To Data Grid. There are two instance settings that control the service endpoint and port: ServiceHttpEndpoint and ServiceHttpEndPointPort.</td>
<td>Multiple-installation</td>
</tr>
<tr>
<td>Data Grid Install Queue Manager</td>
<td>Only 1 per environment</td>
<td>A Data Grid Install Queue Manager agent is used to install the Data Grid queue service. By adding this agent on a particular agent server and enabling the agent all required pre-requisites for Data Grid Queue will be installed on that machine. It will also monitor the uptime of the Data Grid Queue once successfully installed. The login credentials and machine where the Data Grid Queue is installed needs to be reflected in instance setting called QueueManagerSettings.</td>
<td>Single-installation</td>
</tr>
<tr>
<td>Data Grid Process Queue Manager</td>
<td>Only 1 per core on the agent server</td>
<td>A Data Grid Process queue manager is responsible for moving records from Data Grid Queue to Data Grid. This agent creates bulk requests from the queue and reports any errors that were generated. On successful writes, the agent then moves to the verification queue, where Data Grid Verification Queue manager processes them.</td>
<td>Multiple-installation</td>
</tr>
<tr>
<td>Data Grid Verify Queue Manager</td>
<td>Only 1 per core on the agent server</td>
<td>A Data Grid Verify queue manager is responsible for verifying that records were successfully written to Data Grid. If any records were unsuccessfully written, they will be moved to the error queue and reported on by Data Grid Status Queue Manager and Data Grid Error Queue Manager agents.</td>
<td>Multiple-installation</td>
</tr>
<tr>
<td>Data Grid Status Queue Manager</td>
<td>Only 1 per core on the agent server</td>
<td>A Data Grid Status Queue manager agent is responsible for handling status requests for a particular set of records that were written to the Data Grid Queue. Clients who write records can subscribe to the status of the documents that were successfully written/verified in order to perform any remaining cleanup tasks.</td>
<td>Multiple-installation</td>
</tr>
<tr>
<td>Data Grid Error Queue Manager</td>
<td>Only 1 per core on the agent server</td>
<td>A Data Grid Error Queue manager agent is responsible for handling error requests for a particular set of records that were written to the Data Grid Queue and failed. Clients who write records can subscribe to receive error messages given by the Queue.</td>
<td>Multiple-installation</td>
</tr>
</tbody>
</table>
6.2 Data Grid for Audit agents

The Data Grid for Audit application includes the agents shown and described below.

The following table provides a breakdown of what each Data Grid for Audit agent does:
<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 per core on the agent server</td>
<td>A Data Grid Audit Migrator agent migrates audit data from SQL to the Data Grid data store for any workspace that has Data Grid for Audit installed. The frequency with which this agent checks for migrations and runs the migrations is controlled with the agent run interval value. Do not run the Data Grid Audit Deleter agent at the same time as the Data Grid Audit Migrator agent, as migration and deletion can conflict.</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 Data Grid. Along with un-migrated agents, it will leave other existing audits in SQL for a configurable number of days for billing purposes. Don't run the Data Grid Audit Deleter agent at the same time as the Data Grid Audit Migrator agent as migration and deletion can conflict.</td>
<td>Single-installation</td>
</tr>
</tbody>
</table>

For general information on Agents and how to deploy them, see the Agents Guide.

### 6.3 Adding a Data Grid agent

To add any of the Data Grid agent types to your environment, perform the following steps:

1. Navigate to the **Agents** tab from Home mode.
2. Click **New Agent**. Complete the fields described below.
Agent Type - click ..., filter for all available Data Grid agents, and select the agent you want to add.

Number of Agents - contains the number of instances of this agent type that you want to create. If you enter a number that would cause the agent to exceed its maximum agents per server value, you receive an error message and the new agent(s) aren't created.

Note: When you create multiple instances of an agent type, each instance is named with a number following in parentheses. For example, the first instance of a Data Grid Audit Migrator agent is named "Data Grid Audit Migrator (1)," the second instance is named "Data Grid Audit Migrator (2)," and so on.

Agent Server - displays the Select Resource Server dialog, allowing you to select the server on which you want the agent to reside and click OK to return to the Agent Information screen.

Note: After you select the agent type, only servers that are compatible with that agent type appear in the Resource Server dialog. If you select the server first and then select an agent type that is not compatible, you receive an error message.

Run Interval - The interval, in seconds, at which the agent checks the database for available jobs. It populates with a default value based on the agent type.

Logging level of event details - specifies the types of events logged for the agent. It populates with a default selection based on the agent type. You can modify this setting by choosing from the following options:

- Log critical errors only - logs messages about critical system failures
- Log warnings and errors - logs messages about critical and non-critical service errors
and disruptions in activity
  - Log all messages - logs detailed messages about all errors and life cycle events

**Note:** When the Log all messages option is selected, the Event Log is rapidly filled to capacity with detailed messages, which causes previous messages to be purged from the log. This option could result in error messages being purged before you have a chance to view the errors.

- **Enabled** - designates the agent instance as disabled or enabled.

3. Click **Save**. Once you save the agent, it's available in the Agents tab.

To edit an agent, click the **Edit** link on the agent name. You can then modify any the fields except for Name and Agent Type. To enable and disable an agent, change the Enabled field from No to Yes. Note that a disabled Data Grid agent causes interruptions in the Data Grid service and also causes processing to become suspended (if you're currently running it).
7 Data Grid instance settings

After you installed the Data Grid™ Core application, you'll need to configure all the entries in the Instance Settings tab that allow Relativity to communicate with the Data Grid service. The following sections provide information on configuring these instance settings and using them to link Relativity to the Data Grid service.

7.1 Accessing Data Grid instance settings

To view these entries, navigate to the Instance Settings tab from the Home screen in Relativity. The default All Instance Settings view lists over 400 entries, in accordance with how many occur in the database across all Relativity features. You can filter on any of the available fields in this view for the entry you're looking for. When searching for Data Grid-specific entries, you can enter "DataGrid OR Audit" in the filter box for the Name field to narrow down the list.

If you're an infrastructure manager or system admin who needs to edit the default value of an Instance Settings entry, click the Edit link next to the entry and modify the Value field in the subsequent layout.
There are four sections of Instance Settings that are relevant to Data Grid:

- **kCura.Audit** - this section corresponds to the Data Grid for Audit application table and is only present in the Instance Settings tab when you install that application.

- **Relativity.DataGrid** - this section is integral to Data Grid and is present in the Instance Settings table by default.

- **Relativity.DataGrid.Queue** - this section is integral to Data Grid and is present in the Instance Settings table by default.

- **Relativity.DataGridMigrator** - this section corresponds to the Data Grid Migrator application table and is only present in the Instance Settings tab when you install that application.

### 7.2 Data Grid instance settings descriptions

The following table provides a listing of all Data Grid-specific Instance Settings entries, along with their sections, default values, and descriptions. Note that there are a number of Instance Settings entries that should never be modified in your environment, despite the fact that they appear in the EDDS.InstanceSetting table in the database and in Relativity. For these entries, we've included a "Do not modify" statement at the beginning of the description.
<table>
<thead>
<tr>
<th>Name</th>
<th>Default value</th>
<th>Description</th>
<th>Section</th>
</tr>
</thead>
<tbody>
<tr>
<td>AuditBatchSize</td>
<td>1000</td>
<td>Determines the batch size that the agent uses when reading from SQL and writing to the Data Grid.</td>
<td>kCura.Audit</td>
</tr>
<tr>
<td>AuditDataGridEndPoint</td>
<td>blank</td>
<td>Used for Data Grid connection. This value is required to activate Data Grid operation for this Relativity instance (<a href="http://client:9200">http://client:9200</a>).</td>
<td>kCura.Audit</td>
</tr>
<tr>
<td>AuditDeleteBatchSize</td>
<td>200000</td>
<td>Determines the size of the batch that the agent uses when deleting audit records previously migrated to Data Grid.</td>
<td>kCura.Audit</td>
</tr>
<tr>
<td>AuditingEnabled</td>
<td>True</td>
<td>Controls whether auditing is enabled throughout Relativity. Setting this value to True enables auditing. Setting this value to False disables auditing.</td>
<td>kCura.Audit</td>
</tr>
<tr>
<td>AuditMigrateBatchSize</td>
<td>50000</td>
<td>Determines the size of the batch that the agent uses when reading from SQL and writing to Data Grid.</td>
<td>kCura.Audit</td>
</tr>
<tr>
<td>AuditPayloadByteThreshold</td>
<td>1048576</td>
<td>This is the maximum payload size for the Data Grid data store.</td>
<td>kCura.Audit</td>
</tr>
<tr>
<td>AuditQueueMaxDocumentErrors</td>
<td>100</td>
<td>Determines the maximum number of errors allowed for a single batch in the Audit Migration Agent before the job is terminated. If this limit is reached, the respective agent will throw an exception saying that the failure limit has been reached.</td>
<td>kCura.Audit</td>
</tr>
<tr>
<td>DataGridEndPoint</td>
<td>blank</td>
<td>Defines the end point for Data Grid connection. This value is required to activate Data Grid operation for this Relativity instance (<a href="http://client:9200">http://client:9200</a>).</td>
<td>Relativity.DataGrid</td>
</tr>
<tr>
<td>DataGridHttpCLIENTTimeout</td>
<td>100</td>
<td>This value is used to set the default value for the Httpclient timeout value for DataGridHttpCLIENT specified in seconds.</td>
<td>kCura.Audit</td>
</tr>
<tr>
<td>DataGridHttpCLIENTTimeout</td>
<td>100</td>
<td>The default value, in seconds, for the Httpclient timeout value for DataGridHttpClient.</td>
<td>Relativity.DataGrid</td>
</tr>
<tr>
<td>Name</td>
<td>Default value</td>
<td>Description</td>
<td>Section</td>
</tr>
<tr>
<td>----------------------------------</td>
<td>-------------------------------------------------------------------------------</td>
<td>---------------------------------------------------------------------------------------------</td>
<td>--------------------------------</td>
</tr>
<tr>
<td>DataGridIndexCreationSettings</td>
<td>{ &quot;template&quot;: &quot;relativity_<em>&quot;, &quot;aliases&quot;: { &quot;{index}_read&quot;: {}, &quot;{index}_write&quot;: {}, &quot;{index}_verify&quot;: {} }, &quot;settings&quot;: { &quot;index&quot;: { &quot;number_of_shards&quot;: 12, &quot;number_of_replicas&quot;: 2 }, &quot;analysis&quot;: { &quot;analyzer&quot;: { &quot;lwhitespace&quot;: { &quot;tokenizer&quot;: &quot;whitespace&quot;, &quot;filter&quot;: [ &quot;lowercase&quot; ] } } } }, &quot;mappings&quot;: { &quot;document&quot;: { &quot;_size&quot;: { &quot;enabled&quot;: true, &quot;store&quot;: true }, &quot;dynamic_templates&quot;: [ { &quot;raw&quot;: { &quot;path_match&quot;: &quot;Field-s-</em>.Value&quot;, &quot;mapping&quot;: { &quot;type&quot;: &quot;string&quot;, &quot;fields&quot;: { &quot;raw&quot;: { &quot;type&quot;: &quot;string&quot;, ... and so on.</td>
<td>Defines a template to create new indexes for Data Grid.</td>
<td>Relativity.DataGrid</td>
</tr>
<tr>
<td>DataGridIndexPrefix</td>
<td>relativity</td>
<td>Constructs index names for Data Grid to distinguish multiple Relativity instances that write to the same Data Grid cluster.</td>
<td>Relativity.DataGrid</td>
</tr>
<tr>
<td>DataGridMemoryByte-SizeThreshold</td>
<td>10485760</td>
<td>Determines the maximum size (in bytes) of the batch that the Data Grid Migrator Agent will attempt to load documents/audits in memory. A value of 0 or less will attempt to load total batch size in memory.</td>
<td>Relativity.DataGrid</td>
</tr>
<tr>
<td>DataGridMultiIndexSplitSizeThreshold</td>
<td>3221225472 (32212254720 bytes)</td>
<td>Determines the maximum size (in bytes) of each shard in the Data Grid index before Relativity creates a new index. Writing to an index that is over the threshold causes Relativity to create a new index. Set this to 0 to disable the threshold behavior.</td>
<td>kCura.Audit</td>
</tr>
<tr>
<td>DataGridMultiIndexSplitSizeThreshold2</td>
<td>32212254720</td>
<td>This instance setting represents the maximum size (in bytes) of each shard in the Data Grid index before Relativity creates a new index. Writing to an index that is over the threshold causes Relativity to create a new index. Set this to 0 to disable the threshold behavior.</td>
<td>Relativity.DataGrid</td>
</tr>
<tr>
<td>Name</td>
<td>Default value</td>
<td>Description</td>
<td>Section</td>
</tr>
<tr>
<td>---------------------------------</td>
<td>---------------</td>
<td>-----------------------------------------------------------------------------</td>
<td>-----------------------------</td>
</tr>
<tr>
<td>DataGridNumberOfRetries</td>
<td>3</td>
<td>Number of attempts made by the Data Grid to perform an operation before it returns as a failure.</td>
<td>kCura.Audit</td>
</tr>
<tr>
<td>DataGridNumberOfRetries</td>
<td>3</td>
<td>Determines the number of attempts that Data Grid performs an operation before it returns a failure.</td>
<td>Relativity.DataGrid</td>
</tr>
<tr>
<td>DataGridPayloadSizeThreshold</td>
<td>1048576</td>
<td>Determines the size, in bytes, of the internal write batch size for Data Grid.</td>
<td>Relativity.DataGrid</td>
</tr>
<tr>
<td>DataGridRESTservicePort</td>
<td>9400</td>
<td>Defines the port number that is used to connect JAVA CA connector to Data Grid REST service.</td>
<td>Relativity.DataGrid</td>
</tr>
<tr>
<td>DataGridRetryWaitTime</td>
<td>50</td>
<td>The amount of time, in milliseconds, that Data Grid waits between operation retries.</td>
<td>kCura.Audit</td>
</tr>
<tr>
<td>DataGridRetryWaitTime</td>
<td>500</td>
<td>The amount of time, in milliseconds, that Data Grid waits between operation retries.</td>
<td>Relativity.DataGrid</td>
</tr>
<tr>
<td>DataGridSearchProviderCacheTimeout</td>
<td>60</td>
<td>The number of seconds that the workspace Last Update Time cache will be used from memory before refreshing with data from the Data Grid. Value 0 means the default value is used.</td>
<td>Relativity.DataGrid</td>
</tr>
<tr>
<td>DataGridVerification</td>
<td>FALSE</td>
<td>This instance setting enables post import data verification.</td>
<td>Relativity.DataGrid</td>
</tr>
<tr>
<td>DataGridVerificationTimeout</td>
<td>30</td>
<td>Determines the time span upon which post import data verification occurs.</td>
<td>Relativity.DataGrid</td>
</tr>
<tr>
<td>Name</td>
<td>Default value</td>
<td>Description</td>
<td>Section</td>
</tr>
<tr>
<td>--------------------</td>
<td>-----------------------------------------------------------------------------</td>
<td>------------------------------------------------------------------------------</td>
<td>---------------</td>
</tr>
<tr>
<td>ESIndexCreationSettings</td>
<td>Click here to view the default Data Grid template</td>
<td>Defines a template to create new indexes for Data Grid.</td>
<td>kCura.Audit</td>
</tr>
<tr>
<td></td>
<td>{</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>&quot;template&quot;: &quot;audit_*&quot;,</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>&quot;aliases&quot;: {</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>&quot;{index}_read&quot;: {},</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>&quot;{index}_write&quot;: {}</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>},</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>&quot;settings&quot;: {</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>&quot;index&quot;: {</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>&quot;number_of_shards&quot;: 12,</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>&quot;number_of_replicas&quot;: 2</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>},</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>&quot;analysis&quot;: {</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>&quot;analyzer&quot;: {</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>&quot;str_search_analyzer&quot;: {</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>&quot;tokenizer&quot;: &quot;keyword&quot;,</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>&quot;filter&quot;: [</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>&quot;lowercase&quot;,</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>&quot;substring&quot;</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>]</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>},</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>&quot;str_index_analyzer&quot;: {</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>&quot;tokenizer&quot;: &quot;keyword&quot;,</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>&quot;filter&quot;: [</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>&quot;lowercase&quot;,</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>&quot;substring&quot;</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>]</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>},</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>&quot;filter&quot;: {</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>&quot;substring&quot;: {</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>&quot;type&quot;: &quot;nGram&quot;,</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>&quot;min_gram&quot;: 1,</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>&quot;max_gram&quot;: 20</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>}</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>},</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>&quot;mappings&quot;: {</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>&quot;audit&quot;: {</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>&quot;dynamic_templates&quot;: [</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>&quot;raw&quot;: {</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>}</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Name</td>
<td>Default value</td>
<td>Description</td>
<td>Section</td>
</tr>
<tr>
<td>-----------------------------------</td>
<td>---------------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
<td>-----------------</td>
</tr>
<tr>
<td>ESIndexPrefix</td>
<td>relativity</td>
<td>Constructs index names for Data Grid. This instance setting can be used to distinguish multiple Relativity instances that write to the same Data Grid cluster.</td>
<td>kCura.Audit</td>
</tr>
<tr>
<td>ESPayloadSettings</td>
<td></td>
<td>Defines Data Grid settings attached to every payload.</td>
<td>kCura.Audit</td>
</tr>
<tr>
<td>HeartbeatCreateIndexRetryWaitTime</td>
<td>5000</td>
<td>Determines the wait time before the final retry attempt at creating new Data Grid index after receiving an audit event.</td>
<td>Relativity.DataGrid</td>
</tr>
<tr>
<td>Name</td>
<td>Default value</td>
<td>Description</td>
<td>Section</td>
</tr>
<tr>
<td>----------------------------------</td>
<td>---------------</td>
<td>------------------------------------------------------------------------------</td>
<td>--------------------------------</td>
</tr>
<tr>
<td>MaxGridItemsToExport</td>
<td>5000</td>
<td>Maximum number of audits that can be exported from Data Grid.</td>
<td>kCura.Audit</td>
</tr>
<tr>
<td>PostMigrationPersistencePeriod</td>
<td>90</td>
<td>The amount of time, in days, that audit data remains in SQL after it has been created.</td>
<td>kCura.Audit</td>
</tr>
<tr>
<td>QueueManagerSettings</td>
<td>&quot;BatchCount&quot;:3, &quot;BatchSize&quot;:1000, &quot;RabbitServer&quot;:&quot;localhost&quot;, &quot;RabbitPassword&quot;:&quot;2fast&quot;, &quot;RabbitUser&quot;:&quot;toofast&quot;, &quot;VerifyQueue&quot;:true, &quot;RabbitServerPort&quot;:&quot;15672&quot;, &quot;QueueTimeOut&quot;:1000</td>
<td>The settings found in the data grid queue.</td>
<td>Relativity.DataGrid.Queue</td>
</tr>
<tr>
<td>RequestTimeoutMilliseconds</td>
<td></td>
<td>The timeout value, in milliseconds, for all requests made to the Data Grid by the agent. A blank value will result a default of 100,000 milliseconds.</td>
<td>kCura.Audit</td>
</tr>
<tr>
<td>ServiceHttpEndpoint</td>
<td></td>
<td>Determines the HTTP end point for access to the Data Grid services. The default value is empty, and you must manually add a URL here. For example, http://&lt;server-name&gt;:_&lt;ServiceHttpEndPointPort&gt;/dgapi. This URL will represent how users can reach the DataGrid Kepler Server.</td>
<td>Relativity.DataGrid</td>
</tr>
<tr>
<td>ServiceHttpEndPointPort</td>
<td>9005</td>
<td>The port used by the Data Grid Kepler Service. Default Port 9005. The ServiceHttpEndPointPort instance setting is available after you install the Data Grid Core application.</td>
<td>Relativity.DataGrid</td>
</tr>
</tbody>
</table>

### 7.3 Linking Relativity to the Data Grid service

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

Use the following steps to link Relativity to your Data Grid service:

1. Specify the URL and port of the client node in the DataGridEndPoint instance setting value (i.e., http://CLIENT_NODE_COMNAME: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 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. The default instance setting value for number_of_shards is set to 12, which assumes six 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 Data Grid.

   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": "number_of_replicas":
    },
    "analysis": {
      "analyzer": {
        "lwhitespace": {
          "tokenizer": "whitespace",
          "filter": [
            "lowercase"
          ]
        }
      }
    }
  },
  "mappings": {
    "document": {
      "_timestamp": {
        "enabled": true
      },
      "_size": {
        "enabled": true,
        "store": true
      }
    }
  }
}
```
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 Data Grid.

**Note:** If you are upgrading to Relativity 9.3.376.35 from a version before Relativity 9.3.332.21, you must follow alternate steps to update the audit template. For more information, see [Updating the Audit template](#).

e. The ESIndexCreationSettings value populates with the default template that includes settings for 12 shards and two replicas. Enter the SQL command below for 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 Relativity and Data Grid.

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

f. Use the following command to edit the ESIndexCreationSettings instance setting value to match your particular cluster setup:
{
  "template": "audit_*",
  "aliases": {
    "{index}_read": {},
    "{index}_write": {},
    "{index}_verify": {}
  },
  "settings": {
    "index": {
      "number_of_shards": 12,
      "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": {
            "path_match": "Details\.auditElement\..*",
            "match_pattern": "regex",
            "match_mapping_type": "string",
            "mapping": {
              "type": "string",
              "fields": {
                "Raw": {
                  "type": "string",
                  "analyzer": "lwhitespace"
                }
              }
            }
          }
        }
      ]
    }
  }
}
4. Change the **Enable Data Grid** setting on the workspace details tab from No to Yes.

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

6. For information on setting up Data Grid for Audit, see [Data Grid for Audit on page 112](#).

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

After you've configured all the instance settings you need to use Data Grid, you'll need to create and add all necessary agents. For more information, see [Data Grid agents on page 53](#).

### 7.4 Updating the Audit template

The new audit template available in 9.3.376.35 provides a way to prevent mapping errors during audit migration. If you're upgrading to Relativity 9.3.376.35 from a previous version of Relativity with the Data Grid for Audit application installed, you must follow the steps below to update your audit template.

Before you install the updated Data Grid for Audit application, perform the following steps:

1. Stop all instances of Data Grid Audit Migrator agents. See [Editing or disabling agents](#).
2. Navigate to the **Instance Settings** tab from the Home screen in Relativity.
3. Search for the **ESIndexCreationSettings** instance setting and click the name of the instance setting. Take note of the **number_of_shards** and **number_of_replicas** setting within the **Value** field.
4. Delete the **ESIndexCreationSettings** instance setting by clicking the **Delete** button.
5. Delete the audit template by typing the following in Sense:
   
   ```
   DELETE _template/audit
   ```

   Once you’ve completed the steps above, install the updated Data Grid for Audit application from the March update package. The update package re-installs the `ESIndexCreationSettings` instance setting with an updated audit template. To finish the installation and update of the audit template, complete the following steps:

   1. Navigate to the **Instance Settings** tab from the Home screen in Relativity.
   2. Search for the `ESIndexCreationSettings` instance setting and click **Edit** next to the name of the instance setting.
   3. In the **Value** field, update the settings for **number_of_shards** and **number_of_replicas** based on the previously reviewed settings.
   4. Restart all instances of Data Grid Audit Migrator agents. See [Restarting disabled agents](#).
   5. Create a new temporary workspace, and install the Data Grid for Audit application in this workspace. This will write new audits to Data Grid, regenerating the audit template from the instance setting. You can then delete this workspace.
8 Integrating Relativity Analytics with Relativity Data Grid

**Note:** Beginning in Relativity 9.3.376.35, you no longer need to install the custom DataGridRESTService on the Analytics server or install an Data Grid client node on the Analytics server to integrate Analytics with Data Grid. If you are upgrading to Relativity 9.3.376.35, uninstall the DataGridRESTService from your Analytics server(s). For more information, see [Uninstalling the Data Grid service](#). We also recommend uninstalling the client node from the Analytics server if that server is only dedicated to Analytics in order to free up resources.

If you use Relativity Analytics, you must take certain steps while setting up the client node in order to get the full benefit of Analytics functionality. This page explains how to install the Data Grid™ service so that Analytics works properly on data stored in the Data Grid repository.

**Note:** When using Data Grid, you must create and use a saved search for your training and searchable sets in order to include the Extracted Text field. Do not select the `<default training set>`, `<default searchable set>`, or `<all documents in workspace>` data source options.

**Note:** The Set extracted text size field script is not compatible with Data Grid and won’t display results for workspaces that use Data Grid.

### 8.1 Installing the Data Grid service

Use the following steps to integrate Relativity Analytics functionality with Data Grid:

**Note:** You must install an Data Grid client node on the same server as Analytics to integrate Analytics functionality for data stored in Data Grid. Do not use the client node on the Analytics server as an endpoint connection for uploading data into the Data Grid data store.

1. Navigate to `[CA HOME]\webapps\nexus\WEB-INF\lib\`.  
2. Unzip `DataGridRESTService.zip` to `C:\DataGridRESTService`.  
3. Launch the Windows command prompt and type the following:
   1. `cd c:\DataGridRESTService` - navigates the command prompt to the correct Relativity Analytics directory.  
   3. `services.msc` - launches the Windows service console.  
4. Right-click on the `kCura Data Grid Service`, and then click **Start**.  
5. Open a browser and navigate to the following URL to test the service: `http://localhost:9400/api/RelativityDataGridREST/GetCores`  
   The page should look similar to the following image:
8.2 Uninstalling the Data Grid service

Beginning in Relativity 9.3.376.35, you no longer need to install the DataGridRESTService on the Analytics server to integrate Relativity Analytics with Relativity Data Grid.

Use the following steps to uninstall the kCura Data Grid service:

- Launch the Windows command prompt and type the following:
  - `cd c:\DataGridRESTService` - navigates the command prompt to the correct Relativity Analytics directory.
  - `installutil.exe /u Relativity.DataGrid.Service.exe` - uninstalls the kCura Data Grid service.
9 Cluster monitoring tools

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.

9.1 Head

The Data Grid head console provides the following features:

- **Cluster Overview** - this console displays the general structure of your cluster and provides an interface you can use to perform operations on indexes and nodes.

- **Search interfaces** - allow you to perform queries from the head console that retrieve JSON results.

- **RESTful API inputs** - these inputs allow you to perform get, put, post, and delete requests on the data in the cluster. The head console also includes a JSON validator, a timer to perform repeat requests, JavaScript transformation functionality to reformat the JSON results, comparison tools to compare the results of your queries over time, and bar graph data visualization.

Install the head plugin on a master or client node on the cluster. Use the following steps to install the head plugin.

1. Open the command prompt on the chosen node and navigate to the following folder:

\C:\RelativityDataGrid\bin

2. Execute the following command:

```
/plugin -install mobz/elasticsearch-head
```

9.1.1 Performing offline installations of Head

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

1. Download the installer from the following URL from a machine that has internet access.

https://github.com/mobz/elasticsearch-head/archive/master.zip

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

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

```
C:\RelativityDataGrid\elasticsearch-main>bin\plugin -i head -u file:///C:/elasticsearch-head-master.zip
```

See [this site](#) for more information on using the head console.
9.2 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.

The Marvel plugin provides the following features:

- **Overview dashboard** - displays a two table overview of the general metrics necessary to monitor the overall health of your cluster. Metrics that require your attention are automatically highlighted in yellow or red.
- **Node Statistics dashboard** - displays hardware level metrics from the perspective of one or more nodes.
- **Index Statistics dashboard** - displays metrics aggregated across the cluster from the perspective of one or more indexes.
- **Shard Allocation dashboard** - displays shard status, allocation history, and how the shards are currently allocated across nodes.
- **Cluster Pulse dashboard** - allows you to investigate a variety of events that occurred throughout the history of the cluster.
- **Sense developer console** - allows you to research your data and perform administrative work with the Data Grid API.

Beginning in Data Grid 2.1.2, Marvel and Sense are no longer plugins of Elastic. If you are using Data Grid 2.1.2, use the following steps to install Marvel and Sense on the monitoring cluster. For Data Grid versions below Data Grid 2.1.2, see [Installing Marvel in Data Grid versions below Data Grid 2.1.2](#).

1. Open the command prompt and navigate to the kibana/bin folder.
2. Enter the following command to install Marvel:
   
   ```
   kibana plugin --install elasticsearch/marvel/2.1.2
   ```

   **Note:** Marvel versions must match the Data Grid version.

3. Enter the following command to install Sense:

   ```
   kibana plugin --install elastic/sense
   ```


[Installing Marvel in Data Grid versions below Data Grid 2.1.2](#)

Use the following steps to install Marvel on all nodes in the cluster:

**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.
1. Open the command prompt and navigate to the following folder:
   
   ```
   C:\RelativityDataGrid\bin
   ```

2. Enter the following command:
   
   ```
   .\plugin -install elasticsearch/marvel/latest
   ```

### 9.2.1 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:
   
   ```
   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:
   
   ```
   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.

### 9.3 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):
   
   ```
   pip install elasticsearch-curator
   ```

### 9.3.1 Performing offline installations of Curator

Use the following steps to download and install Curator on a machine that doesn’t have access to the internet:
1. Use a machine that has access to the internet to download the Curator Windows binary file from the following site:

   http://packages.elasticsearch.org/curator/3/windows/curator-3.2.2-win64.zip

2. Move the file to the Data Grid master node for installation.

3. Unzip the file to a folder on the root directory of the master node.

   **Note:** You don’t need Python installed. The binary package has all necessary Python libraries included. If the master node has Python installed, don’t unzip the folder into the Python directory.

4. Test that Curator has been installed and is working by running the following command:
   In Windows PowerShell:
   ```
   > .\curator show indices --prefix "audit"
   ```
   In Windows command prompt:
   ```
   > curator show indices --prefix "audit"
   ```

Use the following steps to create backups for offline installations of Curator:

1. Launch **Windows PowerShell ISE**.
2. Click **View** and select **Show Script Pane**.
3. Paste the following text into the script pane:
   ```
   C:\thisisthepathyourcuratordirectory snapshot --repository elasticbackup indices --all-indices
   ```
4. Verify that the path to the Curator folder is correct.
5. Edit the repository name to match the one in your environment.
6. Execute the command once from within the Windows Powershell ISE to ensure that it executes and the path is correct.
7. Save this command to a file, and set up a Windows scheduled task to call this command on a schedule. See **Setting the script as a scheduled task on page 91** for instructions.
8. Click **Save** as and navigate to the directory where Elasticsearch is installed for Relativity Data Grid. The default location is C:\RelativityDataGrid

See [this site](#) for further documentation on using Curator.
10 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.

10.1 Monitoring bulk rejections

Bulk rejections are important phenomena to monitor. You can access information on bulk rejections through Marvel.

Use the following steps to access information on bulk rejections:

1. Navigate to Marvel's Overview dashboard (http://any-server-in-cluster:9200/_plugin/marvel/)
2. Select the **Node Statistics** tab.
3. Click **THREAD POOLS - BULK**.

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™.
# 11 Querying with Marvel Sense

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

This page contains the following sections:

- [Retrieving documents and cluster settings below](#)
- [Shutting down a cluster on page 83](#)
- [Backing up clusters on page 84](#)

## 11.1 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:

```
GET /relativity_relativity_edds1019984_10/document/_search?size=5
{
  "query": {
    "match_all": {}
  }
}
```

Use the following query to retrieve the documents containing a specified term:

```
POST /relativity_relativity_edds1019984_10/document/_search
{
  "query": {
    "query_string": {
      "query": "happy",
      "fields": [
        "Fields.ExtractedText.Value"
      ]
    }
  }
}
```

Use the following queries to retrieve cluster health information:
Use the following query to retrieve all nodes detected by the master node:
GET /_cat/nodes?v

Use the following query to retrieve information on the allocation of shards:
GET /_cat/allocation?v

Use the following queries to retrieve the state of all indexes in the cluster:
GET /_cat/indices
GET /_cat/indices?v
GET /_cat/indices/logstash*
GET /_cat/indices/index_name*

Use the following query to create an index:
POST /index_name

Use the following query to delete an index:
DELETE /index_name*

Use the following queries to retrieve all aliases:
GET /_cat/aliases
GET /_cat/aliases/relativity_integration_edds9999*

Use the following query to update an alias:
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 query to create a template:

```json
PUT /_template/template_name
{
  "order": 0,
  "template": "relativity*",
  "settings": {
    "index.number_of_replicas": "0",
    "index.analysis.analyzer.lwhitespace.tokenizer": "whitespace",
    "index.number_of_shards": "10",
    "index.analysis.analyzer.lwhitespace.filter.0": "lowercase"
  },
  "mappings": {
    "document": {
      "dynamic_templates": [
        { "raw": {
            "mapping": {
              "type": "string",
              "fields": {
                "raw": {
                  "analyzer": "lwhitespace",
                  "type": "string"
                }
              }
            },
            "path_match": "Fields.*.Value"
          }
        }
      ],
      "_size": { "enabled": true, "store": true }
    }
  },
  "aliases": {
    "{index}_read": {},
    "{index}_write": {}
  }
}
```

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

- `POST _flush?wait_if_ongoing=true`
- `POST _refresh`
- `POST /relativity_integration_484a7590-3d90-4215-b5ab-e29aa9b99b8_edds9999_10/_refresh`
- `POST _optimize`

### 11.2 Shutting down a cluster

The following commands change cluster settings, and should only be used before cluster shutdowns:

Use the following command to disable shard allocation:
Use the following command to enable shard allocation:

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

Use the following command to set the number of concurrent shard rebalances per node:

```json
PUT _cluster/settings
{
  "transient": {
    "cluster.routing.allocation.cluster_concurrent_rebalance": 4
  }
}
```

Use the following command to set the number of concurrent recoveries per node:

```json
PUT _cluster/settings
{
  "transient": {
    "cluster.routing.allocation.node_concurrent_recoveries": 4
  }
}
```

Use the following command to shut down all nodes in a cluster:

```bash
POST /_shutdown
```

**Note:** There is no way to automatically restart all nodes in a cluster. You must manually restart each node if you shut down all the nodes with this command.

### 11.3 Backing up clusters

Use the following commands for backup and restore operations:

Use the following command to get all backup repositories:

```bash
GET /_snapshot/_all
```

Use the following command to create a backup repository:

```bash
PUT /_snapshot/my_backup
{
  "type": "fs",
  "settings": {
```
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"
}
```
12 Backing up Relativity Data Grid

We recommend you routinely back up your data. Data Grid 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 the Data Grid™ data store.

Note: Data Grid supports Windows servers only.

12.1 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 click Properties.
3. Select the Sharing tab, and click Share.
4. Enter the user that runs the Elasticsearch Windows service (domain\account), and 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. Near the top right under the Dashboards drop-down launch **Sense**.

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/
```

### 12.2 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](#)

#### 12.2.1 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 following article on the Elasticsearch website:


### 12.2.2 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. For more information on Curator and snapshot capabilities, please see the following website:
12.2.2.1 Creating the snapshot script

Online installations of Curator require Python, so you must download and install it to a machine that can communicate with the cluster. While not required, we recommend using the master node throughout this process.

Use the following steps to create the snapshot script:

1. Download and install the Python 3.4.2 MSI installer for Windows from on Python.org. [Visit the following website: https://www.python.org/downloads/release/python-342/](https://www.python.org/downloads/release/python-342/)

   **Note:** This version of Python includes PIP, which is used to install Curator.

2. Download and install, on the master node, the latest version of Git for Windows from [git-scm.com](https://git-scm.com). [Visit the following website: http://git-scm.com/download/win](http://git-scm.com/download/win) During the installation select the option to allow for usage in Windows command prompt.

3. Launch Windows Command Prompt on the master node as an Administrator.

4. Enter the following command to download Curator:

   ```
   git clone https://github.com/elasticsearch/curator.git
   ```

5. Enter the following command to change the directory to the location of the Python scripts:

   ```
   cd C:\Python34\Scripts
   ```

6. Next, run this: `pip install elasticsearch-curator==3.4.1`.

7. Launch **Windows PowerShell ISE**.

8. Click **View** and select **Show Script Pane**.

9. Paste the following text into the script pane:

   ```
   C:\Python34\Scripts\curator snapshot --repository elasticbackup indices --all-indices
   ```

10. Verify that the path to the Python installation is correct.

11. Edit the repository name to match the one in your environment.

   **Note:** Curator commands are case sensitive.

12. Execute the command once from within the **Windows Powershell ISE** to ensure that it executes and the path is correct.

13. Save this command to a file, and set up a Windows scheduled task to call this command on a schedule.

14. Click **Save as** and navigate to the directory where Elasticsearch was installed for Relativity Data Grid. The default location is `C:\RelativityDataGrid`

See **Performing offline installations of Curator on page 78** for instructions on installing Curator on a node that isn't connected to the internet.
12.2.2.2 Deleting snapshots older than a specified period of time

You can use the following command to delete backups over seven days old from Curator using the Windows command prompt:

```bash
C:\Python34\Scripts\curator delete snapshots --older-than 7 --time-unit days --repository ElasticBackup
```

You must specify the snapshot repository name at the end of above command. You can determine the repository name by running the following command in Marvel Sense:

```bash
GET /_snapshot/
```

Set the script to run periodically as a scheduled task.

12.2.2.3 Deleting indexes older than a specified period of time

You can create a CMD file and enter the following text to delete indexes over 45 days old from Curator:

```bash
@ECHO OFF
SET curator_location=C:\Python34\Scripts
CD "%curator_location%"
curator --host monitoringnodename --port 9200 delete indices --prefix .marvel-es--timestring '%%Y.%%m.%%d' -older-than 45 --time-unit days
```

**Note:** Replace "monitoringnodename" with the name of the monitoring node. This query assumes shield is disabled on the Monitoring node.

Set the script to run periodically as a scheduled task.

**Note:** Verify that Python is installed to the same directory as Curator when running this script.

12.2.2.4 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 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 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 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 click New.

14. Set the Action to Start a program.

15. Enter "Powershell" in the Program/script field.

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 Administrator each time the script is run.

20. Click **OK**.

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

### 12.3 Restoring a snapshot

There are multiple methods for restoring snapshots. Restoring snapshots from the Data Grid 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.

#### 12.3.1 Restoring snapshots from the Data Grid head console

You can restore a snapshot from the Data Grid 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.

#### 12.3.2 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 from the following site: [http://www.confusedbycode.com/curl/](http://www.confusedbycode.com/curl/).

**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 in 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 Data Grid, the following site:

13 Data Grid search

The Data Grid search option provides you with a way to search on long text fields stored in Data Grid for any Data Grid-enabled workspaces in your Relativity environment. Once you enable Data Grid search, the Data Grid option is available in the search drop-down, along with your Keyword Search, dtSearch, and Analytics indexes. The Data Grid search includes single-term search, exact phrase search, wildcards, fuzziness, proximity, Boolean operators, and grouping.

This search functionality is immediately available as text is imported or published into Data Grid. To use Data Grid search, your workspace and long text fields must be enabled for Data Grid.

**Note:** Beginning in Relativity 9.3.297.13, you no longer have to include a backslash in front of quotation marks, dollar signs, apostrophes, or percent signs.

13.1 Using Data Grid search

Once you’ve enabled your workspace for Data Grid and activated the Data Grid index, you can use it through the following steps:

1. Select **Data Grid** from the Search drop-down.

2. Enter terms for the search in the **Search Terms** box. See Data Grid search syntax considerations on page 101. You can click **Clear** to remove the current search terms from the box.

3. (Optional) To display fields for search conditions, click the search conditions icon next to the "search with" menu.

4. Click **Search**. To stop a long running search, click **Cancel Request**.

**Note** the following regarding Data Grid search indexes:

- Once you create a Data Grid search index, you can select it from the Index field on a new Search Terms Report (STR). This means that you have the option of selecting between dtSearch and Data Grid search indexes when creating an STR. For more information, see the Search terms reports section of the Admin Guide.

- When you run a Data Grid search, Relativity caches the results in a search cache table in SQL in the workspace database so that subsequent runs of the query are faster. By default, all cache results are valid until you modify any data in the index. The DataGridSearchProviderCacheTimeout instance setting determines how often Relativity checks the index for updates. When the off-hour agent runs, the Data Grid Manager agent is responsible for cleaning up expired search cache tables.
Errors due to invalid search queries can occur during Data Grid searches. If necessary, you can view these errors in the Errors tab in Relativity.

### 13.2 Using Data Grid search in the new UI framework

To create and run a Data Grid search in the new UI framework, you must first enable the new UI framework in the user drop-down.

1. Click **Add Condition** in the search panel.
2. Select *(Index Search)*.
3. Select **Data Grid**, then enter terms in the **Search Terms** box and click **Apply**.
   Optional: Select the check syntax check box before clicking **Apply** to check the syntax of your Data Grid search terms before adding new terms. For more information, see [Data Grid check syntax](#).

![Data Grid search panel](image)
4. When you return to the Documents list, click **Run Search** at the bottom of the conditions list to execute a search on your terms.

5. Refer to the Documents list to see which documents were returned by your query.

1. Click
to access the Search browser from the Documents List.

2. Click **New Search**.
3. Enter required fields in the Information card.
4. Click **Add Condition**.
5. Select **(Index Search)** from the Add Condition dropdown. The (Index Search) window opens.
6. Select the name of your Data Grid index from the Index dropdown.
7. Enter terms for the search in the Search Terms box. For more information, see Data Grid search syntax considerations on page 101.
8. (Optional) Add any additional search conditions.
9. Click **Apply**. The search terms populate in the Search builder.
10. Click **Save** or **Save As**.
11. Select the name of the dtSearch in the Search Browser.
12. Click **Run Search**. To stop a long running search, click **Cancel**.

### 13.3 Data Grid check syntax

Even though Data Grid search supports many of the same operations as dtSearch, the syntax for running a Data Grid search is slightly different. Beginning in Relativity 9.3.376.35, you now have the option to check the syntax of your Data Grid search terms before adding new terms. This option is only available if you've selected a Data Grid search index.

When you enable the check syntax checkbox, Relativity checks for incompatible dtSearch syntax for any of the following:

- Search operators
- Proximity search
- Stemming search
- Fuzzy search
- Regular expressions

**Note:** Changing the checkbox setting updates the default property for the logged in user.

If you’re using incompatible syntax (such as apple w/5 pear), Relativity displays a warning message with the following options:

- **Add Anyway** - ignores the syntax errors that exist in your search terms and adds the terms to your search.
- **Edit Terms** - closes the syntax warning and allows you to edit your remaining terms.
Note: When you select a Data Grid search index for a Search Terms Report, you have the check syntax option when entering new terms for that STR. For more information, see the Relativity User Guide.

13.3.1 Using check syntax for Data Grid search in the new UI framework

1. If the syntax checker is disabled, select the checkbox next to Check Syntax.

![Image of syntax checker interface]

Note: Changing the checkbox setting updates the default property for the logged in user.

2. Enter your terms in the Search Terms box and click Apply. Relativity checks for any dtSearch syntax that is not compatible with Data Grid search (such as "apple w/5 pear").

3. You have the following options:
   - **Apply Anyway** - ignores the syntax errors that exist in your search terms and adds the terms to the conditions list in the search panel.
   - **Edit Terms** - closes the syntax warning and returns you to the Search Terms box where you can edit your terms.
13.4 Data Grid search syntax considerations

The following sections provide descriptions and examples of the query syntax that Data Grid supports:

Note the following about Data Grid search syntax:

- Relativity doesn’t currently provide a configurable alphabet file for Data Grid search. You can search all characters with the exception of special characters, which Data Grid interprets as spaces. These are + - = && || > < ! ( ) { } [ ] ^ " " * ? : \ /.

  Note: Data Grid indexes commas and periods only when they are surrounded for text. For more information, see Special characters.

- Data Grid search uses a standard tokenizer, which means that it’s optimized for searching on Western European characters, but it’s not optimized for CJK languages. For more information on the language analyzers Data Grid runs on, see this article.

- Data Grid currently doesn’t support a default stop/noise word list. As such, every word within an indexed document field is included in the text.

- Use upper case for all operators in your search strings. For example, AND, OR, and NOT.

- Unlike dtSearch, Data Grid search does not support stemming.
**Note:** Unlike dtSearch, Data Grid search looks in a single field for multiple queried terms and only counts the document a hit if that field contains all of those terms. For example, if the Extracted Text field contains “Johnny” and the Email from field contains “Appleseed”, a Data Grid search query for *Johnny AND Appleseed* would not hit on that document. You aren’t able currently to arrange for Data Grid search to search across multiple fields for different terms, nor can you limit that search to one specific field.

### 13.4.1 Phrases

To search for a phrase (two or more terms), surround the phrase with quotes.

- Make sure to type the quotes in the search box instead of copying and pasting so that the quote formatting isn’t copied over.

Example:

“apple pear”

The following graphic depicts which documents Data Grid search returns when you execute a search for a phrase:

**Phrase in Data Grid search**

![Diagram showing how Data Grid search returns documents based on a phrase]

### 13.4.2 Question mark single wildcard

You can run wildcard searches on individual terms. Use ? to replace a single character.

Example:

`appl?`

The following graphic depicts which documents Data Grid search returns when you execute a search that includes the ? wildcard:
13.4.3 Asterisk wildcard
You can use * to replace zero or more characters.

Example:

appl*

The following graphic depicts which documents Data Grid search returns when you execute a search that includes the * wildcard:

13.4.4 Fuzziness
Using Data Grid search, you can perform fuzzy searches, which return documents containing spelling variations of a specified term. You may want to use fuzzy searching when querying documents that contain misspelled terms, typographical errors, or have been scanned with Optical Character Recognition (OCR).

You can use fuzziness, in conjunction with a whole number value between 0 and 2, in a Data Grid search string. When you run a fuzziness search, we recommend specifying a numerical value after the ~ to
specify the edit distance. This number value is the count of the replacements, inserts, delete, or switches of adjacent characters. If you do not specify an edit distance (and the Data Grid syntax checker is disabled), the query defaults to an edit distance of 1.

**Note:** The numerical value must be a whole number. Data Grid supports a maximum edit distance of 2.

Data Grid fuzziness search operates with four types of one-character edits:

- **Substitution** of one character for another: h_ats → m_ats
- **Insertion** of a new character: mats → m_e_ats
- **Deletion** of a character: meat_s → meat
- **Transposition** of two adjacent characters: me_at → me_ta

The transformation from the word "hats" to the word "meta" reflects an edit distance of 4. The impact that a single edit has on a string depends on the length of the string. Specifying too large of an edit distance may produce results beyond the scope of what you were looking for. If you're performing a fuzzy search for "hats," you probably don't want to return results for "meta." Since 80% of human misspellings have an edit distance of 1, Data Grid supports a maximum edit distance of 2.

Use the following parameters to help determine your edit distance:

- 0 for strings of one or two characters
- 1 for strings of three, four, or five characters
- 2 for strings of more than five characters

**Note:** These parameters are merely a suggestion. You may find that an edit distance of 2 returns results that don't appear to be related, in which case you may get better performance with a maximum fuzziness of 1.

Example:

**apply~1**

The following graphic depicts which documents Data Grid search returns when you execute a search that includes the fuzziness operator:

**Fuzziness in Data Grid search**
13.4.5 Proximity

In Data Grid search, you can run a proximity search to return documents with two or more words occurring within a certain proximity of each other.

Note the following about proximity searching:

- When you run a proximity search you must include a numerical value after the ~ or your search will display no results. This number specifies the number of replacements, inserts, deletions, or switches of adjacent words required for a match (i.e., the edit distance).

- The proximity operator is symmetrical. While a phrase query (e.g., "john smith") expects all of the terms in exactly the same order, a proximity query allows the specified words to be further apart or in a different order. This means that the search expression "apple orange"~5 returns the exact same document as "orange apple"~5.

Example:

"apple orange"~3

The following graphic depicts which documents Data Grid search returns when you execute a search that includes proximity operators:

![Proximity in Data Grid search](image)

13.4.6 AND operator

To search for documents that contain two or more terms, use the AND operator. You can’t start a search with AND.

Examples:

apple AND pear

apple && pear

The following graphic depicts which documents Data Grid search returns when you execute a search that includes the AND operator:
13.4.7 OR operator
To search for documents that contain either of two or more terms, use the OR operator. You can't start a search with OR.

Examples:
apple OR pear
apple || pear

The following graphic depicts which documents Data Grid search returns when you execute a search that includes the OR operator:

13.4.8 NOT operator
To search for documents that contain one term but specifically not another, use the NOT operator.
- NOT affects only terms to the right of the operator.
- You can start a search with NOT.

Examples:
- NOT apple
- apple NOT pear
- apple ! pear

The following graphic depicts which documents Data Grid search returns when you execute a search that includes the NOT operator:

13.4.9 Regular expressions
You can search for regular expressions in Data Grid search. RegEx queries must be surrounded by forward slashes `/`.

Examples:
- /joh?n(ath[oa]n)/
- /abab(ab)?/

The following table describes the special characters that are allowed for regular expressions in Data Grid search:

<table>
<thead>
<tr>
<th>Character</th>
<th>Result</th>
</tr>
</thead>
<tbody>
<tr>
<td>Period (.)</td>
<td>The period &quot;.&quot; can be used to represent any character.</td>
</tr>
<tr>
<td>Question mark (?)</td>
<td>The question mark &quot;?” makes the preceding shortest pattern optional. It matches zero or one times.</td>
</tr>
<tr>
<td>Plus sign (+)</td>
<td>The plus sign &quot;+&quot; can be used to repeat the preceding shortest pattern once or more times.</td>
</tr>
<tr>
<td>Asterisk (*)</td>
<td>The asterisk &quot;*&quot; can be used to match the preceding</td>
</tr>
</tbody>
</table>
### 13.4.10 Grouped queries

You can group multiple terms or clauses together with parentheses, to form sub-queries.

- The operator precedence is NOT, then AND, then OR.
- Use a + to indicate a word must be included.
- Use a – to indicate that a word must not be included.

Examples:

- (apple AND dog) OR cat
- (apple && dog) || cat
- Cat NOT dog
- Cat !dog
- cat +puppy -bird
13.4.11 Special characters
You can’t search for special characters in Data Grid search.

- Data Grid interprets special characters as spaces.
- For periods that are surrounded by text, the period character is indexed with the text.
- For commas that are surrounded by text, the comma character is indexed with the numbers.

Examples:
A search for jdoe@example.com is indexed as:
  jdoe
  Example.com
A search for $300,000 is indexed as:
  300,000

13.4.12 Unsupported dtSearch syntax in Data Grid search
Data Grid does not support the following dtSearch syntax:

- pre/#
- w/# - Data Grid supports proximity searching with the syntax described above.
- xlastword
- xfirstword
- AndAny operator
- Stemming
14 Data Grid infrastructure optimization

This topic describes various factors that contribute to Data Grid™ performance and provides you with information related to optimizing your Data Grid cluster. See the Data Grid system requirements topic for information on determining your tier structure.

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

14.1 Virtual vs. physical servers

We strongly recommend physical servers but there are various ways to achieve the goal of running Data Grid in a virtualized environment. Each platform and solution (cloud-based or not) has unique elements of complexity related to administrating resources including CPU, memory, disk space, and network connectivity. See the following site for more information about virtualizing Data Grid: [https://www.elastic.co/blog/how-to-handle-elasticsearch-virtualization](https://www.elastic.co/blog/how-to-handle-elasticsearch-virtualization).

Because it's common to have more than one active VM running on a single piece of hardware, you can use shard allocation awareness to prevent primary and replica shards from simultaneously residing on the same hardware unit. This allows you to avoid the risk of losing data in a virtualized environment by forcing each replica shard into a VM that isn't on the same hardware unit as the primary shard.

14.2 Storage types

Consider the following factors when selecting storage types for Data Grid:

- Data Grid is a distributed system that you should run on storage local to each server. Regular disks are acceptable in tier 1 setups, but consider SSDs in tier 2 and tier 3 setups for their added performance benefits.
- Don't place the index on a remotely mounted file system (e.g., NFS or SMB/CIFS); use storage local to the machine instead.
- Use modern solid-state disks (SSDs) because they are far faster than even the fastest spinning disks. They have lower latency for random access, higher sequential IO, and are better at highly concurrent IO required for simultaneous indexing, merging, and searching.
- Stripe your index across multiple SSDs by setting multiple path.data directories or configuring a RAID 0 array. The two are similar, except instead of striping at the file block level, Data Grid "stripes" at the individual index files level. Either approach increases the risk of failure for a single shard in exchange for faster IO performance because the failure of any one SSD destroys the index. Optimize single shards for maximum performance and then add replicas across different nodes to create redundancy for any potential node failures. You can also create frequent snapshots of the index for further insurance. See the following site for more information: [https://www.elastic.co/blog/performance-considerations-elasticsearch-indexing](https://www.elastic.co/blog/performance-considerations-elasticsearch-indexing).

14.3 Network connectivity

Network connectivity can impact performance due to the distributed architecture of Data Grid (especially during peak activity). Consider 10 GBit Ethernet for higher tier levels.
Users with 10 GB networks get better performance during recovery times and snapshots consisting of large, bulk file movements.

### 14.4 Shard settings

A shard is a container for your data that's distributed across your server’s nodes. If you set your shard setting too high, it can create performance problems when writing and searching your data; however, if you set your shard setting too low, you can’t improve performance by adding new nodes to your cluster. The number of shards on a node should generally be equal to two times your total number of data nodes. That means you can double your data nodes before you stop seeing performance gains. Take that into consideration if you have plans to increase the number of nodes on your cluster.

<table>
<thead>
<tr>
<th>Tier 1</th>
<th>Tier 2</th>
<th>Tier 3</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Recommended shard settings</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>6 shards total with 2 shards per node</td>
<td>6 to 12 shards total with 2 shards per node</td>
<td>2 shards per node regardless of total number of shards.</td>
</tr>
<tr>
<td>9 shards total with 3 shards per node</td>
<td>9 to 18 shards total with 3 shards per node</td>
<td></td>
</tr>
<tr>
<td><strong>Limitations</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Maximum index size: 300 GB to 450 GB (~600 GB to 900 GB SQL data)</td>
<td>Maximum index size: 600GB to 900GB (~1200GB to 1800GB SQL data)</td>
<td>Maximum index size: 600GB at 6 nodes</td>
</tr>
<tr>
<td></td>
<td></td>
<td>900GB at 9 nodes</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1500GB at 15 nodes</td>
</tr>
</tbody>
</table>

*We recommend a setting of two shards per node, but to store larger indexes and add more nodes easily in the future, three shards per node is an acceptable setting. Using three versus two could decrease your performance.

**There is a limit of 50 GB per shard, so there is a maximum size for your cases based on your shard settings and number of nodes.
15 Data Grid for Audit

Data Grid™ for Audit is an application that allows you to monitor and run reports on audited user activity. Data Grid for Audit provides insight into review planning for productivity analysis and enables you to easily find reviewer or propagation coding values for follow-up action.

Admins might also use Data Grid for Audit for performance investigation into client complaints, server alerts, and performance tuning. It can also be a convenient tool for monitoring servers and performing health checks. You must have the Data Grid data store configured to use the Data Grid for Audit application.

In the Data Grid for Audit application, you can add widgets for custom visualization of audit data, as well as custom dashboards.

See Data Grid for more information on setting up the preliminary requirements.

15.1 Installing the Data Grid for Audit application

To use Data Grid for Audit in your workspace, you must install the Data Grid for Audit application. To do so, perform the following steps:

1. Navigate to the Relativity Applications tab.
2. Click New Relativity Application in the upper left corner of the All Relativity Applications view.
3. In the Application Type field, select the radio button for Select from Application Library.
4. Click ⋅⋅⋅ in the Choose from Application Library field.
5. Locate and select the Data Grid for Audit application in the list of available applications and click Ok.
6. Once Relativity loads the application, click **Import** to install it to your workspace.
7. Configure the Data Grid for Audit instance settings. For more information, see Data Grid instance settings on page 61.

15.2 Home tab

The Home tab of the Data Grid for Audit application allows you to configure and apply a number of different filters and widgets to create custom visualizations of the audit data in your workspace. To get started, you can either add a widget to display all of the unfiltered audit data, or you can select from the available filters in the Filter drop-down.
To apply a filter in the Home tab, perform the following basic steps. Note that each filter type is described in the sections below.

1. Click the **New Filter** drop-down.
2. Select from the available filter types. Each filter type is described below.
3. In the subsequent filter window, enter the parameters of the filter type you selected above. For example, query text, number, name, date, etc.
4. If needed, add a condition to the original parameter by clicking the **+ add condition** option, when available.
5. Click **Apply**. The filter is visible in the pane on the left hand side of the screen.
To remove a single filter from the list, simply click the X in the top right corner of that filter.

To disable a filter without removing it, un-check the box in the bottom right corner of the filter.

To remove all of the filters in the pane, select Clear all Filters.

Any data you’d already displayed is updated immediately to reflect the filter you just added, assuming that the filter affected that data.

If you add an over-exclusive filter that completely eliminates the data that meets the filter’s criteria, any widgets you have currently displayed will become blank. Once you disable or remove the over-exclusive filter, the widgets will return to their previous state and display data according to the other filters you applied.

6. To determine what kinds of data you wish to display based on the filters you’ve added, continue to add widgets through the Add Widget button.
The following sections provide information on each filter type.

15.2.1 Details filter
The Details filter allows you to view the details of an audit record. Select Details from the filter type dropdown to open the window in which you'll enter your text query.
Wildcards and boolean operators aren’t supported in the Details filter.

You can use the text box to enter a word or phrase to filter results by a specific string contained in the audit record details.

Only audit records that contain the entered text appear among the data inside the widgets you select.

Click the + add condition link to add another term to the filter. Each condition applies a Boolean "OR" operation.

### 15.2.2 ID filter

The ID filter lists the audit record’s ID number. Select ID from the filter type drop-down to bring up the window in which you’ll enter the number of at least one audit record.

Enter the number of the audit record you want to display among the data inside the widgets that you select.

Only audit records that have the entered ID number(s) appear in the data set when you apply this filter.

Click the + add condition link to add another number to the filter. Each condition applies a Boolean "OR" operation.
15.2.3 Timestamp filter

The Timestamp filter lists the date and time of the event’s occurrence. Select **Timestamp** from the filter type drop-down to bring up a window containing the operators you’ll use to locate audit records with certain timestamps.

- **Operator** - this drop-down provides you with the following options:
  - **is after or on** - limits the audit data set to records generated after or on the specified date and time.
  - **between** - limits the audit data set to records generated between two specified dates and times.
  - **is before or on** - limits the audit data set to records generated before the specified date and time.
  - **Custom range** - this menu item provides you with the following options:
    - **The Past** - allows you to enter the number of units determined by the next drop-down filter. You have the option of selecting seconds, minutes, hours, days, weeks, months, or years from the drop-down menu.
    - Click **add condition** link to add multiple date filters. Each condition applies a Boolean "OR" operation.

- Click the **MM/DD/YYYY** text box to open a calendar. Select a date on the calendar according to your filtration needs. You can also enter the date in the text box with your keyboard. If your filtration strategy requires you to filter by a specific time of day, enter the hour, minute, and second in the following format: HH:MM (i.e. - 09:06) in the second text box. Enter the earliest date in the text box on the left (MM/DD/YYYY), then enter the latest date in the text box on the right.

- Click **add condition** link to add multiple date filters. Each condition applies a Boolean "OR" operation.
15.2.4 Name filter
The Name filter lists audit records by name. Select Name from the filter type drop-down to open a window in which you'll enter the name of an audit record.

- Only audit records that have the entered name appear in the filtered data set when you apply this filter.
- You can use % as a wildcard in your text query.
- Click the + add condition link to add another name to the filter. Each condition applies a Boolean "OR" operation.
- You can filter the list according to strings contained within the name text by entering a string in the text box filter.

15.2.5 Action filter
The Action filter allows you to filter your data based on the audit actions performed in the workspace. Select Action from the filter type drop-down to bring up a window in which you'll select the audit actions you wish to filter upon.
There are two columns in the pop-up window: Available and Selected. The Available column lists all available user actions you can use to filter the audit data, and the Selected column lists all applied action filters. You can enter the name of an action in the text box to jump to an action in either list. You can also sort the list in either column in Ascending or Descending alphabetical order with the drop-down menu.

See a full list of actions
You can apply a filter for one or more of the following actions:

<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>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>
</tbody>
</table>
### Action name

<table>
<thead>
<tr>
<th>Action name</th>
<th>Description of activity</th>
</tr>
</thead>
<tbody>
<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>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.</td>
</tr>
<tr>
<td>Production - Add Documen</td>
<td>A document was added to a production.</td>
</tr>
<tr>
<td>Production - Remove Docu-</td>
<td>A document was removed from a production.</td>
</tr>
<tr>
<td>ment</td>
<td></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>An Imaging Set, Image-on-the-Fly, or Mass Image job was performed</td>
</tr>
<tr>
<td>Search Cache Table Cre-</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>ation</td>
<td></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>

### 15.2.6 Object Type filter

The Object Type filter allows you to filter your data based on the type of object on which the action was performed. Select **Object Type** from the filter type drop-down to bring up a window in which you'll select the object types on which you'd like to filter your data. The Available window to the left lists every object type that has ever existed in the workspace.
Items in the Available list are not applied by the filter, and items in the Selected list are applied by the filter. To move an item from one list to the other, click on it to select it, and use the arrow button to move it into the other list. You can select multiple entries at once by holding down the CTRL key while clicking each item. You can also select all items between two items you select by holding down the Shift key during selection.

The following table lists all the object types that are available by default. Note that if you've created more objects in your workspace, those additional objects will appear here.

<table>
<thead>
<tr>
<th>Object Types</th>
<th>Available</th>
<th>Selected</th>
</tr>
</thead>
<tbody>
<tr>
<td>AgentType</td>
<td>ImagingProfile</td>
<td>RelativityApplication</td>
</tr>
<tr>
<td>AnalyticsCategorizationResult</td>
<td>ImagingSet</td>
<td>RelativityApplication</td>
</tr>
<tr>
<td>AnalyticsCategorizationSet</td>
<td>InstallEventHandler</td>
<td>RelativityApplication</td>
</tr>
<tr>
<td>AnalyticsCategory</td>
<td>Layout</td>
<td>RepeatedContentFilter</td>
</tr>
<tr>
<td>AnalyticsExample</td>
<td>Lists</td>
<td>Report</td>
</tr>
<tr>
<td>AnalyticsProfile</td>
<td>MarkupSet</td>
<td>Search</td>
</tr>
<tr>
<td>Batch</td>
<td>MassOperation</td>
<td>SearchFolder</td>
</tr>
<tr>
<td>BatchSet</td>
<td>NativeType</td>
<td>SearchIndex</td>
</tr>
<tr>
<td>Code</td>
<td>OCRProfile</td>
<td>SearchTermsReport</td>
</tr>
<tr>
<td>CustomPage</td>
<td>OCRSet</td>
<td>SearchTermsResult</td>
</tr>
<tr>
<td>Document</td>
<td>ObjectType</td>
<td>Tab</td>
</tr>
<tr>
<td>Email</td>
<td>PasswordBank</td>
<td>Transform</td>
</tr>
<tr>
<td>EventHandler</td>
<td>PasswordEntry</td>
<td>TransformSet</td>
</tr>
<tr>
<td>Field</td>
<td>PersistentHighlightSet</td>
<td>View</td>
</tr>
<tr>
<td>Folder</td>
<td>PivotProfile</td>
<td>VirtualField</td>
</tr>
<tr>
<td>Group</td>
<td>Production</td>
<td>Workspace</td>
</tr>
</tbody>
</table>
15.2.7 Execution Time filter

The Execution Time filter allows you narrow your search by duration of execution time. Select **Execution Time** from the filter type drop-down to bring up a window in which you'll specify the parameters you'll use to filter for execution times.

- **Options** - select one of the following radio buttons:
  - **Greater than or equal to** - enter a value in milliseconds to limit audit data to records whose execution time was equal to or greater than the value.
  - **Between** - enter two values in milliseconds to limit audit data to records whose execution time was equal to or greater than the first value, AND equal to or less than the second value. Enter the shortest execution time in the text box on the left, and enter the longest execution time in the text box on the right to filter the list by a specific execution time span. You must enter a value in both text boxes for this filter to work.
  - **Less than or equal to** - enter a value in milliseconds to limit audit data to records whose execution time was equal to or less than the value.

- Use the **+ add condition** link to create a filter with multiple Execution Time conditions. Each condition applies a Boolean "OR" operation.

15.2.8 Artifact ID filter

The Artifact ID filter allows you to filter your audit data based on specific ArtifactID numbers. Select **ArtifactID** from the filter type drop-down to bring up the window in which you'll specify the IDs you'd like to locate.
Enter an Artifact ID number to limit the results by that specific artifact.

Use the + add condition link to create a filter with multiple ArtifactID conditions. Each condition applies a Boolean "OR" operation.

15.2.9 User Name filter

The User Name filter allows you to filter your data based on by one or more user names associated with the audit actions in the workspace. Select **User Name** from the filter type drop-down to bring up a window in which you'll specify the user names you want to filter for.

Items in the Available list are not applied by the filter, and items in the Selected list are applied by the filter. To move an item from one list to the other, click on it to select it, and use the arrow button to move it into the other list. You can select multiple entries at once by holding down the CTRL key while clicking each item. You can also select all items between two items you select by holding down the Shift key during selection.
15.2.10 Advanced filter
The Advanced filter allows you to compose a custom query in JSON based on the Elasticsearch API. Select Advanced from the filter type drop-down to bring up a window in which you'll provide a description and specify a query to filter your data.

- **Description** - provide a description of the query here.
- **Query** - enter the actual text of the query here.

15.2.11 Saved Search filter
The Saved Search filter allows you to filter your audit data based on documents returned by any saved search available in your workspace. Select Saved Search from the filter type drop-down to bring up a window in which you'll select the saved searches you want to apply to the audit data.
Items in the Available list are not applied by the filter, and items in the Selected list are applied by the filter. To move an item from one list to the other, click on it to select it, and use the arrow button to move it into the other list. You can select multiple entries at once by holding down the CTRL key while clicking each item. You can also select all items between two items you select by holding down the Shift key during selection.

15.3 Saving and loading filter sets

You have the option of saving one or more applied filters as a filter set, which you can then treat as a template and load at a future date to filter your audit data. This prevents you from having to manually configure and apply the same individual filters you created before.

To do create a filter set, perform the following steps:
1. Click the save icon in the upper right corner of the filter pane.

2. In the Filter Set layout, complete the following fields.

   - **Name** - enter the name of the filter set.
   - **Order** - enter the order in which you'd like this filter set to appear in the Load Filter Set drop-down.

3. Click **Save** to save the filter set and make it available for selection in the Load Filter Set drop-down. Select **Save As** to save the set as a copy of one that you might have chosen to edit rather than overwriting the filter set you initially used as a template.

   The new filter set is now available for selection in the drop-down.
15.4 Adding widgets

Widgets allow you to visualize audit data according to your preferences.

To add a widget, perform the following steps:

1. Click **Add Widget** in the Home tab.

**Note:** You need to save the dashboard to save a widget to the dashboard. If you don’t save the dashboard and you refresh the page or leave the page and come back, the new widget will be gone.
1. Select from the available widget types.

2. Click **Add Widget** again to bring up the visualization of the audit data for that type.

Once you add a widget you can re-size, drag, and drop it to a new location in the Home tab.

The following sections provide information on how to interact with each widget type.

### 15.4.1 User Actions widget

The User Actions widget displays frequency metrics in a pie chart, bar graph, or grid format. You can switch between data visualization formats from the menu that appears on hover over the widget.

Click 🌠 to display the user actions as a pie chart. Hovering over any segment of the pie chart lists the user action, the audit count, and the percentage of the data set, and hovering over any action in the list in the right highlights the corresponding segment of the pie chart. You can click on any region of the pie chart to limit your results to user actions of that particular type. Relativity adds a filter indicator to the filter list when you use this method to limit the data set.
Click to display the user actions as a bar graph. Hovering over any bar displays the count. You can click on any bar to limit your results to user actions of that particular type. Relativity adds a filter indicator to the filter list when you use this method to limit the data set.

Click to display the user actions in table format. The User Actions column displays the name of the user action, and the Count column displays the number of user actions that occurred within the time span.
limitations of your data set. You can click on any row in the list to limit your results to user actions of that particular type. Relativity adds a filter indicator to the filter list when you use this method to limit the data set.

<table>
<thead>
<tr>
<th>User action</th>
<th>Count</th>
</tr>
</thead>
<tbody>
<tr>
<td>Create</td>
<td>79668</td>
</tr>
<tr>
<td>Update</td>
<td>4260</td>
</tr>
<tr>
<td>Document Query</td>
<td>1385</td>
</tr>
<tr>
<td>Query</td>
<td>596</td>
</tr>
<tr>
<td>Search Cache Table Creation</td>
<td>234</td>
</tr>
<tr>
<td>Delete</td>
<td>24</td>
</tr>
<tr>
<td>Update - Import</td>
<td>21</td>
</tr>
<tr>
<td>Workspace Upgrade</td>
<td>16</td>
</tr>
<tr>
<td>Import</td>
<td>12</td>
</tr>
<tr>
<td>Security</td>
<td>7</td>
</tr>
</tbody>
</table>

The following fields appear in the User Actions widget:

- **Create** - the number of create commands executed on the data set.
- **Update** - the number of updates to objects in the data set.
- **Document Query** - the number of document queries executed on the data set.
- **Query** - the number of SQL queries executed on the data set.
- **Search Cache Table Creation** - the number of search cache tables created in accordance with Data Grid search provider usage.
- **Delete** - the number of deletions executed on the data set.
- **Update - Import** - the number of updates made as a result of imports into Data Grid.
- **Workspace Upgrade** - the number of workspace upgrades on the data set.
- **Import** - the number of imports into Data Grid that were executed.
- **Security** - the number of security edits applied to the data set.
15.4.2 User Names widget

The User Names widget displays frequency metrics in a pie chart, bar graph, or grid format. You can switch between data visualization formats from the menu that appears on hover over the widget.

Click to display the user names as a pie chart. You can click on any region of the pie chart to limit your results to a particular user. Hovering over any segment of the pie chart lists the user name, the audit count, and the percentage of the data set, and hovering over any object type in the list in the right highlights the corresponding segment of the pie chart. Relativity adds a filter indicator to the filter list when you use this method to limit the data set.

Click to display the user names as a bar graph. Hovering over any bar displays the audit count. You can click on any bar to limit your results to a specific user. Relativity adds a filter indicator to the filter list when you use this method to limit the data set.
Click to display the user names in two columns. The User Name column displays the name of the user, and the Count column displays the number of times an action was performed by that user within the time span limitations of your data set. You can click on any row in the list to limit your results to a particular user. Relativity adds a filter indicator to the filter list when you use this method to limit the data set.
15.4.3 Object Types widget

The Object Types widget displays frequency metrics in a pie chart, bar graph, or grid format. You can switch between data visualization formats from the menu that appears on hover over the widget.

Click to display the object types as a pie chart. You can click on any region of the pie chart to limit your results to objects of that particular type. Hovering over any segment of the pie chart lists the object type, the audit count, and the percentage of the data set, and hovering over any object type in the list in the right highlights the corresponding segment of the pie chart. Relativity adds a filter indicator to the filter list when you use this method to limit the data set.

Click to display the object types as a bar graph. Hovering over any bar displays the count. You can click on any bar to limit your results to objects of that particular type. Relativity adds a filter indicator to the filter list when you use this method to limit the data set.
Click to display the object types in two columns. The Object Types column displays the name of the object type, and the Count column displays the number of times an action occurred on that object type within the time span limitations of your data set. You can click on any row in the list to limit your results to objects of that particular type. Relativity adds a filter indicator to the filter list when you use this method to limit the data set.
15.4.4 Actions over Time widget

The Actions over Time widget displays a line graph that shows the number of audited user actions over a specified period of time.
The drop-down menu that appears in the lower left corner of the Actions over Time widget allows you to configure the units of time in the following increments:

- Every 1 second
- Every 1 minute
- Every 5 minutes
- Every 10 minutes
- Every 30 minutes
- Every 1 hour
- Every 3 hours
- Every 1 day
- Every 1 week
- Every 1 month
- Every 1 year

You can also click and drag over a range within the Actions over time widget to apply a filter that narrows your results to the selected range.
15.4.5 Results widget

The Results widget displays all filtered results in a grid format. Select **Results** from the filter type drop-down to bring up a window resembling a standard Relativity item list.

The following information is listed in the Results widget view:

- **Details** - provides the option of viewing the query text of an audit record, as well as the full details of that record in both table and JSON formats. These options are discussed in detail below.
- **ID** - lists the audit ID number.
- **Time Stamp** - lists the date and time of the event's occurrence.
- **Name** - lists the title of the artifact on which the event was performed.
- **Action** - lists the name of the action performed. For example, Query.
- **ObjectType** - lists the type of object on which the event was performed.
- **ExecutionTime** - lists the duration of the event.
- **ArtifactID** - lists the artifact ID of the audit.
- **UserName** - lists the user name of the user who performed the action.

### 15.4.5.1 Expanded record view

To display the query text of an audit record, click + in the Details field for that record.

![Expanded record view](image)

The Name and Value columns display the following information in this view:

- **auditElement.Query Text** - provides the code comments and select parameters of the query text associated with the audit action.
- **auditElement.Milliseconds** - provides the time it took the system to execute the audit action.

### 15.4.5.2 Full Details window

To view the full details of an audit record from the Results widget, click in the Details field for that record.
The names and values of the each field display on the table view of the Full Details window, but the specific field types vary according to audit type.

- **ID** - lists the audit ID number.
- **Time Stamp** - lists the date and time of the event's occurrence.
- **ArtifactName** - lists the title of the artifact on which the event was performed.
- **Action Name** - lists the name of the action performed.
- **ActionID** - lists the action ID number of the action performed.
- **ObjectName** - lists the type of object on which the event was performed.
- **ObjectTypeID** - lists the Object Type ID number of the object on which the action was performed.
- **ExecutionTime** - lists the duration of the event.
- **ArtifactID** - lists the artifact ID of the audit.
- **UserName** - lists the user name of the user who performed the action.
- **UserID** - lists the UserID number of the user who performed the action.
- **Details.auditElement.QueryText** - lists the SQL query text of the action performed.
- **Details.auditElement.Milliseconds** - the time it took the audit action to run.

Click the **Display in JSON format** link at the bottom of the window to display all of the above information related to the audit record in JSON format.
The **Display in table format** link reverts the display to the original formatting.

Click **Close** in the upper right corner of the window to exit the Full Details window.

### 15.5 Creating dashboards

You can organize any of the above widgets according to your preferences by creating a custom dashboard. Use the following steps to create your own custom dashboard:
1. Click the pencil icon to bring up the **Dashboard** layout.

![Dashboard layout](image_url)

2. Enter the name of the dashboard in the **Name** field.

3. Enter an order number in the **Order** field to determine the order the dashboard appears in the drop-down menu.

4. Click **Save** when you've configured your dashboard according to your preferences. You can also use the drop-down menu to **Save As** if you want to save your dashboard as a copy of one that you might have chosen to edit rather than overwriting the dashboard you initially used as a template.

After you save it, the new dashboard is available for you to select from the dashboard drop-down next to the pencil icon.

![Dashboard drop-down menu](image_url)

### 15.6 Admin tab

The **Admin 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 Data Grid. You also have the option of running a migration error retry script.

#### 15.6.1 Migration Error Report

The **Migration Error Report** generates a report that provides information on all errors that have occurred while migrating audit data into the Data Grid repository, including batch and document errors. 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:** Beginning in Relativity 9.3.376.35, Data Grid for 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.
## 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.

### 15.6.2 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 the Data Grid repository.
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.

### 15.6.3 Migration Status Report

The Migration Status Report generates a report that shows the progress of the migration of audit data into the Data Grid repository. 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.
16 Restarting nodes and clusters

Because of the way that Data Grid rebalances data as nodes enter and exit the cluster, you must be careful when bringing down clusters or nodes for maintenance. This page explains how to restart nodes and clusters.

16.1 Preparing the cluster for node restart

Data Grid automatically detects when a data node leaves the cluster and immediately rebalances data across the cluster while accounting for the node's absence. Each time the cluster re-balances, there is a cost to your server resources. If you have a window of maintenance time, you can turn off automatic recovery re-balancing by setting the node to persistent and the cluster.routing.allocation.enable configuration value to "none."

Note: Run any of the Sense queries below on a client node except when executing a shutdown. Run shutdown queries on a master node unless you only want to shut down a specific node locally. Run shutdown queries for the entire cluster on the master node so that all nodes shut down before the master node. If you want to run a shutdown locally, you can run it on that specific node instead of the master.

Before restarting a node, disable your Data Grid Audit Migrator agents and stop all indexing and importing in all Data Grid-enabled workspaces for five minutes. This will allow the sync process to complete before you begin the node restart process.

Run the following command to turn off re-balancing and set the cluster to persistent. The synced flush ensures that all shards have been synced after five minutes. The persistent state ensures that re-balancing stays off when the cluster restarts.

```
GET _cluster/settings
POST /_all/_flush/synced
PUT _cluster/settings
{
  "persistent":{"cluster.routing.allocation.enable": "none"}
}
```

16.2 Shutting down a node

Use the following steps to shut down a node:

1. Run the Windows command prompt as an administrator.
2. Navigate to the bin directory in the RelativityDataGrid folder.

   ```
   C:\RelativityDataGrid\elasticsearch-main\bin
   ```

3. Stop the Data Grid service by running the following command:

   ```
   .\kservice.bat stop
   ```
16.3 Restarting a node

Before restarting the node, apply the following cluster setting:

```
PUT /_cluster/settings
{
  "persistent" : {"cluster.routing.allocation.balance.threshold" : "100.0f"
}
}
```

You must restart the Data Grid service on each node individually to bring the cluster back up online. Use the following steps to restart a node:

1. Run the Windows command prompt as an administrator.
2. Navigate to the bin directory in the RelativityDataGrid folder.

   ```
   C:\RelativityDataGrid\elasticsearch-main\bin
   ```
3. Start the Data Grid service by running the following command and wait for the cluster to go from red to yellow:

   ```
   .\kservice.bat start
   ```
4. Use the following command to re-enable re-balancing of shards:

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

Once the node restarts and displays a green status, re-apply the following default setting:

```
PUT /_cluster/settings
{
  "persistent" : {"cluster.routing.allocation.balance.threshold" : "1.0f"
}
}
```

16.4 Restarting a cluster

Use the following command before restarting a cluster:

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

Once all nodes come back green, use the following command to re-enable allocation:

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

17.1 Configuring a monitoring cluster in Data Grid 2.x

Use the following steps to install Data Grid 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 Data Grid 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 Data Grid 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.

17.1.1 Setting up the Kibana server

Use the following steps to set up the Kibana server on your monitoring cluster:

1. Extract the Kibana 4.3.0 package on your monitoring cluster.
2. Navigate to the kibana.yml file, and update the following in a text editor:
   - elasticsearch.url - enter the name of your monitoring cluster.
3. 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, C:\RelativityDataGrid\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).
4. Browse to the kibana/bin folder and enter the following:
   - bin/kibana plugin -i elasticsearch/marvel/2.1.2
   - bin/kibana plugin -i elastic/sense
5. Start the Kibana server by running the following:
   - bin/kibana.bat
17.1.2 Updating your Data Grid cluster

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

Complete the following on all nodes in the Data Grid 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:\RelativityDataGrid\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
       auth:
         username:
         password:
   
   3. Restart the Data Grid service for the changes to take effect.