Elasticsearch

User Guide

User Guide

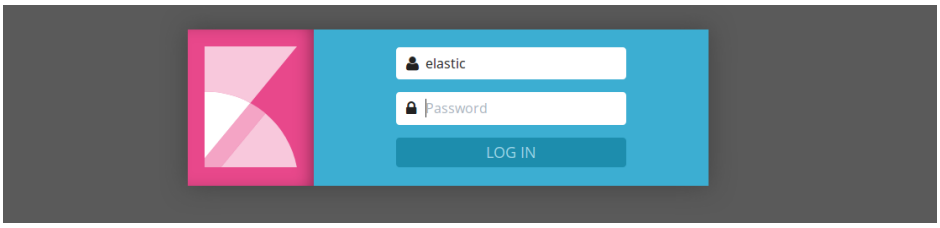
Instance management

Instance management

Elasticsearch instance management

Alibaba Cloud Elasticsearch supports multiple features for instance management, including **Kibana console**, **instance monitoring**, **instance restart**, and **refresh**.

Kibana console



Elasticsearch provides the Kibana console for business scaling.

The Kibana console is a part of the Elasticsearch ecosystem, which has been seamlessly integrated into Elasticsearch. The Kibana console enables you to view the running status of your Elasticsearch instances and manage these instances.

Instance monitoring

Elasticsearch supports instance monitoring. You can customize alert thresholds and enable Elasticsearch to use SMS alerts when any exceptions have been detected. For more information about

alerts, see Elasticsearch monitor alerts.

Instance restart

This feature allows you to use the **restart** and **force-restart** method to restart an Elasticsearch instance. Select a restart method based on your business scenario.

Restart the agent

This method ensures **service continuity** by keeping at least one replica running on the Elasticsearch instance during the restart process. However, a restart using this method takes a long period of time.

Notes:

- Make sure that the health status of your Elasticsearch instance is green.
- The CPU and memory usage of the Elasticsearch instance will experience a usage spike during the restart process. This may affect the stability of your service for a short period of time.

Force-restart

This method may cause service instability on the Elasticsearch instance during the restart process. However, this method takes less time.

Notes:

When an Elasticsearch instance has a high disk usage, such as 85% or higher, the health status of the instance may change to **yellow** or **red**. You cannot **restart** an instance in red or yellow health status. To restart the instance, you must use the **force-restart** method.

We recommend that you do not perform instance operations including **node scaling**, **disk scaling**, **restart**, **password modification**, and **configuration modification** when the health status of your Elasticsearch instance is yellow or red. Perform these operations when the health status of your instance is **green**.

If you change the configuration of an unhealthy instance that contains two or more **nodes**, the instance will remain in the **Applying** status. To resolve this issue, submit a ticket.

If you perform the **update**, **restart**, **scaling**, or **password reset** operation on an Elasticsearch instance that contains only one node, the service on the instance will become unavailable during the execution of the operation. To resolve this issue, create

an Elasticsearch instance and migrate your service to the newly created instance.

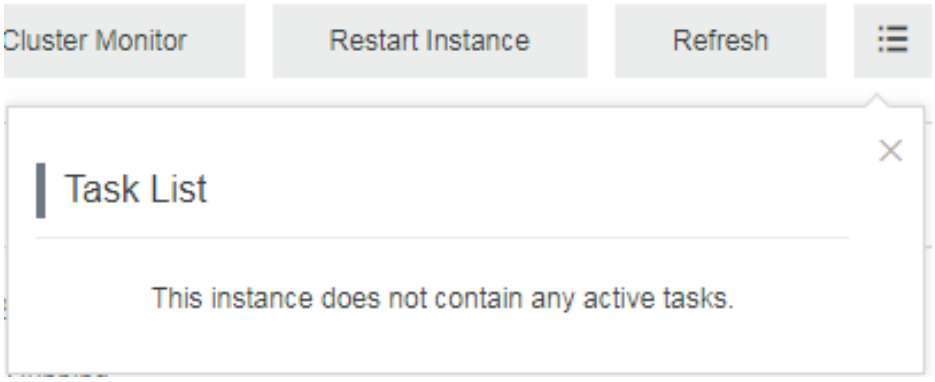
Refresh

In certain cases, the console may fail to update the information. For example, the console may fail to update the status of an Alibaba Cloud Elasticsearch instance after the instance has been successfully created. To resolve this issue, use the refresh function to manually refresh the status of the instance.

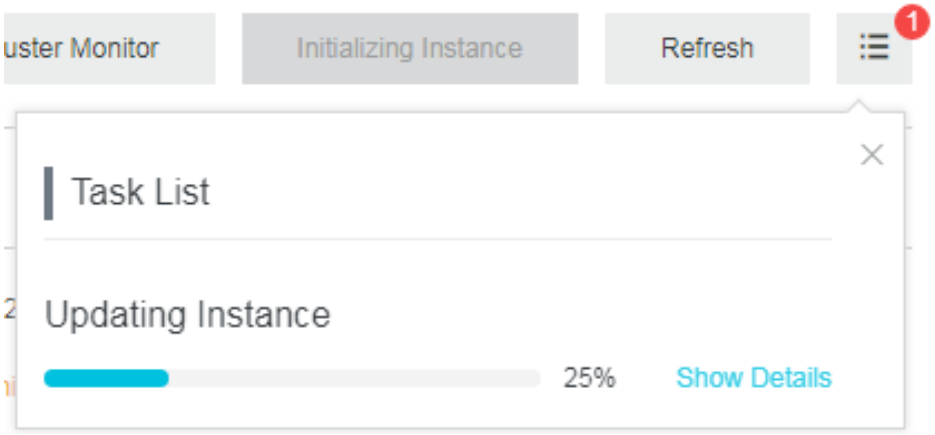
Task list

The Task list page shows running tasks on the current instance.

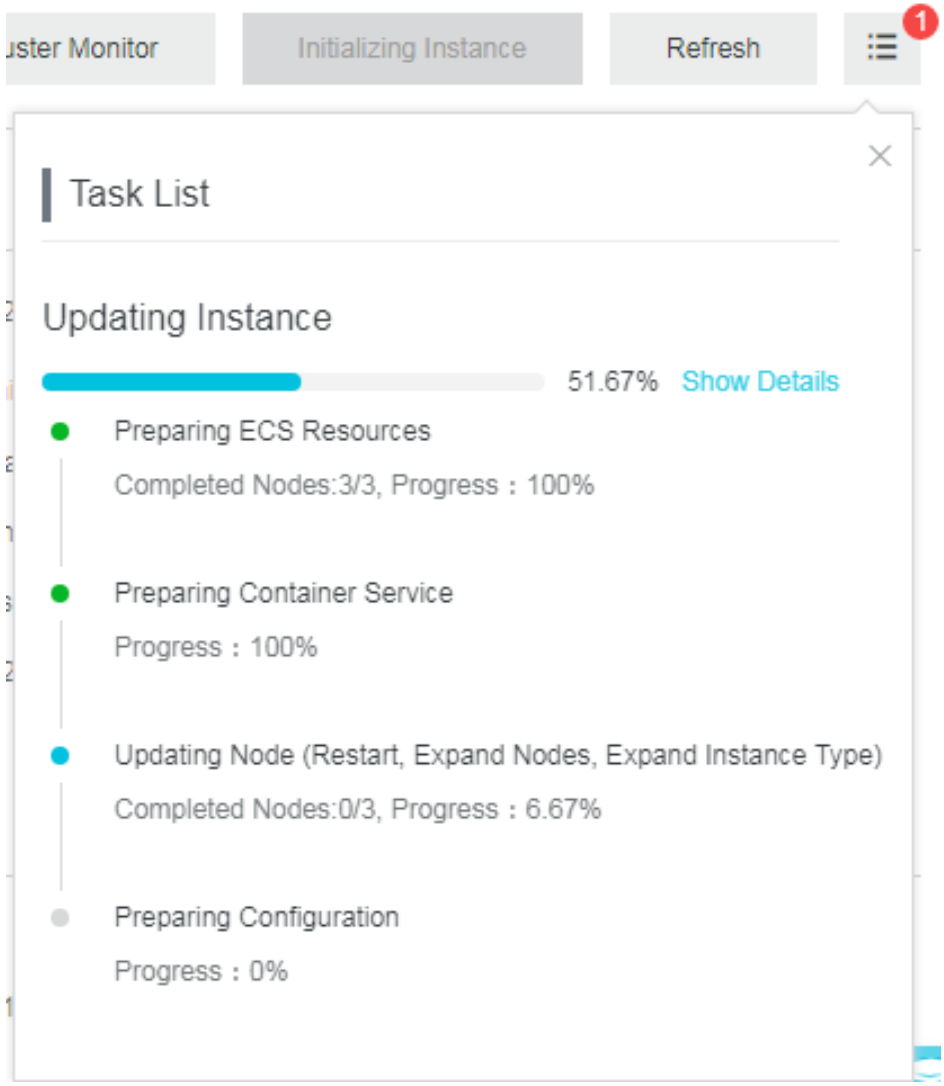
No task is running on the current instance.



Tasks running on the current instance.



Show detailed information about a running task.



Basic information

Renewal

Drag the slider to the right to select the renewal duration. The default renewal duration is 1 month (optional durations: 1–9 months, and 1–3 years).

Node scale-up

The scale-up options include **Instance Specifications, Single-node Storage Space**, and **Node Quantity**.

Scale-down is not supported.

Forced change

The Forced Change option is not selected. By default, scale-up is performed through Restart. However, before performing the scale-up operation, make sure that the ES cluster health status is green.

If the cluster health is in the red or yellow state and services are seriously affected, you need to scale up immediately. In this situation, choose Forced Change. This setting forces the change and restarts the cluster, ignoring the cluster health status.

- **Restart:** The instance continues providing highly-efficient, highly-available services (including at least one copy) during restart. In addition, it takes some time to restart.

  Forced change: The system forces the change and restarts the cluster, ignoring the cluster health status. This may affect service stability of the Elasticsearch cluster during restart, but it takes a short time to restart.

  Note:

  Only one attribute can be modified for every scale-up operation (node quantity, single-node storage space, or instance specifications).

  Before performing scale-up, make an evaluation.

  If the requested capacity is too large, the change goes into the approval state. After being approved, the change you submitted becomes effective.

  If the change volume is small, the change operation is triggered directly.

  If the node quantity is modified, you can check the consumption amount of the updated order. After the order is submitted and the instance takes effect, the consumption amount is calculated upon the modification.

Name

By default, the instance name is the same as the instance ID. You can customize the instance name, and search the instance list on the console for the corresponding instance name.

dedicated master

Add a Dedicated Master node to your Alibaba Cloud Elasticsearch instance. If you select Dedicated Master on the purchase page (recommended for better service stability), the dedicated master service is displayed as enabled on the basic instance information page.

> **Note**:
>
> If you buy 10 or more nodes, Dedicated Master is enabled by default for free.

Intranet address

It is used to access the Alibaba Cloud Elasticsearch service within the VPC. Public addresses cannot access the Elasticsearch service.

Other parameters

For the parameters not mentioned on the basic information page, see the parameter description.

Cluster Extension

You can currently scale the **instance specification**, **number of nodes**, **dedicated master specification**, and **storage space per node**.

Current configuration

This page shows the configuration of the current Elasticsearch instance.

Modify configuration

Change the configuration of your Elasticsearch instance according to the instructions on the page to meet your business needs. Fore information about some parameters, see Buy page parameters.

> **Notes:**
>
> - Each up-scaling can upgrade only one of these specifications: **number of nodes**, **storage space per node**, **instance specification**, and **dedicated master specification**.

- **You cannot change the storage type on the cluster expansion page. You can only change the size of the storage space.**
- The up-scaling operation restarts your Alibaba Cloud Elasticsearch instance.
- Currently, down-scaling is not supported. You cannot reduce the number of nodes or storage space, or scale down the node specification.

Notes:

- If you have already purchased dedicated master nodes, scaling the number of nodes does not restart your Alibaba Cloud Elasticsearch instance.
- If the health status of an Elasticsearch instance is not green, you must select **Ignore the health state of the cluster** to perform the scaling operation. The scaling operation may affect your services.
- If your business requires a scaling, first perform an Elasticsearch resource assessment.
- The console will update the payment amount of the scaling order in real time based on the number of nodes that you have modified.
- After you have submitted the scaling order, Alibaba Cloud Elasticsearch will calculate the fee based on the submitted order.

Specification families and instance types

Change the specification family and instance type according to the instructions on the page. For more information, see Buy page parameters.

Notes:

- Data nodes cannot be scaled up if they are using the local disk specification.
- Specification families cannot be modified.

Amount

Change the number of data nodes according to the instructions on the page. For more information, see Buy page parameters.

Dedicated master nodes

You can select **Dedicated Master Node** on the cluster expansion page to purchase dedicated master nodes or upgrade the specification of your purchased dedicated master nodes. The dedicated master nodes will be billed based on the specification that you have specified after they are upgraded. For more information, see Buy page parameters.

Notes:

If you have already purchased dedicated master nodes of 1-Core 2 GB, you can select the **Dedicated Master Node** option on the cluster expansion page to purchase dedicated master nodes with higher specification. The dedicated master nodes will be billed based on the new specification. **If you are using free dedicated master nodes and you have scaled up these nodes, they will be billed based on the new specification.**

You can select **Dedicated Master Node** on the cluster expansion page to upgrade the **dedicated mater node specifications**. The upgraded dedicated master nodes will be billed based on the new specification.

You can select **Dedicated Master Node** on the cluster expansion page to purchase dedicated master nodes or scale up the specification of your purchased dedicated master nodes. By default, three dedicated master nodes of 2-Core 8 GB are used. The storage type of the dedicated master nodes is cloud disk. Each dedicated master node is assigned 20 GB of storage space.

Client nodes

You can select **Client Node** on the cluster expansion page to purchase client nodes or upgrade the specification of your purchased client nodes. The upgraded client nodes will be billed based on the new specification. For more information, see Buy page parameters.

Notes:

Select the **Client Node** option on the cluster expansion page to upgrade the **client node specifications**. The upgraded client nodes will be billed based on the new specification.

Select a client node on the cluster expansion page to purchase client nodes or scale up your purchased client nodes. By default, two client nodes of 2-Core 8 GB are used. The storage type of the client nodes is cloud disk. Each client node is assigned 20 GB of storage space.

Instance restart

If the health status of your Elasticsearch instance is **green**, typically the instance can continuously provide services during the up-scaling and restart process. However, the restart process is time-consuming. To ensure the continuity of your services, make sure that your instance has a minimum of one replica. **In some situations, the health status of your Elasticsearch instance may temporarily change to red during the restart process.**

Notes:

- The nodes of an Elasticsearch instance may have a CPU and memory usage spike during the restart process. Your queries or pushing services may become unstable or fail. Typically, these services will recover after a brief period of time. **In some situations, the health status of your Elasticsearch instance may temporarily change to red during the restart process.**
- Make sure that the health status of your Elasticsearch instance is **green**.

Force-scaling

If the health status of your Elasticsearch instance is **red** or **yellow**, this means that your service has been severely affected. To resolve this issue, you must immediately scale up your instance. You can select **Force-Scaling** to **forcibly scale up** your instance regardless of the health status of the instance. Force-scaling only takes a short period of time.

Notes:

- The **Force-scaling** operation will **restart** the specified Elasticsearch instance.
- If you do not select **Force-Scaling**, Elasticsearch uses the **restart** method to restart your instance by default.
- The **force-scaling** option is automatically selected by default if the health status of your Elasticsearch instance is **red** or **yellow**. You cannot restart an instance in **red** or **yellow** health status by using the **Restart** method.
- The force-scaling operation will cause service instability during the restart process of the instance.

Storage

Change the storage space of a node according to the instructions on the page. For more information, see **Buy page parameters**.

Advanced configuration

System configuration

Elasticsearch provides highly available security service and advanced system service for your clusters.

Blacklist and whitelist

Alibaba Cloud Elasticsearch provides X-Pack plugin of business edition to protect access security when you access a node through an intranet address of Alibaba Cloud Elasticsearch instance. Configure the IP address blacklist and whitelist to restrict access to Alibaba Cloud Elasticsearch instance.

> **Note**:
>
> If the IP address of a node is in the blacklist, X-Packet allows the link to this node, but the node’ s requests are be dropped.

Blacklist and whitelist configuration

You can configure the blacklist and whitelist on the console page. According to your configuration, the system filters IP addresses by setting xpack.security.transport.filter.allow and xpack.security.transport.filter.deny in elasticsearch.yml.

Priority

The permit rules have higher priority than deny rules.

Configuration method

- Enter the IP addresses, for example, 192.168.0.1 and 192.168.0.0/24, and separate them with commas.
- Permit or deny all IP addresses by entering _all.
- Support IPv6 addresses, for example, 2001:0db8:1234::/48 and 1234:0db8:85a3:0000:0000:8a2e:0370:7334, and separate them with commas.
- Support extensive domain names, for example, .google.com.

  Use localhost as the local domain name, for example localhost.

  > **Note**:
  >
  > Elasticsearch can disable the IP address filter function. Disabling this function can improve system performance in some situations. However, the xpack.security.transport.filter.enabled field in the elasticsearch.yml file must be set to false. To ensure security, Alibaba Cloud Elasticsearch does not open this function to users.

Password reset

Password reset affects cluster management. The new password takes effect only after the cluster update is complete. After the new password takes effect, you can use it to log on to the Kibana console and access the Elasticsearch service.

IK custom dictionary

Edit the dictionary. Upload the main segmentation dictionary and disabled dictionary. They will take effect after the ES instance restarts.

Main segmentation dictionary

New indexes use the updated main segmentation dictionary for word segmentation. Every word occupies one row, and the file is saved as a .dic file in the UTF-8 format. Example:
Brexit
PUBG

Disabled dictionary

The words in the disabled dictionary are not indexed.Every word occupies one row, and the file is saved as a .dic file in the UTF-8 format.Example:
Whereas
For

Yml file configuration

The console page shows the current cluster configuration.

Configuration modification

Automatic index creation: Whether the Elasticsearch automatically creates index information represented by the index name after receiving a new index.

Specified index name deletion: Whether the index name needs to be specified when an index is deleted. If wildcard is supported, a batch of indexes can be deleted. Deleted indexes cannot be recovered. Therefore, be cautious when performing this operation.

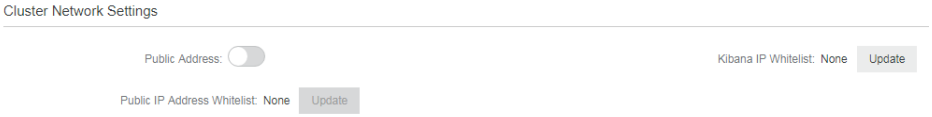
accesslog index: Records the index logs for the Add, Delete, and Query operations on

Elasticsearch. This option is not recommended because the logs occupy the disk space and affect system performance. Exercise caution when using this option.

Network configuration and backup

Cluster network settings

Enable the **Public Address** and set the **Public IP Address Whitelist** and **Kibana IP Whitelist**.



Public addresses

When the public address switch is green, this feature is enabled. **By default, this feature is disabled**.

Public IP address whitelist

You can add **IP addresses** or **network segment addresses** separated with commas (,) to the Kibana access whitelist, such as 192.168.0.1 or 192.168.0.1/24.

Notes:

- **By default, the public address whitelist is set to 127.0.0.1 to stop all public IP addresses from accessing Elasticsearch**.
- You can only use the whitelist to control access to Elasticsearch by using public addresses.
- You cannot set the Public IP address whitelist to allow all public IP addresses to access Elasticsearch.

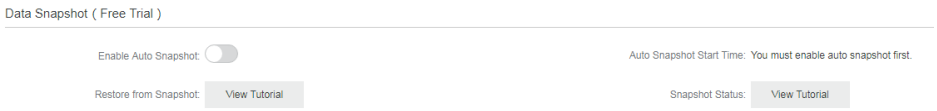
Kibana IP whitelist

You can add **IP addresses** or **network segment addresses** separated with commas (,) to the Kibana access whitelist, such as 192.168.0.1 or 192.168.0.1/24.

Notes:

- **By default, the Kibana IP whitelist is set to allow all public IP addresses to access Kibana.**
- By default, the Kibana access whitelist is set to 127.0.0.1 to prevent all IP addresses from accessing Elasticsearch. **
- You can use only the Kibana access whitelist to control access to Kibana.

Data snapshot (free trial)



Enable auto snapshot

When the **Enable Auto Snapshot** switch is **green**, this feature is enabled. **By default, this feature is disabled**.

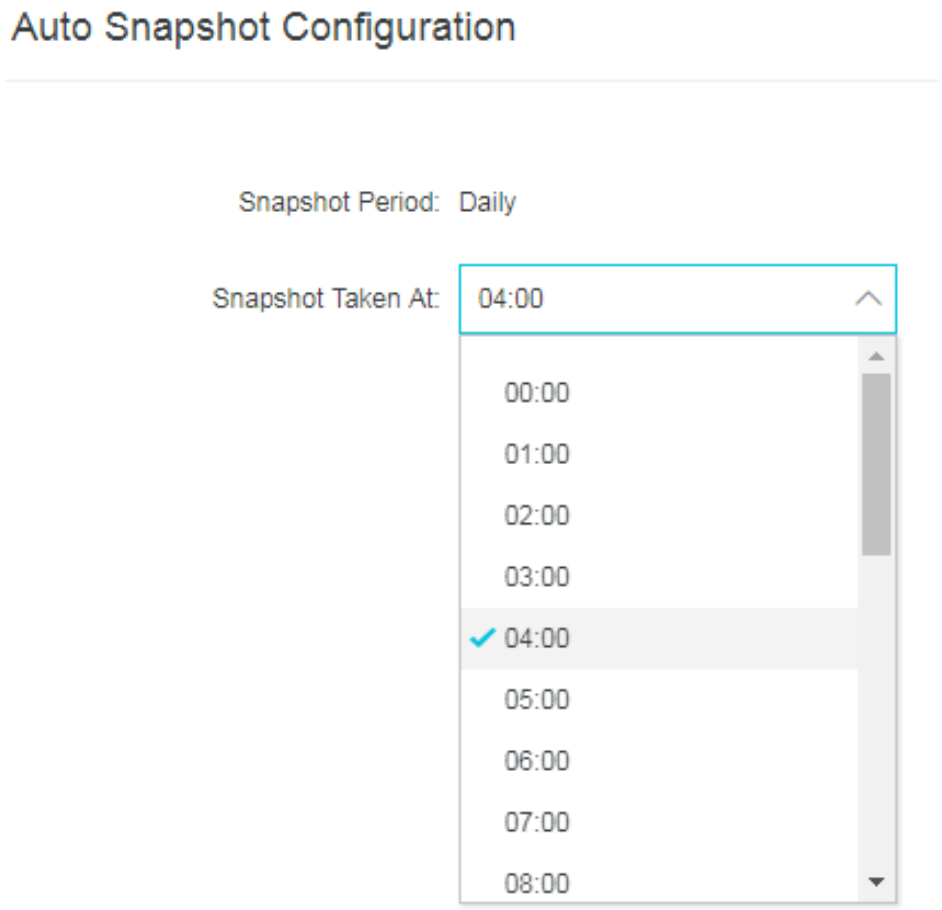
Auto snapshot start time

If the **auto snapshot** feature is disabled, the system displays **You must enable auto snapshot first**.

> **Note:**
>
> If the **auto snapshot** feature is enabled, auto snapshot starts immediately at the current system time in the corresponding region. Do not perform snapshot actions when the system is creating snapshots.

Modify configuration

If the **auto snapshot** feature is enabled, click **Modify Configuration** to modify the auto snapshot start time.



Notes:

- The snapshot period is set to daily.
- The snapshot time is specified in hours. Valid values: 0 to 23.

Restore from snapshots

For more information, see Restore from snapshots.

Snapshot status

For more information, see Restore from snapshots or View backup information.

Automatic backup guide

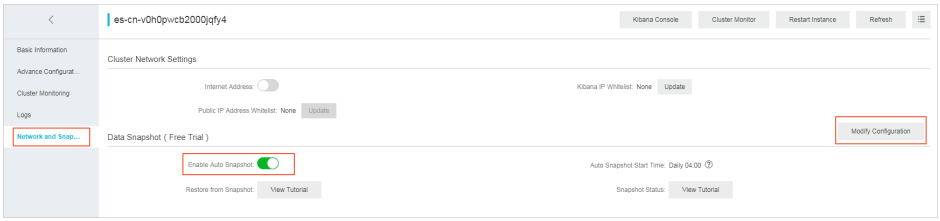
Enable automatic backup

Log on to the Alibaba Cloud Elasticsearch console.

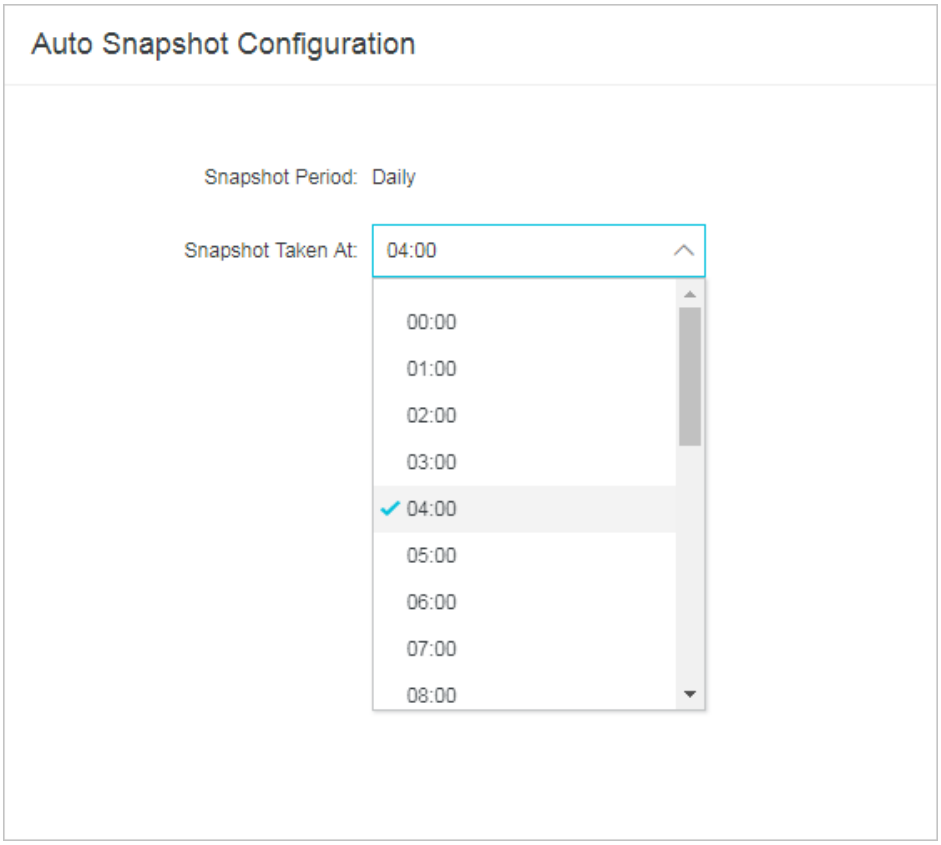
Click the name of your Elasticsearch instance to enter its information page.

Click the **Network and Snapshots** in the left-side navigation pane on the instance information page.

Open the **Enable Auto Snapshot** switch in the **Data Snapshot** area to enable the automatic backup function.



Click **Modify Configuration** in the right corner of the page to change the time for performing the daily automatic backup.



Restore snapshots

If you have enabled automatic backup for your Elasticsearch instance, Elasticsearch creates a daily snapshot of the data on your instance. You can run the Elasticsearch snapshot command to restore a snapshot.

Notes:

If this is the first time that Elasticsearch creates a snapshot for an instance, the snapshot contains all data on the instance. Subsequent snapshots only contain incremental data that is newly added to the instance after the first snapshot creation. Therefore, creating the first snapshot is time-consuming.

Only snapshots created within the last five days are kept.

Snapshots do not store monitoring data, such as indexes prefixed with .monitoring and .security_audit, on Elasticsearch instances.

A snapshot can only be used to restore data for the instance that the snapshot was created from.

Elasticsearch automatically creates a snapshot repository when creating the first snapshot for an instance.

View all repositories

Run the following command to view the snapshot repository information:

```
GET _snapshot
```

Response:

```
{
"aliyun_auto_snapshot": {
"type": "oss",
"settings": {
"compress": "true",
"base_path": "xxxx",
"endpoint": "xxxx"
}
}
}
```

aliyun_auto_snapshot: Specifies the name of the repository.

Type: Specifies the snapshot storage. In this example, Alibaba Cloud Object Storage Service (OSS) is used to store snapshots.

compress:true: Enables compression mode. When this mode is enabled, Elasticsearch compresses the metadata of the indexes when creating snapshots.

base_path: Specifies the path where the snapshots are stored in OSS.

endpoint: Specifies OSS region information.

Default parameters

Automatic snapshot provides the following default parameters, which are not shown in the response:

max_snapshot_bytes_per_sec:40mb: Throttles the maximum snapshot rate per node at 40 MB per second.

max_restore_bytes_per_sec:40mb: Throttles the maximum restore rate per node at 40 MB per second.

chunk_size: Max 1Gb: Specifies the size of chunks that big files can be broken down into during snapshot creation. The chunk size is 1 GB.

View all snapshots

Run the following command to view all snapshots in the aliyun_auto_snapshot repository.

```
GET _snapshot/aliyun_auto_snapshot/_all
```

Response:

```
{
"snapshots": [
{
"snapshot": "es-cn-abcdefghijklmn_20180627091600",
"uuid": "MMRniVLPRAiawSCm8D8Dug",
"version_id": 5050399,
"version": "5.5.3",
"indices": [
"index_1",
".security",
".kibana"
],
"state": "SUCCESS",
"start_time": "2018-06-27T01:16:01.009Z",
"start_time_in_millis": 1530062161009,
"end_time": "2018-06-27T01:16:05.632Z",
"end_time_in_millis": 1530062165632,
"duration_in_millis": 4623,
"failures": [],
"_shards": {
"total": 12,
"failed": 0,
"successful": 12
}
}
]
}
```

Restore indexes

You can use the _restore command of Elasticsearch to restore indexes from a snapshot.

Restore all indexes in the specified snapshot in the **aliyun_auto_snapshot** repository. This

task is executed in the back end.

```
POST _snapshot/aliyun_auto_snapshot/<snapshot>/_restore
```

<snapshot>: Specifies the name of the snapshot, for example, es-cn-abcdefghijklmn_20180627091600.

Restore all indexes in the specified snapshot in the **aliyun_auto_snapshot** repository and wait for the system to finish the task.

The _restore command is an asynchronous command. Elasticsearch immediately returns a response after it confirms to execute a restoration task. The restoration task will be executed in the back end. You can add the wait_for_completion` parameter to the request to require Elasticsearch to returns a response after it finishes the restoration task.

```
POST _snapshot/aliyun_auto_snapshot/<snapshot>/_restore?wait_for_completion=true
```

<snapshot>: Specifies the name of the snapshot, for example, es-cn-abcdefghijklmn_20180627091600.

Restore the specified index in the snapshot in the **aliyun_auto_snapshot** repository and rename the index. This task is executed in the back end.

```
POST _snapshot/aliyun_auto_snapshot/<snapshot>/_restore
{
"indices": "index_1",
"rename_pattern": "index_(.+)",
"rename_replacement": "restored_index_$1"
}
```

<snapshot>: Specifies the name of the snapshot, for example, es-cn-abcdefghijklmn_20180627091600.

indices: Specifies the name of the index to restore.

rename_pattern: Uses regular expression to restore all matching indexes. This parameter is optional.

rename_replacement: Uses regular expression to rename all matching indexes. This parameter is optional.

View backup information

View automatic backup information

After enabling automatic backup, you can log on to the Kibana console that has been integrated into Alibaba Cloud Elasticsearch and run the Elasticsearch snapshot command in Dev Tools to view snapshots.

View all snapshots

Run the following command to view all the snapshots that are located in the **aliyun_auto_snapshot** repository.

```
GET _snapshot/aliyun_auto_snapshot/_all
```

Response:


```
{
"snapshots": [
{
"snapshot": "es-cn-abcdefghijklmn_20180628092236",
"uuid": "n7YIayyZTm2hwg8BeWbydA",
"version_id": 5050399,
"version": "5.5.3",
"indices": [
".kibana"
],
"state": "SUCCESS",
"start_time": "2018-06-28T01:22:39.609Z",
"start_time_in_millis": 1530148959609,
"end_time": "2018-06-28T01:22:39.923Z",
"end_time_in_millis": 1530148959923,
"duration_in_millis": 314,
"failures": [],
"shards": {
"total": 1,
"failed": 0,
"successful": 1
}
},
{
"snapshot": "es-cn-abcdefghijklmn_20180628092500",
"uuid": "frdl1YFzQ5Cn5xN9ZWuKLA",
"version_id": 5050399,
```

```
"version": "5.5.3",
"indices": [
".kibana"
],
"state": "SUCCESS",
"start_time": "2018-06-28T01:25:00.764Z",
"start_time_in_millis": 1530149100764,
"end_time": "2018-06-28T01:25:01.482Z",
"end_time_in_millis": 1530149101482,
"duration_in_millis": 718,
"failures": [],
"shards": {
"total": 1,
"failed": 0,
"successful": 1
}
}
]
}
```

state: Specifies the status of a snapshot. The snapshot status includes the following:

IN_PROGRESS: The snapshot is being restored.

SUCCESS: The snapshot has been restored and all shards have been successfully stored.

FAILED: The snapshot has been restored with an error. Some data cannot be stored.

PARTIAL: The snapshot has been successfully restored to an instance. However, one or more shards cannot be stored.

INCOMPATIBLE: The snapshot version is incompatible with the current instance version.

View specified instance

Run the following command to view detailed information about the specified snapshot in the **aliyun_auto_snapshot** repository.

```
GET _snapshot/aliyun_auto_snapshot/<snapshot>/_status
```

<snapshot>: Specifies the name of the snapshot, for example, es-cn-abcdefghijklmn_20180628092236.

Response:

```
{
```

```
"snapshots": [
{
"snapshot": "es-cn-abcdefghijklmn_20180628092236",
"repository": "aliyun_auto_snapshot",
"uuid": "n7YIayyZTm2hwg8BeWbydA",
"state": "SUCCESS",
"shards_stats": {
"initializing": 0,
"started": 0,
"finalizing": 0,
"done": 1,
"failed": 0,
"total": 1
},
"stats": {
"number_of_files": 4,
"processed_files": 4,
"total_size_in_bytes": 3296,
"processed_size_in_bytes": 3296,
"start_time_in_millis": 1530148959688,
"time_in_millis": 77
},
"indices": {
".kibana": {
"shards_stats": {
"initializing": 0,
"started": 0,
"finalizing": 0,
"done": 1,
"failed": 0,
"total": 1
},
"stats": {
"number_of_files": 4,
"processed_files": 4,
"total_size_in_bytes": 3296,
"processed_size_in_bytes": 3296,
"start_time_in_millis": 1530148959688,
"time_in_millis": 77
},
"shards": {
"0": {
"stage": "DONE",
"stats": {
"number_of_files": 4,
"processed_files": 4,
"total_size_in_bytes": 3296,
"processed_size_in_bytes": 3296,
"start_time_in_millis": 1530148959688,
"time_in_millis": 77
}
}
}
}
}
}
```

```
]
}
```

ES self-built functions

Elasticsearch official documentation

Alibaba Cloud Elasticsearch is based on the open-source official Elasticsearch V5.5.3. For details, see Elasticsearch Definitive Guide.

SDK client

Only HTTP-based access requests from the client are supported. You can use the Java REST Client officially provided by Elasticsearch.

yml configuration

Customize CORS access

For more configurations, visit the Elasticsearch official website and view the HTTP information.

Configurations

- Configurations in the table below are custom HTTP-based configurations opened by Alibaba Cloud Elasticsearch.
- The following configurations only support the **static configuration** mode, and do not support the hot deployment mode. To implement a configuration described below, write it to the elasticsearch.yml file.
- The following configurations depend on the cluster network settings(Network settings).

Configuration item	Description
http.cors.enabled	CORS (Cross-origin Resource Sharing) configuration item. It can be used to **enable** or **disable** CORS resource accesses, in other

	words, whether to allow Elasticsearch to receive requests sent from browsers of resources in other domains. It the parameter is set to true, Elasticsearch can process the OPTIONS CORS requests. If the domain information in the sent request is already declared inhttp.cors.allow-origin, Elasticsearch adds Access-Control-Allow-Origin in the header to respond to the CORS request. If the parameter is set to false (the default value), Elasticsearch neglects the domain information in the request header and does not add the Access-Control-Allow-Origin to the header, to disable CORS accesses. If the client neither supports pre-flight requests that add the domain information header, nor checks Access-Control-Allow-Origin in the header of the packet returned from the server, the secured CORS access is affected. If Elasticsearch disables CORS accesses, the client can check whether a response is returned only by sending the OPTIONS request.
http.cors.allow-origin	CORS resource configuration item. It can be used to specify requests from which domains are accepted. No domain is allowed and the parameter is left blank by default. If / is added before the parameter value, the configuration is identified as a regular expression, which means that HTTP and HTTPS domain requests meeting the regular expression are supported. For example, /https?:\/\/localhost(:[0-9]+)?/ means requests meeting the regular expression are responded. * is deemed as a valid configuration and the cluster supports CORS requests from any **domain**, resulting in **security risks** to the Elasticsearch cluster.
http.cors.max-age	The browser can send an OPTIONS request to get the CORS configuration. max-age can be used to set how long the browser can retain the output result cache. The default value is 1728000 seconds (20 days).
http.cors.allow-methods	Request method configuration item. The optional values are OPTIONS, HEAD, GET, POST, PUT, and DELETE.
http.cors.allow-headers	Request header configuration item. The default value is X-Requested-With, Content-Type, Content-Length.
http.cors.allow-credentials	Credential configuration item. It is used to

	specify whether to return Access-Control-Allow-Credentials in the response header. If the parameter is set to true, Access-Control-Allow-Credentials is returned. The default value is false.

Customize remote re-indexing (whitelist)

The re-indexing component allows you to reconstruct the data index on the remote Elasticsearch cluster. This function can work in remote Elasticsearch of any version available. It allows you to index data of earlier versions to the current version.

```
POST _reindex
{
"source": {
"remote": {
"host": "http://otherhost:9200",
"username": "user",
"password": "pass"
},
"index": "source",
"query": {
"match": {
"test": "data"
}
}
},
"dest": {
"index": "dest"
}
}
```

host must contain the **protocol supported**, **domain name**, **port**, and other information, for example, https://otherhost:9200.

username and password are optional. If the remote Elasticsearch server needs **Basic Authorization**, enter the parameter in the request. When Basic Authorization is required, use the https protocol; otherwise, the password is transmitted as a text.

Only after the remote host address is declared in elasticsearch.yaml by using the reindex.remote.whitelist attribute can it call the API remotely. The combination of host and port is allowed, and multiple host configurations should be separated by commas (,), for example, otherhost:9200, another:9200, 127.0.10.**:9200, localhost:**. The whitelist does not identify the protocol and only uses the host and port information for security policy configuration.

If the host address is already listed in the whitelist, the query request is no longer verified and modified; instead, the request is directly sent to the remote server.

Note:

Indexing from a remote cluster does not support **manual slicing** or **automatic slicing**. For details, see Manual slicing or Automatic slicing.

Batch settings

The remote service uses a stack to cache indexed data. The default maximum size is 100 MB. If the remote index contains a large document, set the size of batch settings to a small value.

In the example below, the size of batch settings is 10, which is the minimum value:

```
POST _reindex
{
"source": {
"remote": {
"host": "http://otherhost:9200"
},
"index": "source",
"size": 10,
"query": {
"match": {
"test": "data"
}
}
},
"dest": {
"index": "dest"
}
}
```

Time-out period

- Use socket_timeout to set the read time-out period of socket. The default value is 30s.
- Use connect_timeout to set the connection time-out period. The default value is 1s.

In the example below, the read time-out period of socket is one minute and the connection time-out period is 10 seconds.

```
POST _reindex
{
"source": {
"remote": {
```

```
"host": "http://otherhost:9200",
"socket_timeout": "1m",
"connect_timeout": "10s"
},
"index": "source",
"query": {
"match": {
"test": "data"
}
}
},
"dest": {
"index": "dest"
}
}
```

Snapshot and recovery

You can use the snapshot API to back up your Alibaba Cloud Elasticsearch cluster. The API obtains the current status and data of your cluster and saves them to a shared repository. The backup process is intelligent.

The first snapshot is a complete copy of data, and all subsequent snapshots only save the difference between existing snapshots and the new data. As you create snapshots for data from time to time, the backups are incrementally added and deleted. It means that the number of subsequent backups increases quite fast because only a very small data volume is transmitted.

> **Note**:
>
> Tags <1>, <2>, and <3> in the code in this article are used to mark the positions for convenient description on the code at the specified position. Remove these tags when running the code.

Create a repository

OSS data sources of the standard storage type are recommended. OSS data sources of the archive storage type are not supported.

<1> The OSS data source must be in the same region of your Alibaba Cloud Elasticsearch cluster. Enter the intranet address of the region in the endpoint field. For details, see the intranet endpoint for ECS access column in Access domain name and data center.

<2> An OSS bucket must exist.

```
PUT _snapshot/my_backup
{
"type": "oss",
"settings": {
"endpoint": "http://oss-cn-hangzhou-internal.aliyuncs.com", <1>
"access_key_id": "xxxx",
"secret_access_key": "xxxxxx",
"bucket": "xxxxxx", <2>
"compress": true
}
}
```

Define the shard size

If the size of the data to be uploaded is very large, we can define the shard size during snapshot. If the shard size is exceeded, the data is divided into multiple shards and then uploaded to the OSS instance.

- <1> Note that the POST instead of PUT method is set. In this way, the exiting repository settings are updated.

<2> The start position of base_path to set the repository is the root directory by default.

```
POST _snapshot/my_backup/ <1>
{
"type": "oss",
"settings": {
"endpoint": "http://oss-cn-hangzhou-internal.aliyuncs.com",
"access_key_id": "xxxx",
"secret_access_key": "xxxxxx",
"bucket": "xxxxxx",
"chunk_size": "500mb",
"base_path": "snapshot/" <2>
}
}
```

List the repository information

```
GET _snapshot
```

You can use GET _snapshot/my_backup to obtain information of the specified repository.

Migrate backup snapshots

To migrate a snapshot to another cluster, you just need to back up the snapshot to the OSS instance, register a snapshot repository (in the same OSS instance) on the new cluster, set the base_path to the position where the backup file is saved, and then execute the backup restoration command.

All opened indexes of a snapshot

A repository can contain multiple snapshots and each snapshot is related to a series of indexes, for example, all indexes, shard of indexes, or a single index. When creating a snapshot, specify an index for the snapshot and create a unique name for the snapshot.

Snapshot command

The following is a most basic snapshot command:

```
PUT _snapshot/my_backup/snapshot_1
```

This command is used to back up all opened indexes to the snapshot named snapshot_1 in the repository my_backup. The call request is immediately returned and the snapshot is executed at the background.

If you want to wait until the execution finishes, add the tag wait_for_completion in the script.

```
PUT _snapshot/my_backup/snapshot_1?wait_for_completion=true
```

Then, the call is blocked until the snapshot execution finishes. If the snapshot size is large, it takes a long time to return the call request.

Specified indexes of a snapshot

All opened indexes are backed up by default. If Kibana is used and you do not want to back up all .kibana indexes related to diagnosis for disk space consideration,

you can back up only the specified indexes when creating snapshots for your cluster:

```
PUT _snapshot/my_backup/snapshot_2
{
"indices": "index_1,index_2"
```

```
}
```

When this snapshot command is run, only index1 and index2 are backed up.

List the snapshot information

Sometimes you may forget details about snapshots in the repository, especially when the snapshots are named according to the creation time, for example, backup_2014_10_28.

You can initialize a GET request on a repository and snapshot name to obtain the information of a single snapshot:

```
GET _snapshot/my_backup/snapshot_2
```

The response contains all the details about the snapshot:

```
{
"snapshots": [
{
"snapshot": "snapshot_1",
"indices": [
".marvel_2014_28_10",
"index1",
"index2"
],
"state": "SUCCESS",
"start_time": "2014-09-02T13:01:43.115Z",
"start_time_in_millis": 1409662903115,
"end_time": "2014-09-02T13:01:43.439Z",
"end_time_in_millis": 1409662903439,
"duration_in_millis": 324,
"failures": [],
"shards": {
"total": 10,
"failed": 0,
"successful": 10
}
}
]
}
```

You can use the placeholder _all to replace the specific snapshot name to obtain a complete list of all snapshots in the repository.

```
GET _snapshot/my_backup/_all
```

Delete a snapshot

You can initialize an HTTP-based call request on a repository/snapshot name through the DELETE API to delete all snapshots that are no longer used:

```
DELETE _snapshot/my_backup/snapshot_2
```

It is important to delete a snapshot through the APIs. Other methods, such as manual deletion, are not supported. Snapshots increase incrementally and many snapshots may depend on historical snapshots. The DELETE API knows which data is still used by recent snapshots and thus only deletes snapshots that are no longer used.

> Note:
>
> If you delete backups manually, there is a risk that the backups may be seriously damaged because the deleted backups may contain data that is still being used.

Monitor the snapshot task progress

The wait_for_completion tag provides the basic monitoring mode. If you want to restore the snapshots of a medium-sized cluster, this mode may be insufficient. The following two APIs provide more details about the snapshot status.

You can run a GET command for a snapshot ID to obtain the information of a specific snapshot:

```
GET _snapshot/my_backup/snapshot_3
```

If this command is run when the snapshot task is undergoing, you can view the start time, running time, and other information about the task.

> Note:
>
> This API uses the thread pool same as that of the snapshot mechanism. If the request is in a very large shard of the snapshot, the status update interval is large because the API is competing for resources in the same thread pool.

A better way is to drag the data through the _status API:

```
GET _snapshot/my_backup/snapshot_3/_status
```

The following are detailed statistics returned by the _status API:

```
{
"snapshots": [
{
"snapshot": "snapshot_3",
"repository": "my_backup",
"state": "IN_PROGRESS", <1>
"shards_stats": {
"initializing": 0,
"started": 1, <2>
"finalizing": 0,
"done": 4,
"failed": 0,
"total": 5
},
"stats": {
"number_of_files": 5,
"processed_files": 5,
"total_size_in_bytes": 1792,
"processed_size_in_bytes": 1792,
"start_time_in_millis": 1409663054859,
"time_in_millis": 64
},
"indices": {
"index_3": {
"shards_stats": {
"initializing": 0,
"started": 0,
"finalizing": 0,
"done": 5,
"failed": 0,
"total": 5
},
"stats": {
"number_of_files": 5,
"processed_files": 5,
"total_size_in_bytes": 1792,
"processed_size_in_bytes": 1792,
"start_time_in_millis": 1409663054859,
"time_in_millis": 64
},
"shards": {
"0": {
"stage": "DONE",
"stats": {
"number_of_files": 1,
"processed_files": 1,
"total_size_in_bytes": 514,
"processed_size_in_bytes": 514,
"start_time_in_millis": 1409663054862,
"time_in_millis": 22
}
},
...
```

- <1> The status of a running snapshot is IN_PROGRESS.

  <2> This specific snapshot still has a shard which is being uploaded. The other four shards have been uploaded.

  The response contains the overall information of the snapshot and statistics on each drilled-down index and shard. The following is a detailed figure about the snapshot task progress. Different shards of the snapshot can be in different states.

  INITIALIZING:The shard is checking the cluster status to see whether a snapshot can be created for it. This process is generally very fast.

  STARTED:The data is being transmitted to the repository.

  FINALIZING:Data transmission finishes and the shard is sending the snapshot metadata.

  DONE:The snapshot task is finished.

  FAILED:An error occurs when the snapshot is being processed, and the shard/index/snapshot task cannot be finished. You can check the log for more information.

Cancel a snapshot task

To cancel a snapshot task, you can run the following command when the task is running:

```
DELETE _snapshot/my_backup/snapshot_3
```

The snapshot process is interrupted. The half-done snapshots in the repository are deleted.

Restoration from a snapshot

If you have backed up the data, you just need to add _restore to the ID of the snapshot which you want to restore to the cluster:

```
POST _snapshot/my_backup/snapshot_1/_restore
```

All indexes saved in the snapshot are restored by default. If snapshot_1 contains five indexes, all the five indexes are restored to the cluster. Like the snapshot API, you can select a specific index to be restored.

You can use an additional option to rename the indexes. The option allows you to use a mode to match the index name and rename the index through the restoration process. If you want to restore historical data to verify the content or perform other operations without replacing existing data, this option is very useful. The following is an example about how to restore a single index from a snapshot and rename the index.

```
POST /_snapshot/my_backup/snapshot_1/_restore
{
"indices": "index_1", <1>
"rename_pattern": "index_(.+)", <2>
"rename_replacement": "restored_index_$1" <3>
}
```

The index_1 is restored to your cluster but is renamed to restored_index_1.

- <1> Only index_1 is restored. Other indexes in the snapshot are neglected.
- <2> Search for indexes being restored that can match the provided mode.
- <3> Rename the indexes to the alternative ones.

Like the snapshot, the restore command is immediately returned and the restoration process runs in the background. If you want your HTTP call request to be blocked until the restoration process is finished, add the wait_for_completion tag.

```
POST _snapshot/my_backup/snapshot_1/_restore?wait_for_completion=true
```

Monitor the restoration operation

The existing restoration mechanism of Elasticsearch is referenced for restoring data from the repository. As for internal realization, shard restoration from a repository is equivalent to restoration from another node.

To monitor the restoration progress, call the recovery API. This API is for general purpose and is used to display the status of moving shards in your cluster.

This API can be used to independently call a specified restored index.

```
GET restored_index_3/_recovery
```

It can also be used to view all indexes in your cluster, including moving indexes that are unrelated to the restoration process.

```
GET /_recovery/
```

The following is an output example. Note that a large quantity of content may be output if your cluster is highly active.

```
{
"restored_index_3" : {
"shards" : [ {
"id" : 0,
"type" : "snapshot", <1>
"stage" : "index",
"primary" : true,
"start_time" : "2014-02-24T12:15:59.716",
"stop_time" : 0,
"total_time_in_millis" : 175576,
"source" : { <2>
"repository" : "my_backup",
"snapshot" : "snapshot_3",
"index" : "restored_index_3"
},
"target" : {
"id" : "ryqJ5lO5S4-lSFbGntkEkg",
"hostname" : "my.fqdn",
"ip" : "10.0.1.7",
"name" : "my_es_node"
},
"index" : {
"files" : {
"total" : 73,
"reused" : 0,
"recovered" : 69,
"percent" : "94.5%" <3>
},
"bytes" : {
"total" : 79063092,
"reused" : 0,
"recovered" : 68891939,
"percent" : "87.1%"
},
"total_time_in_millis" : 0
},
"translog" : {
"recovered" : 0,
"total_time_in_millis" : 0
},
"start" : {
"check_index_time" : 0,
"total_time_in_millis" : 0
}
} ]
}
}
```

- <1> The type field indicates the restoration type. The shard is restored from a snapshot.
- <2> The source field indicates the snapshot and repository from which the shard is

restored.
- <3> The percent field indicates the restoration progress. The specified shard is restored by 94% until now. The restoration task will soon be finished.

All indexes under restoration and all shards in these indexes are listed in the output. Statistics on the start/end time, duration, restoration progress in percentage, and number of transmitted bytes, of each shard are displayed.

Cancel a restoration task

You can delete an index being restored to cancel a restoration task. You can modify the cluster status by calling the DeletIndex API to stop a restoration process. For example:

```
DELETE /restored_index_3
```

If restored_index_3 is being restored, after this deletion command is run, the restoration process stops and all the data that has been restored to the cluster is deleted.